

INFORMATIQUE

Classe de Terminale
ESG - EST - ETN

ALIMA GWLADYS LOUKMANOU MOHAMADOU
NDOBIA BRUNO NGA JEAN

Les auteurs

© Copyright

Tous les droits de traduction, de reproduction et d'adaptation réservés pour tous pays.

Le code de la propriété intellectuelle n'autorisant, aux termes des articles L.122-4 et L.122-5, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective» et, d'autre part, que «les analyses et les courtes citations» dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale ou partielle, faite sans consentement de l'auteur ou de ses ayant droit ou ayant cause, est illicite».

Cette représentation ou reproduction, par quelque procédé que ce soit, sans autorisation de l'éditeur constituerait donc une contrefaçon sanctionnée par les articles 425 et suivant le Code pénal.

SOMMAIRE

Préface	1
I PROGRAMMATION EN JAVASCRIPT	2
1 PRÉSENTATION DU LANGAGE JAVASCRIPT	3
1.1 Introduction	4
1.2 Éditeurs JavaScript	4
1.3 Navigateurs compatibles JavaScript	5
1.4 Connaissance en HTML	5
1.5 Limites de JavaScript	5
2 STRUCTURE D'UN PROGRAMME JAVASCRIPT	9
2.1 Les balises <script> et </script>	10
2.2 Les commentaires	10
2.3 Où inclure le code en JavaScript dans le code html ?	10
2.4 Les instructions	11
2.5 Le premier programme	12
3 SYNTAXE DU LANGAGE JAVASCRIPT	15
3.1 Les variables	16
3.2 les mots réservés	17

3.3	Les structures de contrôle	21
3.4	Les boucles	22
4	INTERACTIVITÉ DE JAVASCRIPT	25
4.1	Les événements	26
4.2	Les boîtes de dialogue	28
4.3	Les formulaires	29
4.4	Les fonctions	31
II	INITIATION AUX BASES DE DONNÉES	37
5	INTRODUCTION AUX SGBD	38
5.1	Introduction	39
5.2	La gestion des bases de données	39
5.3	Les principaux SGBD	39
5.4	Étude du SGBD ACCESS	40
6	CRÉER UNE BASE DE DONNÉES	49
6.1	Les bases de données	50
6.2	Les tables	51
6.3	Les champs d'une table	53
7	LES REQUÊTES	55
7.1	Notion de requêtes	56
7.2	Introduction au langage SQL	57
8	ÉTUDE DU SERVEUR DE BASES DE DONNÉES MYSQL	60
8.1	Concepts du modèle relationnel	61
8.2	Fonctions de MySQL	61

8.3	Interface avec PHP	63
8.4	Administration avec l'outil web phpMyAdmin	65
III	LES RESEAUX LOCAUX	70
9	CRÉER UN RÉSEAU LOCAL	71
9.1	Terminologie	72
9.2	Classification des réseaux	72
9.3	Caractéristiques des réseaux locaux	73
9.4	Équipements	79
10	PROTOCOLE RÉSEAUX	82
10.1	Notion de Protocole	83
10.2	Créer son propre réseau sous Windows XP	85
10.3	L'administration du réseau	86
10.4	Mise en place d'un réseau local	87
11	L'ALTERNATIVE DU SANS FIL : WI-FI	91
11.1	Introduction	92
11.2	Présentation des réseaux sans-fil	93
11.3	Les réseaux sans-fil étendus	96
11.4	Les réseaux sans-fil spécifiques	96
11.5	Les équipements WiFi	98
11.6	Sécuriser un réseau sans fil	99

PRÉFACE

Ce livre est un support pour les cours en informatique des classes de terminales ESG-EST-ETN. La structure et le contenu des chapitres de ce document ont été synchronisés avec le contenu du programme en vigueur. Le livre met l'accent sur les concepts et techniques fondamentaux des programmation web, de bases de données relationnelles, ainsi que sur la conception et l'implémentation d'un réseau local.

Le livre est subdivisé en trois parties :

PARTIE 1 : Programmer en JavaScript (chapitres 1 - 4)

PARTIE 2 : Initiation aux bases de données (chapitres 5 - 8)

PARTIE 3 : Les réseaux locaux (chapitres 9 - 11)

Ce livre n'est pas du tout un manuel d'utilisation de JavaScript, de SQL et de PHP respectivement d'un autre logiciel. Le livre se limite aux concepts importants en relation avec les prescriptions du référentiel de compétences du niveau sus-cité.

Dans les rubriques TP, seulement les manipulations les plus importantes sont mentionnées.

Première partie

PROGRAMMATION EN JAVASCRIPT

1

PRÉSENTATION DU LANGUAGE JAVASCRIPT



Sommaire

1.1	Introduction	4
1.2	Éditeurs JavaScript	4
1.3	Navigateurs compatibles JavaScript	5
1.4	Connaissance en HTML	5
1.5	Limites de JavaScript	5

1.1 Introduction

JavaScript est utilisé dans des millions de pages web afin d'améliorer leur conception. Il s'agit d'une couche de programmation supplémentaire qui vient s'ajouter au langage HTML. Le code HTML est le langage de base que toute page Internet se doit d'utiliser. Quant à JavaScript, il a été conçu pour enrichir, donner plus d'interactivité aux pages HTML. Jusqu'ici la programmation ne pouvait être exécutée que sur le serveur. La possibilité d'inclure des programmes dans les pages HTML et de les exécuter directement sur le poste client est intéressante, car elle permet de décharger le serveur de ce travail et d'éviter les attentes des réponses aux requêtes adressées via le Réseau.

Ce langage, comme le HTML, ne nécessite l'achat d'aucune licence pour pouvoir l'utiliser. JavaScript ne doit pas être confondu avec Java, qui est un langage beaucoup plus complexe permettant de créer des applications autonomes.

1.2 Éditeurs JavaScript

JavaScript a été créé par la société Netscape en 1995. À l'origine il se nommait LiveScript, puis il a été rebaptisé **JavaScript** par Netscape, intégré à Netscape Navigator pour la première fois en 1996 et standardisé par l'ECMA (European Computer Manufacturers Association) en 1997. Pour écrire ou modifier un script JavaScript on utilise un éditeur. Il existe de nombreux éditeurs standard et les éditeurs WYSIWYG(What You See Is What You Get).

1.2.1 Éditeurs Standard

Les éditeurs standard ne permettent pas de générer automatiquement le code JavaScript. On distingue :

- ✓ Notepad
- ✓ Simpletext

1.2.2 Éditeurs WYSIWYG

Les éditeurs WYSIWYG¹ capables de générer du code JavaScript automatiquement, on distingue :

- ✓ WebExpert
- ✓ JavaScript Editor

1. Pour "what you see is what you get" terme désignant les logiciels permettant à l'utilisateur de voir à l'écran ce qu'il obtiendrait sur le papier.

✓ ...

1.3 Navigateurs compatibles JavaScript

Initialement, JavaScript a été développé par Netscape, mais maintenant la plupart des explorateurs permettant de naviguer sur Internet sont compatibles avec JavaScript. Le navigateur ou browser interprète le code HTML et JavaScript, cette interprétation dépend naturellement du type de navigateur utilisé et de sa version. Voici une liste non exhaustive des navigateurs selon les systèmes d'exploitations suivants :

Linux / UNIX

- Firefox
- Netscape
- Mozilla
- Konqueror
- Lynx
-

Windows

- Internet Explorer
- Firefox
- Netscape
- Chrome
- Opéra
- Safari

MacOS

- Internet Explorer
- Konqueror
- Chrome
- Opéra
- Safari

Lynx est un navigateur qui n'interprète pas le JavaScript.

Le JavaScript est difficilement compatible entre les différents navigateurs. Il faut toujours se décider jusqu'à quel point ça doit être compatible.

1.4 Connaissance en HTML

Le code du JavaScript vient s'ajouter au code du langage Html, les deux langages partagent les mêmes éditeurs pour l'édition des codes et les mêmes navigateurs pour la présentation des informations. Le html est le langage de base auquel vient s'insérer le langage JavaScript ; une connaissance approfondie des balises ou tags Html est souhaitable sinon indispensable.

1.5 Limites de JavaScript

Cette section est destinée à définir précisément les possibilités et les limites du langage JavaScript.

1.5.1 Ce que permet JavaScript

Le langage JavaScript est parfaitement adapté pour des traitements sur le poste du navigateur. Grâce à JavaScript il est possible de créer des pages HTML dynamiques, prenant en compte des données de l'utilisateur (transmises au clavier ou par la souris), capturant et conservant ces données dans des variables ou des objets. Voici quelques exemples de cas où JavaScript est efficace :

- Vérification de saisie dans des formulaires
- Calculs simples suite à des saisies de formulaire (Tarifs, calculatrice)
- Gestion des dates et des heures (Date du jour, Date de mise à jour, Calendriers)
- Gestion de la navigation (Menus dynamique)
- Animations graphiques (MouseOver, bannières rotatives, jeux)
- ...

La liste n'est pas terminée. L'imagination du développeur peut repousser les limites du JavaScript, sans pour autant réaliser l'impossible. Voyons maintenant les raisons des limites techniques qu'imposent ce langage.

1.5.2 JavaScript et la sécurité

JavaScript a été conçu pour limiter les risques pour le visiteur. JavaScript ne permet pas d'écrire ou de lire un fichier sur le disque dur du visiteur ou sur le serveur. Il est impossible de détruire des fichiers indispensables ou d'introduire un virus sur le poste d'un visiteur. JavaScript permet quand même d'écrire et de lire le disque dur dans la zone réservée aux cookies. C'est la seule interaction que JavaScript peut avoir avec votre disque dur.

Le prix de cette sécurité et de cette quasi universalité se traduit par des limites techniques que nous allons détailler tout de suite.

1.5.3 Les limites de JavaScript

La principale limite de JavaScript est qu'il ne permet pas de lire et d'écrire sur le disque dur du visiteur et du serveur. De même, JavaScript ne peut pas s'interfacer avec une base de données. JavaScript n'échange pas avec d'autres machines connectées. Impossible donc de créer une page de chat (discussion en direct).

Les limites sont nombreuses et réelles. Heureusement, d'autres technologies existent.

1.5.4 Les alternatives

Il existe une multitude de technologies actuellement qui permettent de combler les lacunes de JavaScript. On peut les séparer en deux catégories :

1.5.4.1 Première catégorie

Dans cette catégorie, on peut citer Flash, Java, et d'autres qui permettent par exemple de créer des univers 3D. Ils repoussent les limites de l'animation graphique et offrent en général plus de possibilités de lecture/écriture sur le disque. La sécurité n'est pas assurée : plantages fréquents, accès au disque, ...

1.5.4.2 Deuxième catégorie

De nombreux langages de scripts sont exécutés sur le serveur, permettent ainsi de stocker des informations et de générer des pages dynamiques à partir de bases de données. La technologie PHP est recommandée à tous car les risques pour la sécurité sont nuls, et le visiteur ne reçoit que des pages en HTML, éventuellement enrichies par du JavaScript.



TAF : Identifier les éditeurs et les navigateurs installés dans votre poste de travail

Retenons

- JavaScript est un langage de programmation, créé par la société Netscape en 1995.
- Le code JavaScript est intégré complètement dans le code HTML.
- Un script est un petit programme inséré dans les pages HTML.
- JavaScript n'est pas Java.
- C'est le navigateur qui interprète le code JavaScript.



1. Ouvrir un éditeur de votre choix et saisir le code html suivant :

```
<html>
<head>
<h1>L'environnement de saisie du code</h1>
</head>
<body>
<p>Familiariser vous avec l'éditeur de script</p>
</body>
</html>
```

2. Lancer un navigateur de votre choix et exécuter le code ci-dessus.



1. Quel est le rôle de JavaScript dans une page web ?
2. Cocher la bonne réponse

Questions	Réponses
1. JavaScript est...	☐ Un langage de programmation ☐ Un navigateur ☐ Un langage internet
2. JavaScript rend les pages web dynamiques	☐ Vrai ☐ Faux
3. Java et JavaScript ont été créés par la même société	☐ Vrai ☐ Faux
4. Quelle société a inventé JavaScript	☐ Sun Microsystems ☐ Konqueror ☐ Netscape
5. Le code HTML s'intègre dans le script JavaScript	☐ Vrai ☐ Faux
6. Quel est l'intrus ?	☐ Mozilla ☐ Linux ☐ Internet Explorer
7. JavaScript est supporté par tous les navigateurs.	☐ Vrai ☐ Faux

2

STRUCTURE D'UN PROGRAMME JAVASCRIPT

```
<html>
<head>
<title>Titre du document</title>
<script language="javascript">
/*Instructions JavaScript*/
</script>
</head>
<body>
<!--code html-->
<script language="javascript">
/*Instructions JavaScript*/
</script>
<!--code html-->
</body>
</html>
```

Sommaire

2.1	Les balises <code><script></code> et <code></script></code>	10
2.2	Les commentaires	10
2.3	Où inclure le code en JavaScript dans le code html?	10
2.4	Les instructions	11
2.5	Le premier programme	12

2.1 Les balises `<script>` et `</script>`

Le langage html utilise des balises pour dire au navigateur d'afficher une portion de texte en gras, en italique, couleur, etc. Dans cette logique du langage html, il faut donc signaler au navigateur à travers la balise `<script>`, que ce qui suit est du code JavaScript. Cette balise a un paramètre **language** qui permet de préciser la nature du script, JavaScript dans ce cas. Cette balise s'écrit alors comme suit : `<script language="JavaScript">`. De même, il faudra informer le navigateur de la fin du script : C'est la balise `</script>`.

2.2 Les commentaires

Les commentaires permettent d'enrichir le script avec des explications. JavaScript utilise deux types de commentaire : Les commentaires sur une ligne sont indiqués à l'aide du symbole `//`. Tout ce qui est écrit entre ce symbole et la fin de la ligne n'est pas interprété par le navigateur.

Exemple : `//voici du commentaire sur une seule ligne.`

Les commentaires sur plusieurs lignes sont indiqués grâce aux symboles `/*` pour le début du commentaire et `*/` pour la fin du commentaire. Tout ce qui se trouve entre ce début et cette fin de commentaire est ignoré par le navigateur.

Exemple :

```
/* voici du commentaire
Sur plusieurs lignes
*/
```

2.3 Où inclure le code en JavaScript dans le code html ?

Le code JavaScript peut être placé dans l'entête (head) comme dans le corps (body) de la page html. Dans tous les cas, il est annoncé par la balise `<script language=javascript>`.

Lorsque le code JavaScript se situe dans l'entête de la page, les balises `<script language=javascript>` et `</script>` de début et fin de code JavaScript se trouvent entre les balises de début d'entête `<head>` et de fin d'entête `</head>`.

Une page html peut avoir plusieurs scripts JavaScript. Le script peut être placé dans l'entête comme dans le corps de la page. Un script JavaScript début par la balise la balise `<script>`. Cette balise contient le paramètre language qui permet de préciser le langage utilisé :

```
<script language="JavaScript">
```

Exemple :

```

<html>
<head>
<title>
<script language="javascript">
//Ce commentaire explique que ce script_en
JavaScript est place dans l'entête;
</script>
</title>
</head>
<body>
<!-- ce commentaire explique que cette partie du Code en html
concerne le body -->
</body>
</html>

```

Lorsque le code JavaScript est situé dans le corps de la page, ces balises de début et de fin se trouvent entre les balises <body> et </body>.

Exemple :

```

<html>
<head>
<title>
<!-- ce commentaire explique que cette partie du
Code_concerne l'entête en html -->
</title>
</head>
<body>
<script language="javascript">
/* Ce commentaire explique que ce script
en JavaScript est placé dans le corps de la page*/
</script>
</body>
</html>

```

Une page HTML peut avoir plusieurs scripts JavaScript en entête, dans le corps ou dans les deux parties en même temps. Le browser traite la page Html de haut en bas (y compris les ajouts en JavaScript). Par conséquent, toute instruction ne pourra être exécutée que si le browser possède à ce moment précis tous les éléments nécessaires à son exécution.

2.4 Les instructions

2.4.1 Une première instruction JavaScript

L'instruction **alert()** : **alert("votre texte");**

Cette instruction affiche un message (**votre texte**) dans une boite de dialogue pourvue d'un bouton **OK**.

Le texte à afficher par cette instruction doit être placé dans la double cotes : " et ". Une instruction JavaScript se termine par un point virgule (;), cependant, ce point virgule

est optionnel et il n'est obligatoire que lorsque plusieurs instructions sont écrites sur une même ligne.

2.5 Le premier programme

Pour pouvoir fonctionner, le code JavaScript d'une page html doit en principe être placé à l'intérieur d'une balise `<script>`. Une page html peut contenir plusieurs balises `<script>`, mais elles ne doivent pas être imbriquées. On peut placer une balise `<script>` soit dans l'entête, soit dans le corps de la page html. On placera de préférence une balise `<script>` contenant les procédures et les fonctions dans l'entête, car cela permet qu'elles soient chargées avant le reste de la page. Une page html intégrant les scripts JavaScript a la structure suivante :

```
<html>
<head>
<title>titre </title>
<script language='javascript'>
Instructions ; ... Instructions ;
</script>
</head>
<body>
<script language='javascript'>
Instructions ; ... Instructions ;
</script>
</body>
</html>
```

Activité 1 : Script placé dans le Head



Écrire un programme qui affiche d'abord dans une boîte de dialogue, le message "Que c'est merveilleux ! le monde du web !" ; avant d'afficher ensuite le reste d'information de la page html décrite ainsi : fond de couleur rose, écriture de couleur bleu.

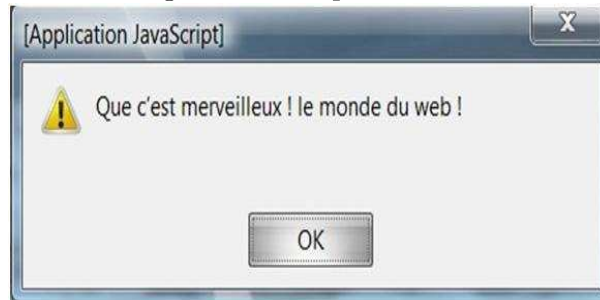
Corrigé de l'activité

```
<html>
<head>
<title>Activité1: Monde nouveau</title>
<script language='javascript'>
alert('Que c'est merveilleux !le monde du web !');
</script>
</head>
<body bgcolor='\#FF3300' text='\#0000CC'>
<p>Bonjour monde nouveau ! </p>
```

```
<p>comme c'est merveilleux de naviguer dans le web !!!</p>  
</body>  
</html>
```

Interprétation par un navigateur :

1. Message dans la boîte de dialogue JavaScript :



2. Appuyer sur OK
3. Page web obtenue



Retenons

- Les scripts JavaScript commencent par la balise `<script>` et se termine la balise de fin correspondante `</script>`
- On peut placer du code JavaScript dans une page html à trois endroits et sous des formes bien différentes.
- Toute déclaration JavaScript se termine par un point-virgule.
- JavaScript possède une liste de mots réservés.



1. Écrire un script qui affiche le message "Hello World !" sur la page.
2. Exécuter le script de la question 1) dans votre navigateur.
3. Que fait le script suivant ?

```
<html>
<head>
<title> My Page </title>
<script language='javascript'>
  alert('Bienvenue dans mon univers!');
</script>
</head>
<body bgcolor='\"#f33f00\"' text='\"#0f00CC\"'>
<p>Welcome! </p>
<p>Cher visiteur </p>
</body>
</html>
```



1. Quel est le rôle de la balise <script> ?
2. Présenter les différents types de commentaires dans JavaScript.
3. À quels endroit peut-on insérer les scripts JavaScript ?

3

SYNTAXE DU LANGAGE JAVASCRIPT

Sommaire

3.1	Les variables	16
3.2	les mots réservés	17
3.3	Les structures de contrôle	21
3.4	Les boucles	22

3.1 Les variables

Les variables contiennent des données qui peuvent être modifiées lors de l'exécution d'un programme. Une variable comporte un nom de variable qui permet de la référencer ; Un nom de variable doit commencer par une lettre de l'alphabet ou le signe `_` et se composer de lettres, de chiffres et des caractères `_` et `$` (à l'exclusion du blanc). Le nombre de caractères n'est pas précisé. JavaScript est sensible à la case. Donc le nom de variable écrit en majuscules et en minuscules sont différents.

3.1.1 La déclaration de variable

Les variables peuvent se déclarer de deux façons (explicitement et implicitement) :
La déclaration explicite de variable.

La commande qui permet de déclarer une variable de manière explicite est le mot `var`.

Exemples de déclaration explicite de variable :

```
var Numero = 1  
var Prenom = " Jean"
```

La première variable nommée `numero` contient la donnée 1 ; la seconde nommée contient la donnée Luc les deux noms sont précédés par le mot clé `var`.

La déclaration implicite de variable.

On écrit directement le nom de la variable suivi de la valeur que l'on lui attribue et Javascript s'en accomode.

Exemples de déclaration explicite de variable :

```
Numero = 1  
Prenom = "Jean"
```

La façon dont on déclare la variable aura une grande importance pour la "visibilité" de la variable dans le programme JavaScript.

Pour la clarté de votre script et votre facilité, on ne peut que conseiller d'utiliser à chaque fois le mot `var` pour déclarer une variable.

Variables globales et variables locales

Les variables déclarées tout au début du script, en dehors et avant toutes fonctions seront toujours globales, qu'elles soient déclarées avec `var` ou de façon contextuelle. On pourra donc les exploiter partout dans le script.

Dans une fonction, une variable déclarée par le mot clé `var` aura une portée limitée à cette seule fonction. On ne pourra donc pas l'exploiter ailleurs dans le script. D'où

son nom de locale. Par contre, toujours dans une fonction, si la variable est déclarée contextuellement (sans utiliser le mot var), sa portée sera globale.

3.2 les mots réservés

Les mots de la liste ci-après ne peuvent être utilisés pour des noms de fonctions et de variables. Certains de ces mots sont des mots clés JavaScript, d'autres ont été réservés par Netscape pour un futur usage éventuel.

A abstract	L long
B boolean, break, byte	N native, new, null
C case, catch, char, class, const, continue	P package, private, protected, public
D default, do, double	R return
E else, extends	S short, static, super, switch, synchronized
F false, final, finally, float, for, function	T this, throw, throws, transient, true try
G goto	V var, void
I if, implements, import, in, instanceof, int, interface	W while, with



JavaScript possède quelques règles et conventions syntaxique :

- **Sensible à la casse** : JavaScript est un langage sensible à la casse, autrement dit ce langage traite ces mots différemment : exemple, Exemple, EXEMPLE
- **Espace** : JavaScript, comme le html, ignore les espaces, les tabulations et le retour à la ligne appliqué dans une déclaration.



TAF : Écrire un script JavaScript dans lequel les variables initialisées aux valeurs suivantes sont déclarées :

1. la valeur 8
2. le double de la première variable

3.2.1 Notion d'objet

Le Javascript traite les éléments qui s'affichent dans votre navigateur comme des objets, c'est-à-dire des éléments :

- classés selon une hiérarchie pour pouvoir les désigner précisément ;
- auxquels des propriétés sont associées.

Cette notion semble floue pour l'instant mais voyons voir cela sur un exemple concret : Imaginez un arbre dans un jardin comportant une branche sur laquelle se trouve un nid. On suppose la hiérarchie d'objets est définie comme ceci :

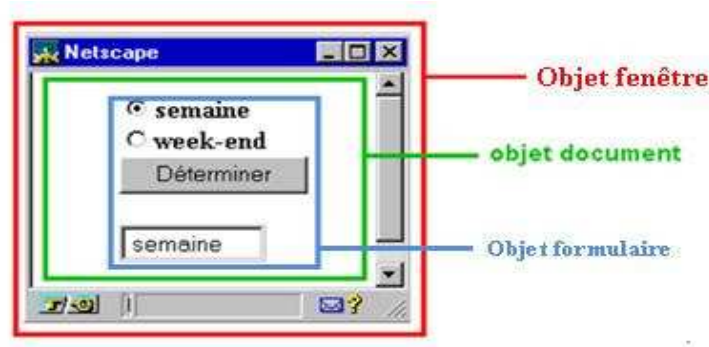
```
⊙jardin
→ arbre
  ▷ branche
    - feuille
    - nid
      - largeur : 20
      - couleur : jaune
      - hauteur : 4
  ▷ tronc
  ▷ racine
→ salade
→ balançoire
  - trapèze
  - corde
  - nid
    - largeur : 15
    - couleur : marron
    - hauteur : 6
```

Le nid sur l'arbre est donc désigné comme suit : **jardin.arbre.branche.nid** Contrairement au nid situé sur la balançoire : **jardin.balançoire.nid**

Pour changer la couleur du nid (dans l'arbre) et le peindre en vert, il suffit d'une commande telle que la suivante : **jardin.arbre.branche.nid.couleur= vert** ;

Les objets en JavaScript sont représentés de cette manière à ceci près que ce n'est pas un jardin qui est représenté sous forme d'objets mais la fenêtre du navigateur.

3.2.2 Les objets du navigateur



Cette illustration est une page web affichée dans la fenêtre du navigateur Netscape. C'est l'objet **fenêtre** de Windows aussi appelé **objet Windows**.

Dans cette fenêtre, il y a un document Html. C'est l'objet **document**. Autrement dit l'objet fenêtre contient l'objet document.

Dans ce document, on trouve un formulaire au sens Html. C'est l'objet **formulaire**.

On note donc une hiérarchie entre les objets du navigateur qui détermine l'accès à ces objets. Par exemple, pour accéder à l'objet formulaire, on procède ainsi :

Windows.document.formulaire

Pour accéder à l'objet bouton, on écrit :

Windows.document.formulaire.bouton

Dans cette nomenclature, Windows est facultatif. L'accès à l'objet formulaire devient : **document.formulaire** ;

Et celle à l'objet bouton devient : **document.formulaire.bouton**.

D'autres objets Windows existent, c'est le cas de l'instruction **alert** vu plus haut. Cette instruction est un objet Windows qui permet d'afficher une information dans une boîte de dialogue. L'accès à cette méthode selon le même principe d'hiérarchie entre objets : **Windows.alert** ou **alert** étant donnée que l'objet Windows est facultatif.



TAF : Écrire un script JavaScript qui modifie la largeur et la hauteur du trapèze précédent respectivement avec les valeurs 20 et 12

3.2.3 Les opérateurs

Les opérateurs permettent de manipuler et d'évaluer les variables. JavaScript distingue différents types d'opérateurs à savoir :

- Les opérateurs d'affectation,
- Les opérateurs arithmétiques,
- Les opérateurs logiques,
- Les opérateurs de comparaison,
- Les opérateurs spéciaux.

Les opérateurs d'affectation

Une affectation permet de donner la valeur d'une expression à une variable. L'opérateur de base est `=`. Exemple : $x = y + 3$

Les opérateurs arithmétiques

On trouve les opérateurs classiques `+`, `-`, `/` et `*`. L'opérateur de modulo calcule le reste de la division entière de deux nombres il se note `%`.

Les opérateurs logiques

`&&` c'est le «et» logique

`||` le «ou» logique

`!` le NON logique

Les opérateurs de comparaison

`==` teste l'égalité `!=` teste l'inégalité

`>` plus grand que

`<` plus petit que

`>=` plus grand ou égal

`<=` plus petit ou égal

Les opérateurs spéciaux

new permet de créer un objet.

void permet de spécifier une expression à évaluer sans qu'elle ne retourne de valeur.
Syntaxe : JavaScript :**void**(expression).

new permet de créer un objet.

void permet de spécifier une expression à évaluer sans qu'elle ne retourne de valeur.
Syntaxe : JavaScript :**void**(expression).

3.3 Les structures de contrôle

Qu'est-ce qu'une structure conditionnelle ?

On appelle structure conditionnelle les instructions qui permettent de tester si une condition est vraie ou non, ce qui permet notamment de donner de l'interactivité à vos scripts.

3.3.1 L'instruction if

L'instruction **if** est la structure de test la plus basique, elle permet d'exécuter une série d'instructions (un bloc d'instructions) si jamais une condition est réalisée.

La syntaxe de cette expression est la suivante :

```
if (condition réalisée) {  
    liste d'instructions  
}
```



- Il est possible de définir plusieurs conditions à remplir avec les opérateurs ET et OU (&& et ||)
 - S'il n'y a qu'une instruction dans le bloc d'instruction, les accolades ne sont pas nécessaires...
- ⇒ Exemple : `if (x == 2) document.write("x vaut 2");`

3.3.2 L'instruction **if ... else**

L'instruction **if** dans sa forme basique ne permet de tester qu'une condition, or la plupart du temps on aimerait pouvoir choisir les instructions à exécuter en cas de non réalisation de la condition... L'expression **if ... else** permet d'exécuter une autre série d'instruction en cas de non-réalisation de la condition.

La syntaxe de cette expression est la suivante :

```
if (condition réalisée)
{
    //liste d'instructions
}
else
{
    //autre série d'instructions
}
```

3.4 Les boucles

Les boucles sont des structures qui permettent d'exécuter plusieurs fois la même série d'instructions jusqu'à ce qu'une condition ne soit plus réalisée... La façon la plus commune de faire une boucle, est de créer un compteur (une variable qui s'incrémente, c'est-à-dire qui augmente à chaque tour de boucle) et de faire arrêter la boucle lorsque le compteur dépasse une certaine valeur.

3.4.1 La boucle **for**

L'instruction **for** permet d'exécuter plusieurs fois la même série d'instructions : c'est une boucle !

Dans sa syntaxe, il suffit de préciser le nom de la variable qui sert de compteur (et éventuellement sa valeur de départ, la condition sur la variable pour laquelle la boucle s'arrête (basiquement une condition qui teste si la valeur du compteur dépasse une limite) et enfin une instruction qui incrémente (ou décrémente) le compteur.

La syntaxe de cette expression est la suivante :

```
for (compteur; condition; modification du compteur)
{
    liste d'instructions
}
```

Par exemple :

```
for (i=1; i<6; i++)
{
    alert(i)
}
```

Cette boucle affiche 5 fois la valeur de i , c'est-à-dire 1,2,3,4,5 Elle commence à $i=1$, vérifie que i est bien inférieur à 6, etc... jusqu'à atteindre la valeur $i=6$, pour laquelle la condition ne sera plus réalisée, la boucle s'interrompt et le programme continuera son cours.



- Il faudra toujours vérifier que la boucle a bien une condition de sortie (i.e le compteur s'incrémente correctement)
- Une instruction *alert(i)* ; dans votre boucle est un bon moyen pour vérifier la valeur du compteur pas à pas !
- Il faut bien compter le nombre de fois que l'on veut faire exécuter la boucle :
 - *for(i = 0; i < 10; i++)* exécute 10 fois la boucle (i de 0 à 9)
 - *for(i = 0; i <= 10; i++)* exécute 11 fois la boucle (i de 0 à 10)
 - *for(i = 1; i < 10; i++)* exécute 9 fois la boucle (i de 1 à 9)
 - *for(i = 1; i <= 10; i++)* exécute 10 fois la boucle (i de 1 à 10)

3.4.2 L'instruction while

L'instruction **while** représente un autre moyen d'exécuter plusieurs fois la même série d'instructions.

La syntaxe de cette expression est la suivante :

```
while (condition réalisée)
{
    liste d'instructions
}
```

Cette instruction exécute la liste d'instructions tant que (**while** est un mot anglais qui signifie tant que) la condition est réalisée.

3.4.3 Saut inconditionnel

Il peut être nécessaire de faire sauter à la boucle une ou plusieurs valeurs sans pour autant mettre fin à celle-ci.

La syntaxe de cette expression est "continue;" (cette instruction se place dans une boucle!), on l'associe généralement à une structure conditionnelle, sinon les lignes situées entre cette instruction et la fin de la boucle seraient obsolètes.

Exemple : Imaginons que l'on veuille imprimer pour x allant de 1 à 10 la valeur de $1/(x-7) \dots$ il est évident que pour $x = 7$ il y aura une erreur. Heureusement, grâce à l'instruction continue il est possible de traiter cette valeur à part puis de continuer la

boucle !

```
x=1;
while (x<=10) {
  if (x == 7)
  {
    alert('division par 0');
    continue;
  }
  a = 1/(x-7);
  alert(a);
  x++;
}
```

Il y avait une erreur dans ce script... peut-être ne l'avez-vous pas vue : Lorsque x est égal à 7, le compteur ne s'incrémente plus, il reste constamment à la valeur 7, il aurait fallu écrire :

```
x=1;
while(x<=10) {
  if (x == 7)
  {
    alert('division par 0');
    x++;
    continue;
  }
  a = 1/(x-7);
  alert(a);
  x++;
}
```

4

INTERACTIVITÉ DE JAVASCRIPT

☒ <u>Congés Sans Solde</u>	Du : 01 - Lundi 02 - Mardi 03 - Mercredi 04 - Jeudi 05 - Vendredi (inclus)	Au : (inclus)	Soit : ze jours	
☐ <u>RTT</u>	Motif :	Du : 08 - Lundi 09 - Mardi 10 - Mercredi 11 - Jeudi (inclus)	Au : (inclus)	Soit : jours

Sommaire

4.1	Les événements	26
4.2	Les boîtes de dialogue	28
4.3	Les formulaires	29
4.4	Les fonctions	31

4.1 Les événements

4.1.1 Qu'appelle-t-on un événement ?

Les événements sont des actions de l'utilisateur, qui vont pouvoir donner lieu à une interactivité. L'événement par excellence est le clic de souris, car c'est le seul que le HTML gère. Grâce au JavaScript il est possible d'associer des fonctions, des méthodes à des événements tels que le passage de la souris au-dessus d'une zone, le changement d'une valeur, ...

Ce sont les gestionnaires d'événements qui permettent d'associer une action à un événement.

Les gestionnaires d'événements sont associés à des objets, et leur code s'insèrent dans la balise de ceux-ci...

4.1.2 Liste des événements

- Abort(**onAbort**) : Cet événement a lieu lorsque l'utilisateur interrompt le chargement de l'image
- Change(**onChange**) Se produit lorsque l'utilisateur modifie le contenu d'un champ de données.
- Click(**onClick**) Se produit lorsque l'utilisateur clique sur l'élément associé à l'événement.
- dblclick(**ondblclick**) Se produit lorsque l'utilisateur double-clique sur l'élément associé à l'événement (un lien hypertexte ou un élément de formulaire).
- dragdrop(**ondragdrop**) Se produit lorsque l'utilisateur effectue un glisser-déposer sur la fenêtre du navigateur.
- error (**onError**) Se déclenche lorsqu'une erreur apparaît durant le chargement de la page.
- Focus(**onFocus**) Se produit lorsque l'utilisateur donne le focus à un élément, c'est-à-dire que cet élément est sélectionné comme étant l'élément actif
- keydown(**onkeydown**) Se produit lorsque l'utilisateur appuie sur une touche de son clavier.
- keypress(**onkeypress**) Se produit lorsque l'utilisateur maintient une touche de son clavier enfoncée.
- keyup(**onkeyup**) Se produit lorsque l'utilisateur relâche une touche de son clavier préalablement enfoncée.
- Load(**onLoad**) Se produit lorsque le navigateur de l'utilisateur charge la page en cours
- MouseOver(**onmouseover**) Se produit lorsque l'utilisateur positionne le curseur de la souris au-dessus d'un élément
- MouseOut(**onmouseout**) Se produit lorsque le curseur de la souris quitte un élément.

- Resize(**onResize**) Se produit lorsque l'utilisateur redimensionne la fenêtre du navigateur
- Select(**onSelect**) Se produit lorsque l'utilisateur sélectionne un texte dans un champ de type "text"
- Submit(**onSubmit**) Se produit lorsque l'utilisateur clique sur le bouton de soumission d'un formulaire (le bouton qui permet d'envoyer le formulaire)
- Unload(**onUnload**) Se produit lorsque le navigateur de l'utilisateur quitte la page en cours

4.1.3 Association des événements aux objets

Chaque événement ne peut pas être associé à n'importe quel objet. Il est évident par exemple qu'un événement **OnChange** ne pourra pas s'appliquer à un lien hypertexte. Voici un tableau récapitulant les objets auxquels peuvent être associés chaque événement :

Événements	Objets concernés
abort	Image
change	FileUpload, Select, Submit, Text, TextArea
error	Image, window
click	Button, document, Checkbox, Link, Reset, Select, Submit
load	Image, Layer, window
mouseover	Area, Layer, Link



Le code correspondant à une boîte de dialogue est très simple :

```
<html>
<head>
<title>Ouverture d'une boîte de dialogue lors d'un
click</title>
</head>
<body>
<a href="javascript:;" onClick="window.alert('Message_d\'alerte_à
utiliser_avec_moderation');"> Cliquez ici!</a>
</body>
</html>
```

Analyse du script :

- le gestionnaire d'événement onClick a été inséré dans la balise de lien hypertexte <A href...
- le seul but du script est de faire apparaître une boîte de dialogue, ainsi on ne désire pas que le lien nous entraîne sur une autre page, il faut alors insérer "javascript :;" dans l'attribut href pour signaler au navigateur que l'on désire rester sur la page en

cours. Il ne faut pas mettre un attribut vide au risque de révéler le contenu de votre répertoire à vos visiteurs...

- Remarquez l'emploi de `document.write()` dans la phrase "Message d'alerte a utiliser avec moderation"

Aperçu de l'effet du script :



4.2 Les boîtes de dialogue

Javascript met à votre disposition 3 boîtes de message :

- `alert()`
- `prompt()`
- `confirm()`

Ce sont toutes trois des méthodes de l'objet `window`.

4.2.1 La méthode `alert()`

La méthode **`alert()`** doit, à ce stade de votre étude, vous être familière car nous l'avons déjà souvent utilisée. La méthode **`alert()`** affiche une boîte de dialogue dans laquelle est reproduite la valeur (variable et/ou chaîne de caractères) de l'argument qui lui a été fourni. Cette boîte bloque le programme en cours tant que l'utilisateur n'aura pas cliqué sur "OK". **`Alert()`** sera aussi très utile pour vous aider à débbuger les scripts. Sa syntaxe est :

- `alert(variable)`,
- `alert("chaîne de caractères")`,
- `alert(variable + "chaîne de caractères")`.

4.2.1.1 La méthode `prompt()`

Dans le même style que la méthode `alert()`, Javascript vous propose une autre boîte de dialogue, dans le cas présent appelée boîte d'invite, qui est composée d'un champ comportant une entrée à compléter par l'utilisateur. Cette entrée possède aussi une valeur par défaut. La syntaxe est :

```
prompt("texte de la boîte d'invite", "valeur par défaut");
```

En cliquant sur OK, la méthode renvoie la valeur tapée par l'utilisateur ou la réponse proposée par défaut. Si l'utilisateur clique sur Annuler ou Cancel, la valeur null est alors renvoyée. Prompt() est parfois utilisé pour saisir des données fournies par l'utilisateur. Selon certaines sources, le texte ne doit cependant pas dépasser 45 caractères sous Netscape et 38 sous Explorer.

4.2.2 La méthode confirm()

Cette méthode affiche 2 boutons "OK" et "Annuler". En cliquant sur OK, continue() renvoie la valeur true et bien entendu false si on a cliqué sur Annuler. Ce qui peut permettre, par exemple, de choisir une option dans un programme.

La syntaxe de l'exemple est : confirm("Voulez-vous continuer?")

4.2.2.1 Une minuterie

Javascript met à votre disposition une minuterie (ou plus précisément un compteur à rebours) qui permettra de déclencher une fonction après un laps de temps déterminé.

La syntaxe de mise en route du temporisateur est :

nom_{du}compteur = setTimeout("fonction_{appele}()", temps en milliseconde) Ainsi, setTimeout("demarrer()",5000) va lancer la fonction demarrer() après 5 secondes. Pour arrêter le temporisateur avant l'expiration du délai fixé, il y a : clearTimeout(*nom_{du}compteur*)

4.3 Les formulaires

4.3.1 Généralités

Avec Javascript, les formulaires Html prennent une toute autre dimension. N'oublions pas qu'en Javascript, on peut accéder à chaque élément d'un formulaire pour, par exemple, y aller lire ou écrire une valeur, noter un choix auquel on pourra associer un gestionnaire d'événement... Tous ces éléments renforceront grandement les capacités interactives de vos pages.

Mettons au point le vocabulaire que nous utiliserons. Un formulaire est l'élément Html déclaré par les balises <form></form>. Un formulaire contient un ou plusieurs éléments que nous appellerons des contrôles (widgets). Ces contrôles sont notés par exemple par la balise <input type= ...>.

4.3.2 Déclaration d'un formulaire

La déclaration d'un formulaire se fait par les balises `<form>` et `</form>`. Il faut noter qu'en Javascript, l'attribut `name="nom_du_formulaire"` a toute son importance pour désigner le chemin complet des éléments. En outre, les attributs `ACTION` et `METHOD` sont facultatifs pour autant que vous ne faites pas appel au serveur.

Une erreur classique en Javascript est, emporté par son élan, d'oublier de déclarer la fin du formulaire `</form>` après avoir incorporé un contrôle.

4.3.2.1 Lire une valeur dans une zone de texte

Voici un exemple que nous détaillerons :

Voici une zone de texte. Entrez une valeur et appuyer sur le bouton pour contrôler celle-ci.



The image shows a graphical representation of a web form. It consists of a rectangular text input field at the top, and a button with the text 'Contrôler' centered below it.

Le script complet est celui-ci :

```
<html>
<head>
<script language="javascript">
function controle(form1) { var test = document.form1.input.value;
alert("Vous_avez_tapé:_ " + test);
}
</script>
</head>
<body>
<form name="form1">
<input type="text" name="input" value=""><br>
<input type="button" name="bouton" value="contrôler" onclick="controle(form1)">
</form>
</body>
</html>
```

Lorsqu'on clique le bouton "contrôler", Javascript appelle la fonction `controle()` à laquelle on passe le formulaire dont le nom est `form1` comme argument.

Cette fonction `controle()` définie dans les balises `<HEAD>` prend sous la variable `test`, la valeur de la zone de texte. Pour accéder à cette valeur, on note le chemin complet de celle-ci (voir le chapitre "Un peu de théorie objet"). Soit dans le document présent, il y a l'objet formulaire appelé `form1` qui contient le contrôle de texte nommé `input` et qui a comme propriété l'élément de valeur `value`. Ce qui donne `document.form1.input.value`.

4.3.2.2 Écrire une valeur dans une zone de texte

Entrez une valeur quelconque dans la zone de texte d'entrée. Appuyer sur le bouton pour afficher cette valeur dans la zone de texte de sortie.

The diagram shows a web form layout. At the top is a rectangular input field labeled 'Zone de texte d'entrée' to its right. Below the input field is a button labeled 'Afficher'. Below the button is another rectangular field labeled 'Zone de texte de sortie' to its right.

Voici le code :

```
<html>
<head>
<script language="javascript">
function afficher(form2) {
var testin =document. form2.input.value;
document.form2.output.value=testin
}
</script>
</head>
<body>
<form name="form2">
<input type="text" name="input" value=""> zone de texte d'entrée <br>
<input type="button" name="bouton" value="afficher" onclick="afficher(form2)"><br>
<input type="text" name="output" value=""> zone de texte de sortie
</form>
</body>
</html>
```

Lorsqu'on clique le bouton "Afficher", Javascript appelle la fonction `afficher()` à laquelle on passe le formulaire dont le nom est cette fois `form2` comme argument.

Cette fonction `afficher()` définie dans les balises `<HEAD>` prend sous la variable `testin`, la valeur de la zone de texte d'entrée. A l'instruction suivante, on dit à Javascript que la valeur de la zone de texte `output` comprise dans le formulaire nommé `form2` est celle de la variable `testin`. A nouveau, on a utilisé le chemin complet pour arriver à la propriété `valeur` de l'objet souhaité soit en Javascript `document.form2.output.value`.

4.4 Les fonctions

4.4.1 Définition

Une fonction est un groupe de ligne(s) de code de programmation destiné à exécuter une tâche bien spécifique et que l'on pourra, si besoin est, utiliser à plusieurs reprises. De plus, l'usage des fonctions améliorera grandement la lisibilité de votre script. En Javascript, il existe deux types de fonctions :

- les fonctions propres à JavaScript. On les appelle des "méthodes". Elles sont associées à un objet bien particulier comme c'était le cas de la méthode `Alert()` avec l'objet `window`.
- les fonctions écrites par vous-même pour les besoins de votre script. C'est à celles-là que nous nous intéressons maintenant.

4.4.1.1 Les boutons case à cocher (checkbox)

Les boutons case à cocher sont utilisés pour noter un ou plusieurs choix (pour rappel avec les boutons radio un seul choix) parmi un ensemble de propositions. A part cela, sa syntaxe et son usage est tout à fait semblable aux boutons radio sauf en ce qui concerne l'attribut `name`.

Ses propriétés sont : `name`, `checked`, `defaultchecked` et `value`.

Entrez votre choix :

Il faut sélectionner les numéros 1, 2 et 4 pour avoir la bonne réponse.

The image shows a web form with four checkboxes stacked vertically, each followed by the text 'Choix numéro 1', 'Choix numéro 2', 'Choix numéro 3', and 'Choix numéro 4' respectively. Below the checkboxes is a button labeled 'Corriger'.

```
<html>
<head>
<script language="javascript">
function reponse(form4) {
if ( (form4.check1.checked) == true && (form4.check2.checked) == true
&& (form4.check3.checked) == false
&& (form4.check4.checked) == true)
{ alert("C'est la bonne réponse!") }
else
{ alert("Désolé, continuez à chercher.") }
}
</script>
</head>
<body>
Entrez votre choix :
<form name="form4">
<input type="checkbox" name="check1" value="1">Choix numéro 1<br>
<input type="checkbox" name="check2" value="2">Choix numéro 2<br>
<input type="checkbox" name="check3" value="3">Choix numéro 3<br>
<input type="checkbox" name="check4" value="4">Choix numéro 4<br>
<input type="button" name="but" value="Corriger" onClick="reponse(form4)">
</form>
</body>
</html>
```

4.4.2 Déclaration des fonctions

Pour déclarer ou définir une fonction, on utilise le mot (réservé) **function**. La syntaxe d'une déclaration de fonction est la suivante :

```
function nom_de_la_fonction(arguments)
{ ... code des
instructions ... }
```

Le nom de la fonction suit les mêmes règles que celles qui régissent le nom de variables (nombre de caractères indéfini, commencer par une lettre, peuvent inclure des chiffres...).

Pour rappel, JavaScript est sensible à la case. Ainsi `fonction()` ne sera pas égal à `Fonction()`. En outre, Tous les noms des fonctions dans un script doivent être uniques.

La mention des arguments est facultative mais dans ce cas les parenthèses doivent rester. C'est d'ailleurs grâce à ces parenthèses que l'interpréteur JavaScript distingue les variables des fonctions. Nous reviendrons plus en détail sur les arguments et autres paramètres dans la partie JavaScript avancé.

Lorsque une accolade est ouverte, elle doit impérativement, sous peine de message d'erreur, être refermée. Prenez la bonne habitude de fermer directement vos accolades et d'écrire votre code entre elles.

Le fait de définir une fonction n'entraîne pas l'exécution des commandes qui la composent. Ce n'est que lors de l'appel de la fonction que le code de programme est exécuté.

4.4.3 L'appel d'une fonction

L'appel d'une fonction se fait le plus simplement du monde par le nom de la fonction (avec les parenthèses). Soit par exemple `nom_de_la_fonction()` ;

Il faudra veiller en toute logique (car l'interpréteur lit votre script de haut vers le bas) que votre fonction soit bien définie avant d'être appelée.

4.4.4 Les fonctions dans <head>...</head>

Il est donc prudent ou judicieux de placer toutes les déclarations de fonction dans l'entête de votre page c-à-d .dans la balise <head> ... </head>. Vous serez ainsi assuré que vos fonctions seront déjà prises en compte par l'interpréteur avant qu'elles soient appelées dans le <body>.

4.4.5 Exemple

Dans cet exemple, on définit dans les balises head, une fonction appelée **message()** qui affiche le texte "Bienvenue à ma page". cette fonction sera appelée au chargement de la page voir **onLoad=....** dans le tag <body>.

```
<html>
<head>
<script language="Javascript">
<— function
message() { document.write("Bienvenue à ma page"); }
//—>
</script>
</head>
<body onLoad="message()">
</body>
</html>
```

4.4.6 Passer une valeur à une fonction

On peut passer des valeurs ou paramètres aux fonctions Javascript. La valeur ainsi passée sera utilisée par la fonction. Pour passer un paramètre à une fonction, on fournit un nom d'une variable dans la déclaration de la fonction.

Un exemple un peu simplet pour comprendre. J'écris une fonction qui affiche une boîte d'alerte dont le texte peut changer.

Dans la déclaration de la fonction, on écrit :

```
function Exemple(Texte) {
  alert(texte);
}
```

Le nom de la variable est Texte et est définie comme un paramètre de la fonction. Dans l'appel de la fonction, on lui fournit le texte : **Exemple("Salut à tous")**;

4.4.7 Passer plusieurs valeurs à une fonction

On peut passer plusieurs paramètres à une fonction. Comme c'est souvent le cas en JavaScript, on sépare les paramètres par des virgules.

```
function $nom_de_la_fonction$(arg1, arg2, arg3) {
  ...
  code des instructions
  ...
}
```

Notre premier exemple devient pour la déclaration de fonction :

```
function Exemplebis(Texte1, Texte2){...}
```

et pour l'appel de la fonction `Exemplebis("Salut à tous", "Signé Luc")`

4.4.8 Retourner une valeur

Le principe est simple (la pratique parfois moins). Pour renvoyer un résultat, il suffit d'écrire le mot clé **return** suivi de l'expression à renvoyer. Notez qu'il ne faut pas entourer l'expression de parenthèses. Par exemple :

```
function cube(nombre) {
  var cube = nombre*nombre*nombre
  return cube;
}
```

Précisons que l'instruction `return` est facultative et qu'on peut trouver plusieurs `return` dans une même fonction. Pour exploiter cette valeur de la variable retournée par la fonction, on utilise une formulation du type **document.write(cube(5))**.

4.4.9 Variables locales et variables globales

Avec les fonctions, le bon usage des variables locales et globales prend toute son importance.

Une variable déclarée dans une fonction par le mot clé **var** aura une portée limitée à cette seule fonction. On ne pourra donc pas l'exploiter ailleurs dans le script. On l'appelle donc **variable locale**.

```
function cube(nombre) {
  var cube = nombre*nombre*nombre
}
```

Ainsi la variable `cube` dans cet exemple est une variable locale. Si vous y faites référence ailleurs dans le script, cette variable sera inconnue pour l'interpréteur JavaScript (message d'erreur).

Si la variable est déclarée contextuellement (sans utiliser le mot `var`), sa portée sera **globale** – et pour être tout à fait précis, une fois que la fonction aura été exécutée–.

```
function cube(nombre) {
  cube = nombre*nombre*nombre
}
```

La variable `cube` déclarée contextuellement sera ici une variable globale.

Les variables déclarées tout au début du script, en dehors et avant toutes fonctions, seront toujours globales, qu'elles soient déclarées avec **var** ou de façon contextuelle.

```
<script language="javascript">
var cube=1 function cube(nombre) {
var cube = nombre*nombre*nombre
```

```
}  
</script>
```

La variable `cube` sera bien globale. Pour la facilité de gestion des variables, on ne peut que conseiller de les déclarer en début de script (comme dans la plupart des langages de programmation). Cette habitude vous met à l'abri de certaines complications.

Deuxième partie

INITIATION AUX BASES DE DONNÉES

5

INTRODUCTION AUX SGBD



Sommaire

5.1	Introduction	39
5.2	La gestion des bases de données	39
5.3	Les principaux SGBD	39
5.4	Étude du SGBD ACCESS	40

5.1 Introduction

Un Système de Gestion de Bases de Données (S.G.B.D.) représente un ensemble coordonné de logiciels qui permet de décrire, manipuler, traiter les ensembles de données formant la base. Il doit également assurer la sécurité et la confidentialité des données dans un environnement où de nombreux utilisateurs ayant des besoins variés peuvent interagir simultanément sur ces données. Il doit pouvoir être utilisé par des non-informaticiens. Il doit assurer la définition des structures de stockage et des structures de données et le suivi de leurs évolutions ; c'est ce qu'on appelle l'administration des données. Il doit pouvoir au maximum vérifier la cohérence des données. Le SGBD sert donc d'interface entre les programmes d'application des utilisateurs d'une part, et la base de données d'autre part. Microsoft Access est un SGBD de type relationnel.

5.2 La gestion des bases de données

Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion de la base de données se fait grâce à un système appelé SGBD (système de gestion de bases de données). Le SGBD est un ensemble de services permettant de gérer les bases de données, c'est-à-dire :

- Permettre l'accès aux données de façon simple
- Autoriser un accès aux informations à de multiples utilisateurs
- Manipuler les données présentes dans la base de données (insertion, suppression, modification)

Le SGBD peut se décomposer en trois sous-systèmes :

- Le système de gestion de fichiers : il permet le stockage des informations sur un support physique
- Le SGBD interne : il gère l'ordonnancement des informations
- Le SGBD externe : il représente l'interface avec l'utilisateur

5.3 Les principaux SGBD

Les principaux systèmes de gestion de bases de données sont les suivants :

- Microsoft Access
- Microsoft SQL server
- MySQL
- Oracle
- SQL Server

5.4 Étude du SGBD ACCESS

5.4.1 Interface MS ACCESS

Lorsque vous ouvrez ACCESS, une boîte de dialogue (fig. 1) s'affiche vous proposant de créer une nouvelle base de données ou d'en ouvrir une existante. Une base de données ACCESS est un fichier (extension .MDB) qui inclue tous les composants de la base de données : tables, requêtes, formulaires, rapports.

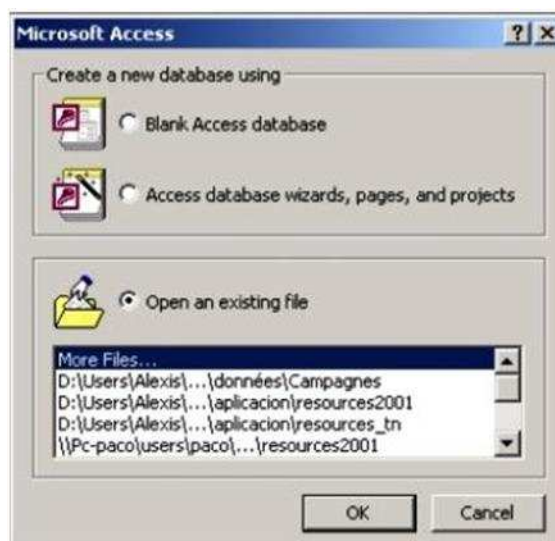


FIGURE 5.1 – Boîte de dialogue ouverture

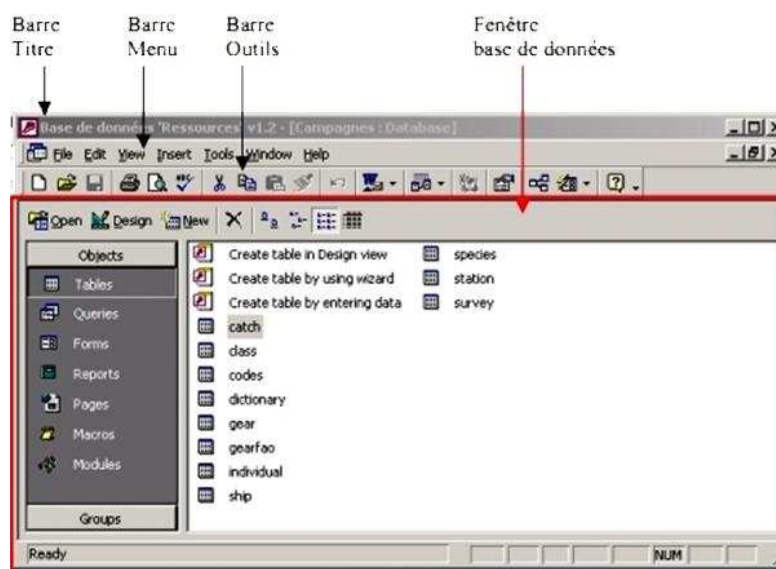


FIGURE 5.2 – Interface d'accueil

5.4.2 Les Composants d'une base de données ACCESS

La fenêtre Base de Données contient tous les «objets» qui peuvent composer une base Access, classés par catégorie :

- Tables,
- Requêtes,
- Formulaires,
- États.

Pour accéder à un objet, il faut cliquer sur le bouton correspondant à sa catégorie.

5.4.2.1 Tables

Point central de la base de données, car toutes les données vont être stockées dans des tables. La performance de la base de données repose sur la structuration des tables selon un modèle de données. Chaque table est identifiée par son nom.

Pour ouvrir une table, on peut la sélectionner et utiliser le bouton d'ouverture. On peut aussi l'ouvrir par double-click sur son nom.

Le nom des attributs sont portés en titre des colonnes. Le nombre total d'enregistrements (de lignes) ainsi que la position de l'enregistrement courant sont indiqués en bas dans la barre système.

srv_code	sta_id	date	start lat	start long	start time	start bath
DEMSUD2	1	23/5/00	34,424	11,0866666666667	2.00	2
DEMSUD2	2	22/5/00	34,5181666666667	11,2105	11.40	3
DEMSUD2	3	22/5/00	34,5821666666667	11,361	20.20	2
DEMSUD2	4	22/5/00	34,6013333333333	11,3463333333333	19.00	2
DEMSUD2	5	22/5/00	34,5045	11,3148333333333	21.55	4
DEMSUD2	6	23/5/00	34,458	11,1876666666667	1.15	4
DEMSUD2	7	23/5/00	34,3856666666667	11,009	4.55	3
DEMSUD2	8	27/5/00	34,3983333333333	10,5565	18.55	2
DEMSUD2	10	27/5/00	34,4516666666667	10,5	20.45	2
DEMSUD2	12	25/5/00	34,2293333333333	10,7705	3.00	3
DEMSUD2	13	29/5/00	34,2691666666667	10,6573333333333	17.40	3
DEMSUD2	14	27/5/00	34,2286666666667	10,503	23.30	2
DEMSUD2	16	28/5/00	34,1305	10,4706666666667	0.45	2
DEMSUD2	17	27/5/00	34,3458333333333	10,4696666666667	22.00	2
DEMSUD2	18	28/5/00	34,1358333333333	10,4016666666667	2.05	2

FIGURE 5.3 – Exemple de table ouverte

5.4.2.2 Requêtes

Les requêtes vont être utilisées pour obtenir des données contenues dans une ou plusieurs tables à partir d'une question. Par exemple, une requête va pouvoir nous afficher la liste des clients habitant à Paris.

Pour accéder à un objet requête, sélectionner le bouton Requêtes. Chaque requête est identifiée par un nom. Comme pour une table, l'ouverture d'une requête se fait au moyen du bouton Ouvrir ou par double-click sur le nom de la requête.

5.4.2.3 Formulaires

Le formulaire est utilisé pour faciliter la saisie et la modification des données d'une table, le formulaire permet d'entrer de façon conviviale les informations concernant une ou plusieurs tables. On peut les saisir sans formulaire, ce que nous verrons, mais cette méthode est moins conviviale.

5.4.2.4 États

Les états permettent d'imprimer des données contenues dans des tables selon une présentation définie, en y intégrant éventuellement des calculs.

5.4.3 Structuration d'une base de données dans ACCESS : les tables

5.4.3.1 Construire une table

Access propose 5 méthodes pour créer une table :

Mode feuilles de données : ce mode permet de saisir directement des informations dans la table sans se préoccuper du nom ou du format des champs, la définition du nom et du format sera faite plus tard.

Mode création : On utilisera cette méthode le plus souvent, elle permet de créer les noms et la structure de chaque champ de la table.

Assistant Table : Access propose différents modèles de tables prédéfinis (carnet d'adresses, factures, etc...) et crée la table, vous devrez la personnaliser ensuite pour qu'elle réponde à vos besoins.

Importer la table : Si les données de votre table sont déjà contenues dans une autre table, dans un fichier ou dans une autre base de données, vous pouvez utiliser cette méthode pour importer vos données dans une nouvelle table.

Attacher une table : Cette option sert à attacher une table avec une autre table contenue dans une autre base de données, nous ne l'utiliserons pas.

Dans ce livre, nous étudierons les deux modes que nous jugeons les plus utiles à savoir : modes création et importation.

5.4.3.2 Création d'une table par définition de sa structure



TAF : Ouvrir une base de données existante dans votre serveur de base de données et créer une nouvelle table campagne.mdb dont la structure est ci-dessous donnée.

Clé	Nom	Type	Définition
#	SHIP_CODE	texte(15)	Code Bateau
#	FLAG	texte(3)	Pavillon
	SHIP_NAME	texte(30)	Nom Bateau
	TJB	Numérique (entier)	Tonnage Jauge Brute
	LENGTH	Numérique (octet)	Longueur (en mètres)
	POWER	Numérique (entier)	Puissance en Cvx

Construire cette table en utilisant le mode Création.

Démarche :

1. Ouvrir la base de données CAMPAGNES.MDB
2. Cliquez sur l'onglet « Tables » dans la fenêtre Base de données.
3. Cliquez sur le bouton « Nouveau ».
4. Marquez la ligne « Mode création » dans la boîte de dialogue « Nouvelle table ». Puis validez par le bouton « Ok ». La fenêtre d'une table en mode création s'ouvre. Chaque ligne de ce tableau contient les informations de chaque champ :
 - Le **nom** du champ de la table (de 1 à 64 caractères)
 - Le **type** de données (voir plus bas) à choisir parmi 9 types proposés par Access
 - La **description** du champ (255 caractères maximum)

Nom champ	Type	Description
ship : Table		
Field Name	Data Type	
ship_code	Text	Code Bateau
flag	Text	Pavillon
ship_name	Text	Nom Bateau
tjb	Number	Tonnage Jauge Brute
length	Number	Longueur (en mètre)
power	Number	Puissance en Cvx

5. Saisir les informations nécessaires pour les 6 champs de la table SHIP
6. Choisissez le champ SHIP-CODE comme clé primaire (pour cela sélectionnez le champ SHIP-CODE, cliquez sur le bouton « Clé primaire » dans la barre d'outils « Création de table »).
7. Cliquez sur le bouton « Enregistrer » dans la barre d'outils « Création de table ».
8. Tapez le nom de la table : SHIP et validez par le bouton « Ok ».

5.4.3.3 Liste des types supportés par Access

Chaque champ (colonne) peut contenir des données de différents types : un nom va contenir des caractères alphabétiques, un code postal va contenir des chiffres, une date de livraison une date, un champ prix va contenir des valeurs monétaires.

Il faut choisir, pour chaque champ de la table, le type de données le plus approprié, en effet, rien n'interdit de choisir comme type de données pour un champ contenant une date ou un code postal un type «texte» ou pour champ contenant un prix un type «numérique», mais autant utiliser les types les plus appropriés !

Texte (Valeur par défaut) Pour un champ destiné à contenir des caractères alphanumériques (lettres et/ou chiffres). On l'utilise pour un nom, une adresse, ainsi que pour des nombres qui ne nécessitent pas de calculs, tels que des numéros de téléphone, etc... 255 caractères maximum.

Mémo Texte utilisé pour une description longue. Jusqu'à 65 535 caractères (64 Ko)

Numérique Données numériques utilisées dans des opérations mathématiques. Il peut être soit : un octet, un entier ou un réel. 1 octet (Nombre entre 0 et 255)

Date/heure Valeurs de date et heure pour les années comprises entre 100 et 9999. 8 caractères.

Monétaire Valeurs monétaires et données numériques utilisées dans des opérations mathématiques sur des données de 1 à 4 décimales. Précision à 15 chiffres à gauche de la virgule et à 4 chiffres à droite. 8 caractères.

NuméroAuto Numéro séquentiel unique (incrémenté de 1 en 1) ou numéro aléatoire attribué par Microsoft Access lorsqu'un nouvel enregistrement est ajouté à une table. Les champs NuméroAuto ne peuvent pas être modifiés. 4 octets.

Oui/Non Valeurs Oui et Non et champs qui contiennent exclusivement une valeur duale (Oui/Non, Vrai/Faux, ou Actif/Inactif). 1 bit.

Création d'une table par importation des données



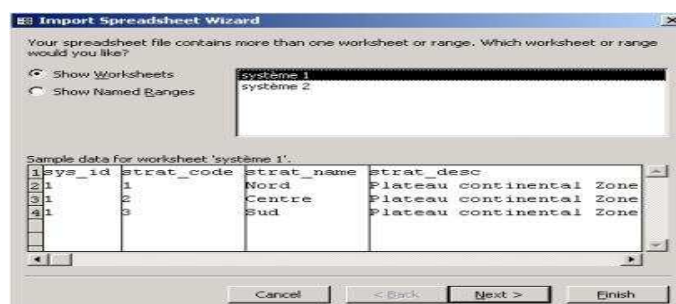
TAF : Dans la base de données de l'activité précédente, créer une nouvelle table en important les données qui se trouvent dans un fichier strate.xls

Clé	Nom	Type	Définition
#	SYS_ID	octet	Identifiant interne du système de stratification
#	STRAT_CODE	Text(15)	Identifiant de la strate
	STRAT_NAME	Texte(50)	Nom de la strate
	STRAT_DESC	Mémo	Description de la strate
	SURFACE	Entier Long	Surface de la strate en Km ²

Construire cette table par importation du fichier EXCEL STRATES.XLS

Démarche :

1. Sélectionnez le menu Fichier, Données externes, importer...
2. Dans la boîte de dialogue « Importer », sélectionnez dans la liste « Microsoft Excel » comme type de fichier.
3. Sélectionnez le fichier EXCEL STRATES.XLS qui se trouve dans le répertoire des données de la formation.
4. Cliquez sur le bouton « Importer ».
5. La première page de l'assistant vous propose de sélectionner la feuille du fichier Excel depuis laquelle vous voulez importer les données. Sélectionnez la première, appelée « système 1 » qui s'affiche alors dans la grille. Ensuite cliquez sur le bouton « Suivant ».



6. Indiquez dans la 2ème page si votre feuille contient ou pas un en-tête des colonnes, qui sera utilisé pour établir les noms des champs.
7. Dans la 3ème page, sélectionnez l'option « Dans une nouvelle table », puis cliquez sur le bouton « Suivant » pour passer à l'étape suivante.
8. Marquez une colonne dans la grille. l'assistant vous donne la possibilité de préciser le nom du champs, indexé ou non, s'il sera importé ou pas. Une fois terminé, cliquez sur le bouton « Suivant ».
9. Dans cette étape, sélectionnez l'option « Pas de clé primaire ». En effet, vous ne pouvez ici choisir qu'un seul attribut comme clé primaire, et la relation Stratum a une clé primaire composite. Nous la définirons plus tard. Cliquez sur le bouton « Suivant ».
10. Dans la dernière étape vous devez spécifier le nom de votre nouvelle table. Dans notre cas la table sera nommée STRATUM. Cliquez sur le bouton « Terminer » pour créer votre table.

sys_code	strat_code	strat_name	strat_desc	surface
1	1	Nord	Plateau continental Zone Nord	2000
1	2	Centre	Plateau continental Zone Centre	3000
1	3	Sud	Plateau continental Zone Sud	2500

11. Éditer la structure de la nouvelle table :
 - Faire les modifications nécessaires sur le type et la description des champs
 - Définir la clé primaire

Field Name	Data Type	Description
sys_id	Number	Surface de la strate en km2
strat_code	Text	Identifiant interne du système de stratification
strat_name	Text	Nom donné à la strate
strat_desc	Memo	Description de la Strate (critères qui ont servis à sa définition)
surface	Number	Surface de la strate en km2

Éditer les données d'une table

Les attributs d'un enregistrement peut être renseignés par saisie au clavier en ouvrant la table en mode *Feuille de données*. Chaque ligne correspond à un enregistrement. Chaque colonne correspond à un attribut.

En bas de la fenêtre de saisie, une barre indique le numéro de l'enregistrement sur lequel on est positionné. Des boutons de navigation permettent de se déplacer d'un enregistrement à l'autre ou de créer un nouvel enregistrement :

Pour supprimer un enregistrement, on sélectionne la ligne correspondance (clique à gauche de la ligne) et on utilise la touche SUP du clavier.

Selon les propriétés du champ, des contrôles peuvent être effectués lors de la saisie. En particulier :

- deux enregistrements ne peuvent pas avoir la même clé primaire
- un élément d'une clé primaire ne peut avoir une valeur nulle

D'autres propriétés permettent :

- la saisie automatique de l'attribut (champs auto numériques)
- la saisie par choix des valeurs dans une liste de valeurs possibles
- l'initialisation automatique de la valeur d'un champ lors de l'ajout d'un nouvel enregistrement

La saisie d'un champ est validée par déplacement du curseur sur une autre cellule.

Saisie au clavier d'un nouvel enregistrement



Saisir un enregistrement dans la table STRATSYSTEM avec les données ci-dessous :

sys_id	sys_name	sys_desc
2	stratification campagnes Maroc	Système de stratification basé sur le croisement de zones géographiques et de strates bathymétriques

Démarche :

1. Ouvrir la table STRATSYSTEM
2. La ligne de saisie d'un nouvel enregistrement est indiquée par une dans la colonne de gauche.
3. Saisir les valeurs de ce nouvel enregistrement à partir des données du tableau ci-dessus.



4. Saisir les valeurs de ce nouvel enregistrement à partir des données du tableau ci-dessus.

sys_id	sys_name	sys_desc
2	Système de stratification campagnes Maroc	Système de stratification basé sur des secteurs géographique croisés avec des strates bathymétriques

5. Corriger l'orthographe de Tunisie dans le nom du système de stratification ayant pour clé primaire 1

6. Afin de mieux comprendre la notion de clé primaire, modifier la valeur de la clé du deuxième enregistrement (mettre 2 au lieu de 1) et vérifier que cette valeur n'est pas acceptée. En effet, il existe déjà un enregistrement ayant pour valeur 1 comme clé primaire.

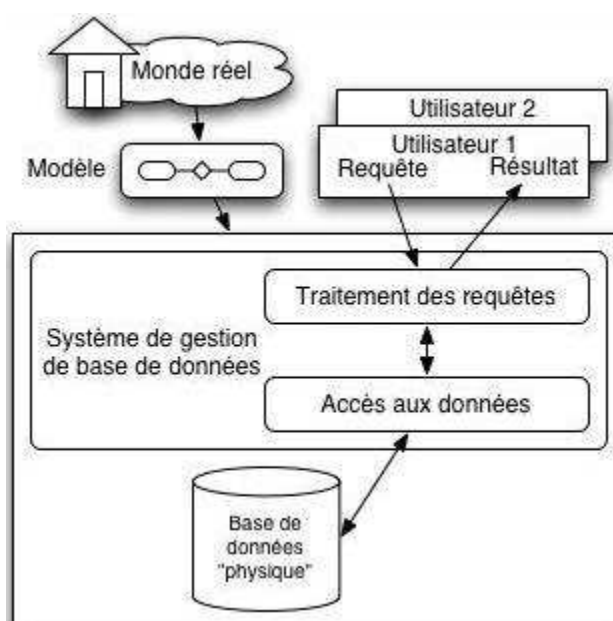


Saisir dans la table SHIP créée précédemment les 3 enregistrements suivants :

Ship_code	flag	ship_name	tjp	length	power
CAI	MAR	Charif Al Idrissi	397	41	1100
HAN	TUN	HANNOUN		24	900
HANNI	TUN	HANNIBAL		33	

6

CRÉER UNE BASE DE DONNÉES



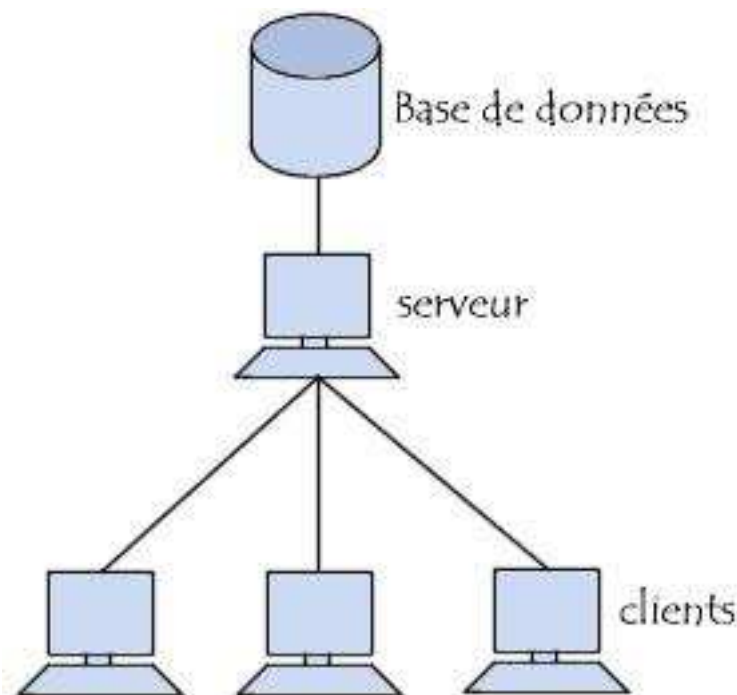
Sommaire

6.1	Les bases de données	50
6.2	Les tables	51
6.3	Les champs d'une table	53

6.1 Les bases de données

6.1.1 Qu'est-ce qu'une base de données ?

Une base de données est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.



6.1.2 Utilité d'une base de données ?

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses.

Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau.

L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.

6.1.3 La gestion des bases de données

Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion de la base de données se fait grâce à un système appelé SGBD (système de gestion de bases de données). Le SGBD est un ensemble de services permettant de gérer les bases de données, c'est-à-dire :

- permettre l'accès aux données de façon simple
- autoriser un accès aux informations à de multiples utilisateurs
- manipuler les données présentes dans la base de données (insertion, suppression, modification)

Le SGBD peut se décomposer en trois sous-systèmes :

- le système de gestion de fichiers : il permet le stockage des informations sur un support physique
- le SGBD interne : il gère l'ordonnancement des informations
- le SGBD externe : il représente l'interface avec l'utilisateur

6.1.4 Les principaux SGBD

Les principaux systèmes de gestion de bases de données sont les suivants :

- Microsoft Access
- Microsoft SQL server
- MySQL
- Oracle
- SQL Server

6.2 Les tables

Définition

Une table est une collection de données relatives à un domaine bien défini, par exemple les employés d'une société ou les livres d'une bibliothèque. Elle contient des enregistrements dont chacun est composé par les mêmes champs de données.

Voici, à titre d'exemple, quelques employés d'une société :



Voici la table nécessaire pour stocker les informations concernant ces employés dans une BD :

Table: tb1Employés Un champ de données

Nom	Prénom	Âge	Salaire	Service
Weber	Jos	34	68 000 FCFA	Comptabilité
Da Costa	Antonio	27	70 000 FCFA	Informatique
Feller	Emil	43	65 000 FCFA	Expédition
...	...	...	...	...

← Un enregistrement

Propriétés des tables

- Les champs de données définissent les informations, qu'on veut stocker dans la table (p.ex. des informations concernant les employés d'une société).
- Chaque enregistrement représente une occurrence de ce qu'on veut stocker (p.ex. un employé).
- Chaque table possède un nom unique (p.ex. : tblEmployés).
- Chaque enregistrement correspond à une ligne de la table.
- Chaque champ correspond à une colonne de la table.
- Chaque champ peut représenter des données de nature différente (Nom, Salaire, Date de naissance etc.).
- Chaque champ peut représenter des données de type différent (Texte, Nombres, Dates etc.).

6.3 Les champs d'une table

Une table reprenant les données concernant les voitures d'une société de taxis contient par exemple pour chaque enregistrement (= chaque taxi) les informations suivantes :

- Marque
- Modèle
- Cylindrée
- Poids

Il est évident que les informations sont de types différents. Tandis que la marque et le modèle sont représentés par des chaînes de caractères (p.ex. "Ford", "BMW", etc.), la cylindrée et le poids sont représentés par des valeurs numériques.

Voici, à titre d'exemple, une table qui représente les taxis dans une BD :

Marque	Modèle	Cylindrée	Poids
BMW	525i	2500	1360
Ford	Orion	1800	1080
BMW	320i	2000	1200
....	...	...	...

Afin de pouvoir représenter des données de types différentes, les SGBD offrent des types de données standards pour les champs de données. Voici les types de données connus par la plupart des SGBD :

Type de données	Description
Date/Heure	Date et heure
Valeur booléenne	Deux valeurs <i>Oui / Non</i> (Yes / No) sont possibles
Texte	Chaînes de caractères
Numérique	Nombres entiers ou décimaux
Mémo	Documents (textes longs)

Consultez le manuel d'utilisation de votre SGBD pour trouver des informations plus détaillées concernant les types de données supportés.

Remarque

Les nombres qui ne sont pas utilisés lors de calculs numériques (p.ex. No.Tél) sont généralement représentés à l'aide du type de données "Texte".



Réfléchissez pour chaque champ des 3 tables, que vous avez défini dans l'exercice du chapitre 6.1, sur le type de données approprié.



Lors de la création d'une table, nous devons indiquer au SGBD, pour chaque champ:

- Le nom du champ, qui doit être unique dans la table
- Le type de données du champ

7

LES REQUÊTES

SQL Importe les fichiers Historique SQL

Exécuter une ou des requêtes SQL sur le serveur "localhost": ?

```
CREATE TABLE Personne (  
  nom VARCHAR(40),  
  'prénom' VARCHAR(40),  
  adresse TINYTEXT,  
  'téléphone' DECIMAL(10,0),  
  PRIMARY KEY (nom, 'prénom')  
)
```

Conserver cette requête dans les signets:

☐ Signet visible pour les autres utilisateurs

☒ Remplacer un signet existant du même nom

☒ Ne pas écraser cette requête depuis une autre fenêtre

[Délimiteur] ☒ Réafficher la requête après exécution

Exécuter

Sommaire

7.1	Notion de requêtes	56
7.2	Introduction au langage SQL	57

7.1 Notion de requêtes

Définition

Nous avons vu que la plupart des SGBD offrent la possibilité d'effectuer des recherches directement dans les tables. Les possibilités de formuler des critères de recherche sont cependant souvent assez limitées. Heureusement, la plupart des SGBD nous offrent également la possibilité de poser pratiquement n'importe quelle "question" à nos tables, sous forme de requêtes.

Les requêtes servent donc à répondre aux questions basées sur le contenu d'une ou de plusieurs tables. Nous allons plus tard étudier des requêtes, qui se basent sur plusieurs tables, mais pour l'instant nous allons nous limiter aux questions simples basées sur une seule table.

Exemple :

Reprenons notre table avec les taxis :

Taxi	fldMarque	fldModèle	fldCylindrée	fldPoids
1	BMW	525i	2500	1360
2	Ford	Orion	1800	1080
3	BMW	320i	2000	1200
		...	...	...

Une requête simple serait par exemple :

Quelles sont les marques et modèles des voitures ayant une cylindrée supérieure à 2000 ? Le résultat serait un sous-ensemble de la table avec seulement les enregistrements qui vérifient le critère de sélection ($\rightarrow$ cylindrée $>$ 2000). Pour chacun de ces enregistrements, le SGBD affiche en plus seulement les champs explicitement demandés ($\rightarrow$ fldMarque et fldModèle).

Une requête simple produit donc comme résultat un sous-ensemble des enregistrements d'une table. En plus, une requête nous permet d'afficher seulement certains champs pour les enregistrements appartenant à ce sous-ensemble.

On appelle ces requêtes "Requêtes de Sélection", puisqu'il s'agit d'une sélection de certains enregistrements.

Il n'existe cependant pas seulement des requêtes de sélection, mais également :

- Des requêtes d'insertion → insérer des enregistrements dans la table.
p.ex. Insérer un nouveau taxi : 4, BMW, 325i, 2500, 1270.
- Des requêtes de modification → modifier des enregistrements dans la table.
p.ex. Modifier la cylindrée des Ford Orion de façon à ce qu'elle devienne 2000.
- Des requêtes de suppression → supprimer des enregistrements de la table.
p.ex. Supprimer toutes les BMW.

Les requêtes possèdent l'avantage de pouvoir manipuler facilement un grand nombre d'enregistrements sans que l'utilisateur ne doive s'occuper de sélectionner enregistrement par enregistrement. Il lui suffit de spécifier des critères de sélection pour la requête, ainsi que l'opération à effectuer (→ simple sélection et affichage, insertion, modification ou suppression).

Bien que les requêtes de sélection soient implémentées d'une manière plus ou moins cohérente à travers les SGBD actuels, il existe des différences subtiles en ce qui concerne les requêtes d'insertion, de modification ainsi que de suppression. En plus, l'insertion et la suppression se font souvent de manière plus facile directement dans la table.

Bon à Savoir

Il existe 4 types de requêtes :

1. Requêtes de sélection
2. Requêtes d'insertion
3. Requêtes de modification
4. Requêtes de suppression.

Pour chaque requête nous retrouvons le cycle suivant :



7.2 Introduction au langage SQL

7.2.1 Généralités

Nous avons vu au chapitre précédent qu'il faut d'abord formuler une requête et puis l'exécuter, afin d'avoir des résultats. Vous pouvez probablement bien vous imaginer que les SGBD actuels ne comprennent pas le langage naturel. Aucun SGBD n'offre une possibilité

d'écrire p.ex. Je veux voir tous les taxis dont la marque est Ford Pour formuler une requête, l'utilisateur doit donc utiliser un langage spécialisé pour ce domaine.

Le langage SQL (**Structured Query Language**) est un standard international, en ce qui concerne les langages de manipulation des BD. SQL est connu par tous les SGBDR. Il faut cependant mentionner que, désormais la présence de standards internationaux tels que SQL-86, SQL-89 ou SQL-92, chaque SGBD sur le marché utilise un peu son propre dialecte du langage SQL.

7.2.2 Syntaxe SQL de base

Nous distinguons les 4 types de requêtes suivantes.

1. Requetes de sélection.

```
select <Nom_des_champs>  
from <Nom_de_la_table>  
where <Critères_de_sélection>;
```

2. Requetes d'insertion.

```
insert into <Nom_de_la_table>[(<Liste_des_champs>)]  
values(<Valeurs_pour_les_champs>);
```

Attention :

Lorsque vous n'indiquez pas la liste des champs derrière **insert into**, vous devez spécifier une valeur pour chaque champ de la table derrière VALUES . Les parenthèses derrière VALUES sont obligatoires. La liste des champs, lorsqu'elle est indiquée, contient les noms des champs, séparés par une virgule, et doit également être entourée de parenthèses.

3. Requetes de modification.

```
update <Nom_de_la_table>  
set <Nom_d'un_champ>={valeur},<Nom_d'un_champ>={valeur}, . . .  
where <Critères de sélection>;
```

4. Requetes de suppression.

```
delete from <Nom_de_la_table>  
where <Critères_de_sélection>;
```

Exemple :

Soit une table des employés d'une entreprise avec la structure suivante :

tblEmployés

Nom du champ	Type de données	Description
idEmployé	Numérique	Numéro de l'employé (Clé primaire)
fldPrénom	Texte	Prénom de l'employé
fldNom	Texte	Nom de l'employé
fldNationalité	Texte	Nationalité de l'employé
fldAge	Numérique	Age de l'employé
fldSexe	Texte	Sexe de l'employé
fldService	Texte	Service auquel l'employé est affecté

Voici le code SQL nécessaire pour effectuer quelques requêtes élémentaires :

1. Afficher le prénom et le nom de tous les employés

```
select fldPrénom, fldNom
from tblEmployés;
```

2. Insérer une nouvelle employée :

20	Angela	Portante	ITA	27	F	Comptabilité
----	--------	----------	-----	----	---	--------------

```
insert into tblEmployés
values (20, 'Angela', 'Portante', 'ITA', 27, 'F', 'Comptabilité');
```

3. Afficher toutes les nationalités représentées dans la société

```
select fldNationalité
from tblEmployés;
```

Quel est l'inconvénient de cette requête ?

Soit l'adaptation suivante de la requête :

```
select distinct fldNationalité from tblEmployés;
```

Expliquez l'utilité de l'option

```
distinct
```

4. Afficher tous les champs pour tous les employés

```
select idEmployé, fldPrénom, fldNom, fldNationalité, fldAge, fldSexe, fldService
from tblEmployés;
```

ou

```
select * from tblEmployés;
```

Les valeurs sont séparées par des virgules.

En SQL, les données du type TEXTE sont entourées d'apostrophes.

Les dates sont entourées du caractère # et indiquées dans le format américain #Mois/-Jour/Année#

Exemple : 20/4/98, #04/20/98# ou #4/20/98# ou #04/20/1998# ou #4/20/1998#
L'opérateur * permet d'afficher tous les champs définis dans la table.

8

ÉTUDE DU SERVEUR DE BASES DE DONNÉES MYSQL

The screenshot shows the 'Actions' and 'Interface' sections of a MySQL management tool. The 'Actions' section is titled 'MySQL localhost' and contains a 'Créer une base de données' (Create a database) option with a text input field, a 'Interclassement' (Collation) dropdown menu set to 'utf8_general_ci', and a 'Créer' (Create) button. Below this is a section titled 'Interface' which includes a 'Langue - Language' dropdown set to 'Français - French', a 'Thème / Style' dropdown set to 'Original', a 'Couleur au choix' (Color of choice) section with a 'Réinitialiser' (Reset) button, and a 'Taille du texte' (Text size) dropdown set to '82%'.

Sommaire

8.1	Concepts du modèle relationnel	61
8.2	Fonctions de MySQL	61
8.3	Interface avec PHP	63
8.4	Administration avec l'outil web phpMyAdmin	65

8.1 Concepts du modèle relationnel

Avant d'attaquer le vif du sujet, un petit glossaire du jargon des bases de données :

Domaine : ensemble des valeurs d'un attribut.

Relation : sous ensemble du produit cartésien d'une liste de domaines. C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux domaines et dont les lignes contiennent des tuples. On associe un nom à chaque colonne.

Attribut : une colonne d'une relation, caractérisé par un nom.

Tuple : liste des valeurs d'une ligne d'une relation.

8.1.1 Les relations

Une relation est une table comportant des colonnes (appelées aussi attributs) dont le nom et le type caractérisent le contenu qui sera inséré dans la table.

Imaginons que l'on veuille stocker dans notre base de données notre carnet d'adresses. On va donc créer la relation Personne qui aura pour attributs : nom, prénom, adresse, téléphone. Autrement dit, c'est une table nommée Personne possédant les colonnes : nom, prénom, adresse, téléphone.

Les lignes que contiendra cette table seront appelées enregistrements ou tuples.

Personnes

<i>nom</i>	<i>prénom</i>	<i>adresse</i>	<i>téléphone</i>
Dupond	Marc	8 rue de l'octet	0123456789

8.2 Fonctions de MySQL

8.2.1 Les fonctions

Bien que ces fonctions appartiennent typiquement à MySQL , la création d'un chapitre à part se justifie par le fait que je vais me contenter ici d'énumérer les fonctions les plus courantes.

Reportez-vous au manuel MySQL pour la liste détaillée de toutes les fonctions disponibles dans votre version du serveur MySQL.

Ces fonctions sont à ajouter à vos requêtes dans un SELECT, WHERE, GROUP BY ou encore HAVING. D'abord sachez que vous avez à votre disposition :

- les parenthèses (),
- les opérateurs arithmétiques (+, -, *, /,
- les opérateurs binaires (<, <=, >, >=, |, &),
- les opérateurs logiques qui retournent 0 (faux) ou 1 (vrai) (AND, OR, NOT, BETWEEN, IN),
- les opérateurs relationnels (<, <=, =, >, >=, <>).

Les opérateurs et les fonctions peuvent être composés entre eux pour donner des expressions très complexes.

8.2.2 Quelques exemples

Liste du nom des produits dont le prix est inférieur ou égale à 100.5 Fcfa.

```
SELECT nom
FROM produits
WHERE prix <= 100.5
```

Liste des nom et prénom des élèves dont l'âge est compris entre 12 et 16 ans.

```
SELECT nom, prénom
FROM élèves
WHERE age BETWEEN 12 AND 16
```

Liste des modèles de voiture dont la couleur est dans la liste : rouge, blanc, noir.

```
SELECT modèle
FROM voitures
WHERE couleur IN ( 'rouge', 'blanc', 'noir' )
```

Liste des modèles de voiture dont la couleur n'est pas dans la liste : rose, violet.

```
SELECT modèle
FROM voitures
WHERE couleur NOT IN ( 'rose', 'violet' )
```

8.2.3 Fonctions de comparaison de chaînes

Le mot clé LIKE permet de comparer deux chaînes. Le caractère '_' est spécial et signifie : 1 seul caractère, n'importe lequel.

L'exemple suivant permet de rechercher tous les clients dont le prénom commence par 'Jean', cela peut être 'Jean-Pierre', etc... :

```
SELECT nom
FROM clients
WHERE prénom LIKE 'Jean%'
```

Pour utiliser les caractères spéciaux ci-dessus en leur enlevant leur fonction spéciale, il faut les faire précéder de l'antislash : \'.

Exemple, pour lister les produit dont le code commence par la chaîne 'XE' :

```
SELECT *  
FROM produit  
WHERE code LIKE $'\_XE%'
```

8.3 Interface avec PHP

8.3.1 Connexion

Pour se connecter à une base depuis un script php, il faut spécifier un nom de serveur, un nom d'utilisateur, un mot de passe et un nom de base.

Aucune connexion n'est possible sans authentification auprès du serveur de base de données.

Les actions possibles de l'utilisateur sur la base à laquelle il se connecte dépendent des droits qui lui auront été fournis par l'administrateur de la base de données.

mysql_connect(\$server,\$user,\$password) : permet de se connecter au serveur \$server en tant qu'utilisateur \$user avec le mot de passe \$password, retourne l'identifiant de connexion si succès, **FALSE** sinon. Si ces arguments manquent, les valeurs par défaut du fichier de configuration php.ini seront utilisées.

mysql_select_db(\$base[\$id]) : permet de choisir la base \$base, peut prendre un identifiant \$id de connexion; retourne TRUE en cas de succès, sinon FALSE. Les identifiants de connexion ne sont pas nécessaires si on ne se connecte qu'à un seul serveur à la fois, ils permettent seulement de lever toute ambiguïté en cas de connexions multiples (vers plusieurs serveurs dans le même script).

mysql_close([\$id]) : permet de fermer la connexion à un serveur de bases de données, l'argument optionnel \$id est l'identifiant de session retourné à l'ouverture de la connexion.

A noté que toutes les connexions aux serveurs de bases de données sont automatiquement fermées à la fin de l'exécution du script qui les aura ouvertes.

Dans le cas où le visiteur du site doit naviguer à travers différents script PHP qui se connectent tous au même serveur, il est préférable d'avoir recours aux « connexions persistantes ». Une connexion persistante est ouverte avec la fonction **mysql_pconnect()** qui est en tout point comparable à **mysql_connect()** à la seule différence que la connexion n'est pas fermée à la fin du script qui a ouvert la connexion. Ainsi, les scripts suivants peuvent continuer à lancer des requêtes à la base de données sans à avoir à rouvrir de connexion en direction du serveur.

Une connexion persistante ne peut pas être fermée avec la fonction **mysql_close()**. Au delà d'un certain temps d'inactivité, la ou les connexions persistantes ouvertes sont

automatiquement fermées.

8.3.2 Script de connexion

8.3.2.1 Exemple 1

```
if( $id = mysql_connect('localhost','foobar','0478') ) {  
    if(mysql_select_db('gigabase')) {  
        echo 'Succès de connexion.';  
        /* code du script... */  
    } else {  
        die('Echec de connexion à la base.');    }  
    mysql_close($id);  
} else {  
    die('Echec de connexion au serveur de base de données.');}
```

8.3.2.2 Exemple 2

```
@mysql_connect('localhost','foobar','0478') or die('Echec  
de connexion au serveur.');
```

```
@mysql_select_db('gigabase') or die('Echec de sélection de la base.');
```

Cet exemple est équivalent au précédent mais plus court à écrire. Le symbole @ (arobase) permet d'éviter le renvoi de valeur par la fonction qu'il précède.

Une connexion persistante évite d'avoir à rouvrir une connexion dans chaque script. Les connexions sont automatiquement fermées au bout d'un certain temps en cas d'absence de toute activité...

8.3.3 Interrogation

Pour envoyer une requête à une base de donnée, il existe la fonction : **mysql_query(\$str)** qui prend pour paramètre une chaîne de caractères qui contient la requête écrite en SQL et retourne un identificateur de résultat ou FALSE si échec.

8.3.3.1 Exemple :

```
$result = mysql_query('SELECT téléphone FROM Personnes  
WHERE nom=\''.$name.'');'
```

Cet exemple recherche le téléphone d'une personne portant pour nom la valeur de la chaîne **\$name**. L'identificateur de résultat **\$result** permettra à d'autres fonctions d'extraire ligne par ligne les données retournées par le serveur. Chaque appel à cette fonction retournera un tuple du résultat. C'est pourquoi cette instruction pourra être utilisée au sein d'une boucle **while** qui s'arrêtera lorsque **mysql_query()** renverra FALSE.

8.3.4 Fonctions sur le serveur

mysql_create_db(\$base [, \$id]) : création de la base **\$base**. **mysql_db_name(\$result, \$row [, \$field])** : Lit les noms des bases de données. **\$result** est l'identifiant de résultat issu de **mysql_list_dbs()**. **\$row** est l'index dans le résultat. Retourne FALSE si échec.

mysql_db_query(\$base, \$query [, \$id]) : exécution de la requête **\$query** sur la base **\$base**. Retourne un identifiant de résultat si succès ou FALSE si échec.

mysql_query(\$query [, \$id]) : exécution de la requête sur la base ouverte. Retourne un identifiant de résultat si succès ou FALSE si échec. **mysql_drop_db(\$base [, \$id])** : supprime la base de données **\$base**. Retourne TRUE si succès ou FALSE si échec.

mysql_select_db(\$base [, \$id]) : sélectionne la base de données **\$base** sur le serveur sur lequel on est connecté et dont **\$id** est l'identifiant de connexion. Retourne TRUE si succès ou FALSE si échec.

Ce script ci-dessous affiche la liste des bases de données du serveur.

```
$result = mysql_list_dbs();
$num = mysql_num_rows($result);
for ($i=0; $i<$num; $i++)
echo mysql_db_name($result, $i). '<br />';
```

8.4 Administration avec l'outil web phpMyAdmin

L'outil phpMyAdmin est développé en PHP et offre une interface intuitive pour l'administration des bases de données du serveur.

Cet outil permet de :

- créer de nouvelles bases
- créer/modifier/supprimer des tables
- afficher/ajouter/modifier/supprimer des tuples dans des tables
- effectuer des sauvegarde de la structure et/ou des données
- effectuer n'importe quelle requête
- gérer les privilèges des utilisateurs

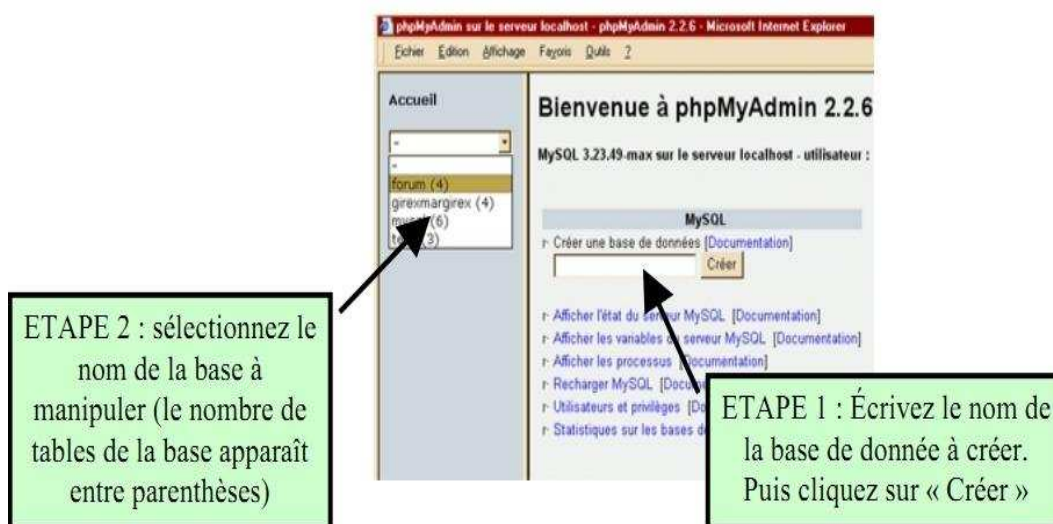
8.4 Administration avec l'outil web phpMyAdmin



8.4.1 Création/sélection d'une base de données

Avant de manipuler des données, il faut créer une ou des bases de données.

Avant de manipuler des données, il faut créer une ou des bases de données.



8.4.2 Gestion de la base de données

Choix d'une table à gérer en particulier

Actions sur les tables : afficher leur contenu intégral, faire une sélection sur critères, ajouter des données, gérer ses propriétés intrinsèques, supprimer, vider.

Écrire une requête MySQL à exécuter

Exécuter une requête MySQL contenue dans un fichier

Accès à un formulaire détaillé pour effectuer une requête

Option permettant de transmettre le schéma sous la forme d'un fichier

Affichage du schéma (structure et/ou données) des tables sélectionnées de la base

Accès à un formulaire détaillé d'ajout d'une table dans la base

Supprimer la base

8.4.3 Affichage d'une table

Base de données forum - table mforum sur le serveur localhost

Affichage des enregistrements 3 - 5 (12 total)

requête SQL : [Modifier]
SELECT * FROM 'mforum' LIMIT 3, 3

Rappel de la requête

Rappel de la base, de la table et du serveur

Colonnes = noms des attributs de la table

Liste des enregistrements de la table par pages de X lignes

Supprimer un enregistrement

Accès au formulaire de modification d'un enregistrement

Permet de naviguer dans les pages de résultats

Insertion d'un nouvel enregistrement

Afficher par page de X lignes

	id	titre	mesg	hits	thread_idx	author_idx	date
Modifier Effacer	4	Concours de logo AML 4	Vous êtes tous invités à participer au concours de...	2	0	1	2002-09-18 11:13:40
Modifier Effacer	5	Concours de logo AML 5	Vous êtes tous invités à participer au concours de...	2	0	1	2002-09-18 11:13:40
Modifier Effacer	6	Concours de logo AML 6	Vous êtes tous invités à participer au concours de...	0	0	1	2002-09-18 11:13:40

8.4.4 Insertion/modification d'un enregistrement

Nom du champ

Type

Choix d'une fonction à appliquer à la valeur saisie

Valeur à saisir

Champ	Type	Fonction	Valeur
id	bigint(20) unsigned		
titre	tinytext		
mesg	text	SOUNDEX LCASE UCASE NOW PASSWORD MD5 ENCRYPT RAND LAST_INSERT_ID COUNT AVG	hello CyberZoide
hits	mediumint(8) unsigned		0
thread_idx	bigint(20) unsigned		0
author_idx	bigint(20) unsigned		0
date	datetime		2003-01-11 18:10:02

Les champs et leurs types sont définis lors de la création de la table : tous les champs sont pas forcément obligatoires...

Les formulaires d'insertion et de modification sont similaires.

Insérer en tant que nouvel enregistrement -- et --

Retourner à la page précédente
Ou
Insérer un nouvel enregistrement

Exécuter

On accède à cet écran var le lien « *Insérer des données provenant d'un fichier texte dans la table* » de la page de gestion de la table.

8.4 Administration avec l'outil web phpMyAdmin

Emplacement du fichier texte		Parcourir...
Remplacer les données de la table avec le fichier	☐ Remplacer	Le contenu du fichier remplacera le contenu de la table pour les enregistrements ayant une valeur de clé primaire ou unique identique.
Champs terminés par		Le caractère qui sépare chacun des champs.
Champs entourés par	☐ ☐ OPTIONNEL	Souvent des guillemets. OPTIONNEL signifie que seuls les champs de type char et varchar sont entourés par ce caractère.
Caractère spécial		Optionnel. Indique le caractère qui permet d'enlever l'effet des caractères spéciaux.
Lignes terminées par	\n	Retour de chariot : \r Saut de ligne : \n
Nom des colonnes		Si vous désirez ne charger que certaines colonnes, indiquez leurs noms, séparés par des virgules.
[Documentation]		
Exécuter Réinitialiser les valeurs		

8.4.5 Création d'une clé

----- Créer une nouvelle clé -----									
Nom de la clé :	("PRIMARY" doit et ne peut être que le nom d'une clé primaire !)								
Type de clé :	INDEX [Documentation]								
<table border="1"><thead><tr><th>Champ</th><th>Taille</th></tr></thead><tbody><tr><td> -- Ignorer -- </td><td> </td></tr><tr><td> -- Ignorer -- </td><td> </td></tr><tr><td> -- Ignorer -- </td><td> </td></tr></tbody></table>	Champ	Taille	-- Ignorer --		-- Ignorer --		-- Ignorer --		
Champ	Taille								
-- Ignorer --									
-- Ignorer --									
-- Ignorer --									
Sauvegarder									
Ajouter à la clé 1 colonne(s) Exécuter									

On accède à cet écran via le lien « Créer une clé sur X colonne(s) » de la page de gestion de la table. Permet de créer une clé sur une ou plusieurs colonnes.

Il faut nommer la clé, en spécifier le type, et les attributs sur lesquels elle s'applique.

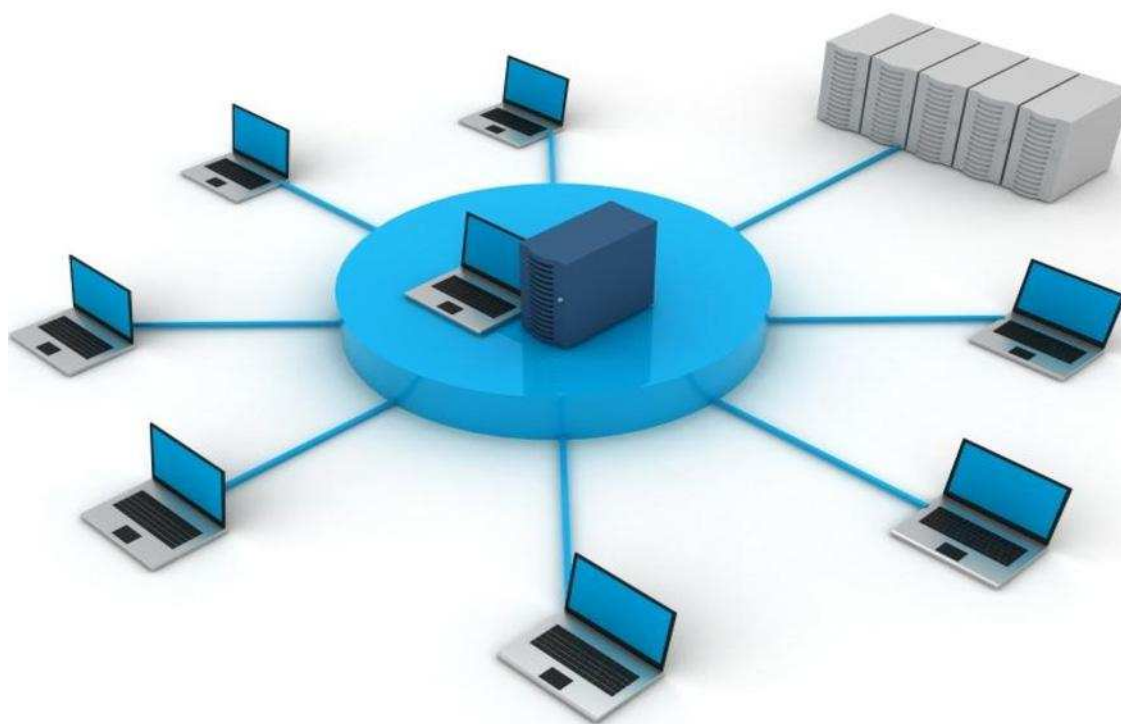
On peut rajouter une autre colonne à cette clé avant de valider l'ajout de la clé.

Troisième partie

LES RESEAUX LOCAUX

9

CRÉER UN RÉSEAU LOCAL



Sommaire

9.1	Terminologie	72
9.2	Classification des réseaux	72
9.3	Caractéristiques des réseaux locaux	73
9.4	Équipements	79

9.1 Terminologie

nœud Équipement (ordinateur, concentrateur, imprimante etc.) du réseau informatique possédant une adresse physique.

nœud TCP/IP : Équipement fonctionnant avec le protocole TCP/IP.

adresse physique : Adresse unique attribuée par le fabricant, gérée par un organisme international et inscrite de façon définitive dans l'équipement. Appelée aussi adresse Ethernet ou adresse **MAC** (Media Access Control).

adresse logique : Adresse unique attribuée par les répondants informatique locaux

paquet, trame : Information électrique transitant sur le réseau et représentant une suite de bits. Il s'agit de l'unité de base du réseau (frame).

protocole : Description des mécanismes permettant la gestion des paquets d'information et leur transition du réseau à l'application. Par extension, logiciel (software) fonctionnant sur une machine et permettant cette gestion interne.

suite de protocoles : Ensemble de protocoles communiquant entre deux.

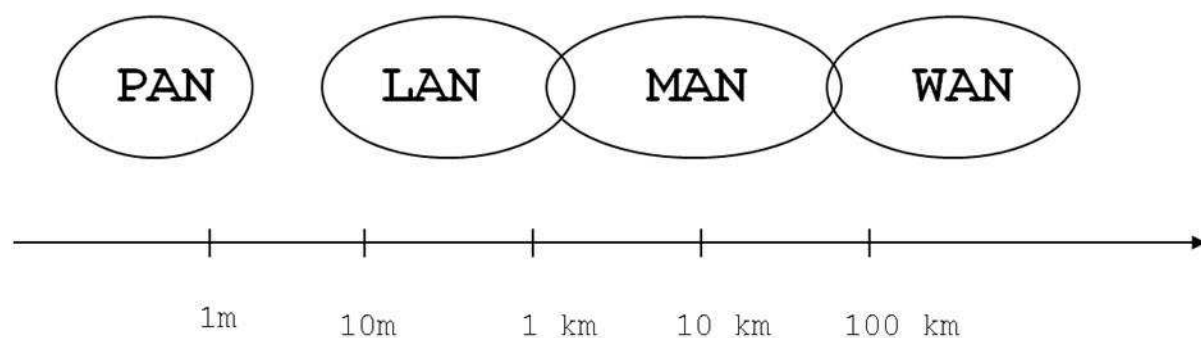
media : Support permettant le passage de l'information : différents types de câble, ondes hertziennes, laser etc.

bande-passante : Comme en téléphonie ou en hi-fi, la bande passante détermine la quantité d'informations capables de transiter par seconde sur un media donné.

backbone : Réseau physique (câble et matériel actif) servant de colonne vertébrale au reste du réseau. En général composée de fibres optiques est assurant parfois une plus grande bande-passante (155Mbits/s).

interconnexion : Ensemble matériel et logiciel permettant la connexion d'équipements utilisant des média et des protocoles différents

9.2 Classification des réseaux



Un réseau local ou LAN (Local Area Network) est un système de communication permettant d'interconnecter des ordinateurs et d'autres équipements informatiques dans un

domaine géographiquement limité. Il utilise des supports physiques de types paires filaires, câble coaxial, fibre optique, ondes électromagnétiques ou autres. Les dispositifs reliés entre eux sont de tous types : ordinateurs, imprimantes, mémoires de masse, tables traçantes, modems, fax, etc ... Par opposition, les communications à distance utilisent des réseaux de types WAN (Wide Area Network). De plus en plus les réseaux locaux sont reliés entre eux par des WAN .

9.3 Caractéristiques des réseaux locaux

Les caractéristiques principales permettant sur le plan physique de définir un réseau local sont :

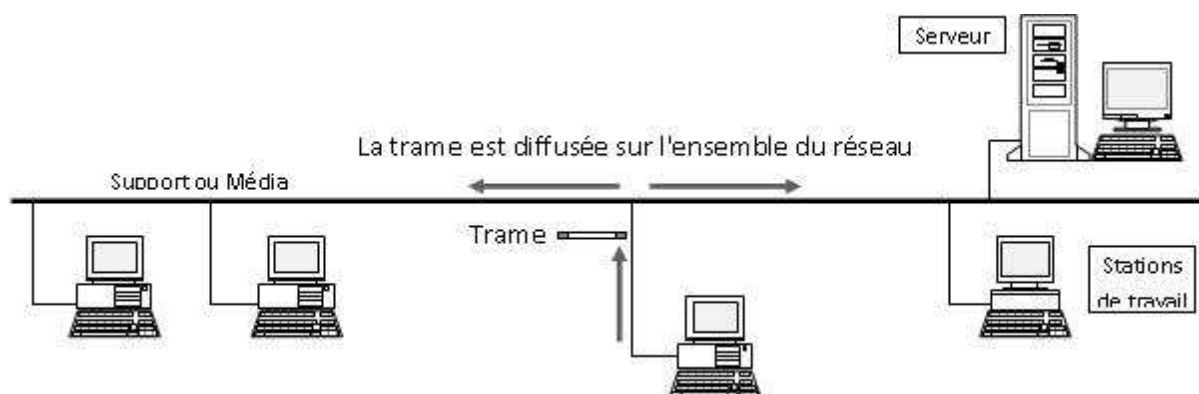
- La Topologie
- La Méthode d'accès au Support
- La Technique de transmission
- Le Support de transmission
- Le Débit Binaire

9.3.1 Topologie

On distingue deux types de topologies à savoir :

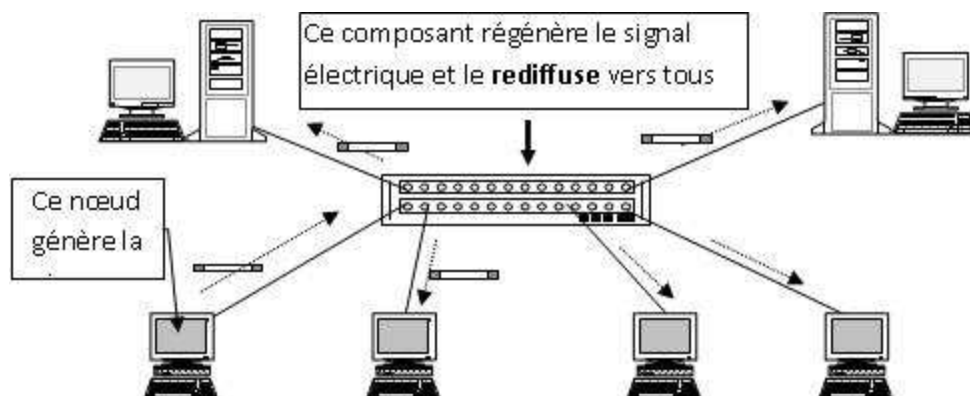
- **La topologie logique** qui définit la manière dont circulent les informations sur le réseau. Les informations véhiculées sur un réseau local le sont en utilisant des bits envoyés en mode série sur un support. Les bits sont expédiés regroupés en entités structurées appelées Trames.
- **La topologie physique** qui définit la manière dont le câblage réseau interconnecte les noeuds entre eux.

9.3.1.1 Topologie en Bus série



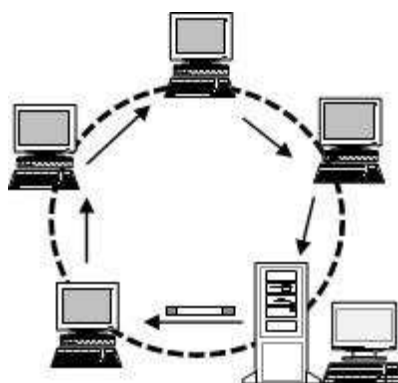
Tous les noeuds sont connectés électriquement au même support. Lorsqu'une trame est envoyée par une station, le signal électrique correspondant est diffusé sur l'ensemble du réseau. Cette diffusion n'est pas instantanée. Il existe un temps de propagation du signal électrique qui dépend des caractéristiques du support. Le temps de propagation varie de $0,6$ et $0,8 \times C$. C représente la vitesse de la lumière, soit 300.000 Km/s .

9.3.1.2 Topologie Bus étoilé



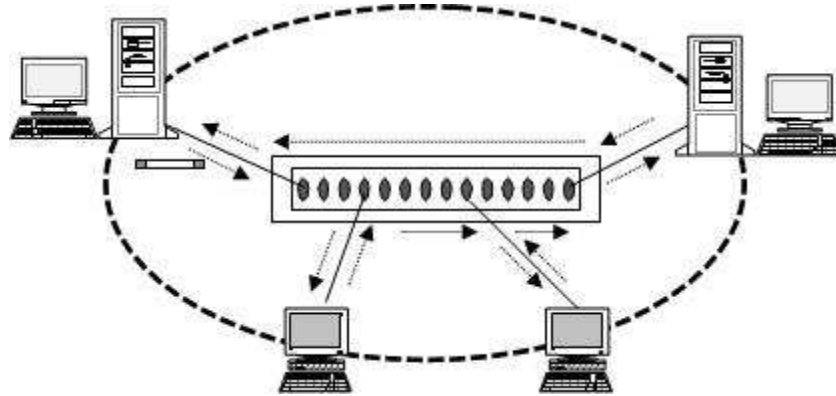
Dans cette topologie, on utilise un élément central qui régénère le signal électrique envoyé par un noeud et le rediffuse vers tous les noeuds du réseau. La topologie logique reste un bus série, alors que la topologie physique (le câblage) est une étoile.

9.3.1.3 Topologie en anneau



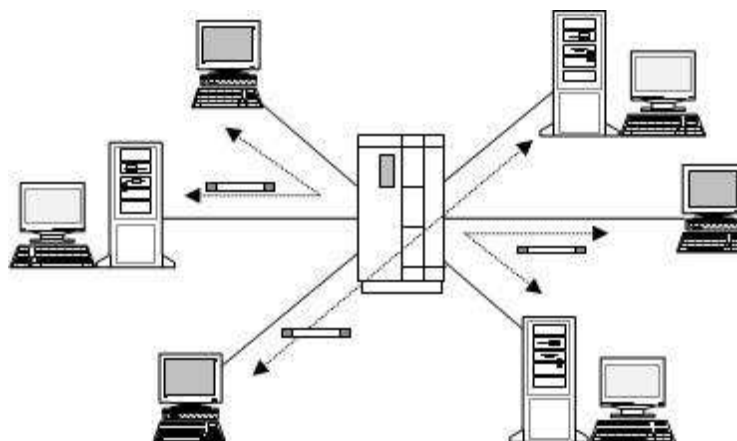
Dans ce type de topologie, les trames sont envoyées d'un noeud du réseau à l'autre et reviennent au point de départ. Un anneau logique est ainsi formé. En réalité, on notera qu'il s'agit d'une succession de liaisons point à point se rebouclant. Le signal électrique n'est pas diffusé sur l'ensemble du réseau, mais successivement d'un noeud à l'autre.

9.3.1.4 Topologie en anneau étoilé



Dans la topologie logique de type anneau, il suffit qu'un élément soit déconnecté ou en panne pour que l'anneau soit inutilisable. En pratique, on utilise une topologie physique de type anneau étoilé (figure 8) qui permet d'annuler cet inconvénient. Tous les noeuds sont connectés à un élément central. Les connecteurs de cet élément sont reliés entre eux en série. La dernière prise à droite est connectée en interne à la prise de gauche. Si un noeud émet une trame, elle circule de station en station en passant à chaque fois par l'élément central. Si une connexion à un noeud ou si un noeud est défectueux, le connecteur correspondant est court-circuité et l'anneau est malgré tout refermé.

9.3.1.5 Topologie en étoile



L'élément central connecte les noeuds 2 à 2. Il existe plusieurs liaisons point à point simultanées.

9.3.2 Méthodes d'accès au support

Chaque type de réseau local possède une méthode d'accès au support. Elle concrétise la manière dont chaque noeud peut envoyer des trames sur le réseau sans créer de conflits avec des trames émises par d'autres noeuds. La méthode d'accès est souvent conditionnée par la topologie utilisée. Ainsi, sur un bus série, 2 stations ne peuvent émettre en même temps sans provoquer une interférence entre les 2 signaux électriques émis. Cette interférence est appelée "Collision".

9.3.3 Techniques de Transmission

Il existe 2 méthodes de transmission possibles sur les réseaux locaux :

- La méthode Large Bande (Signal Analogique)
- La méthode Bande de Base (Signal Numérique)

Pour des raisons pratiques, seule cette dernière est vraiment utilisée. Les données sont envoyées en mode série et sous forme numérique sur le support. Pour des raisons tenant à la synchronisation du récepteur et à la largeur de bande du signal à transmettre, les données sont toujours envoyées de façon codée sur le support.

9.3.4 Supports de transmission

Le signal représentant les données doit disposer d'un support pour être véhiculé. Le signal électrique utilise des supports à base de cuivre (paires torsadées ou câbles coaxiaux). Le signal lumineux utilise les différents types de fibres optiques ou l'air (infrarouge, rayon laser). Dans tous les cas, le signal subit une atténuation (exprimée en dB/Km) et une déformation qui sont des fonctions de la qualité du support (bande passante) et de sa longueur. Pour chaque type de support, on se verra imposer des limites quant à la longueur maximale utilisable.

9.3.4.1 Paires torsadées

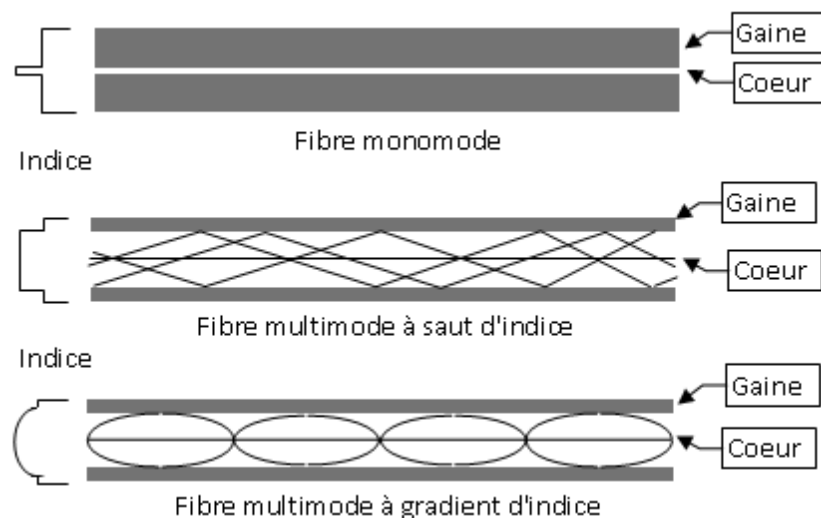
Les câbles paires torsadées utilisés dans les réseaux locaux possèdent en principe 4 paires (8 fils). Selon la qualité des câbles, on les classe en 3 catégories :

- catégorie 3 : câbles d'impédance 100 ou 120 dont les caractéristiques sont spécifiées jusqu'à 16 MHz.
- catégorie 4 : câbles d'impédance 100 ou 120 dont les caractéristiques sont spécifiées jusqu'à 20 MHz.
- catégorie 5 : câbles d'impédance 100 ou 120 dont les caractéristiques sont spécifiées jusqu'à 100 MHz.

L'impédance 100 correspond à une norme internationale, alors que l'impédance 120 est d'origine européenne. Dans les nouvelles installations, seule la catégorie 5 doit être utilisée pour préserver l'avenir. Les câbles peuvent avoir des paires blindées (STP = Shielded Twisted Pairs) ou non blindées (UTP = Unshielded Twisted Pairs). Certains câbles possèdent un écran général pour toutes les paires (câble écrané). Les paires torsadées sont de plus en plus utilisées pour le pré-câblage d'immeubles. Elles peuvent autoriser l'utilisation des mêmes types de câbles pour les différents types de réseaux LAN ou WAN (liaison RS232, RNIS, Téléphonique, Ethernet ou Token-Ring)

9.3.4.2 Fibres optiques

La fibre optique est un conducteur des signaux lumineux. Le matériau de base est la silice ou le plastique. Ce dernier matériau moins cher, possède cependant des performances inférieures. Selon le mode de propagation de la lumière dans le coeur de la fibre, on distingue 3 types de fibres optiques :



Les fibres optiques, bien qu'ayant des bandes passantes très importantes ne sont utilisées que pour des liaisons point à point entre des segments de réseaux ou en anneau pour connecter des réseaux entre eux. Les difficultés de connectique empêchent, pour l'instant, une utilisation systématique des fibres optiques

9.3.4.3 Air

L'air est utilisé comme support des ondes électromagnétiques. Quelques tentatives de réseau local employant les ondes radio sont observées. Mais les problèmes d'encombrement des fréquences, d'interférences et de limitation de débits font que ces initiatives sont limitées.

Pour des liaisons point à point entre bâtiments ou sites distants, le faisceau laser peut être

utilisé. Ce moyen est soumis à réglementation et reste très coûteux. Mais, il peut faire concurrence dans l'avenir à des liaisons WAN entre LAN.

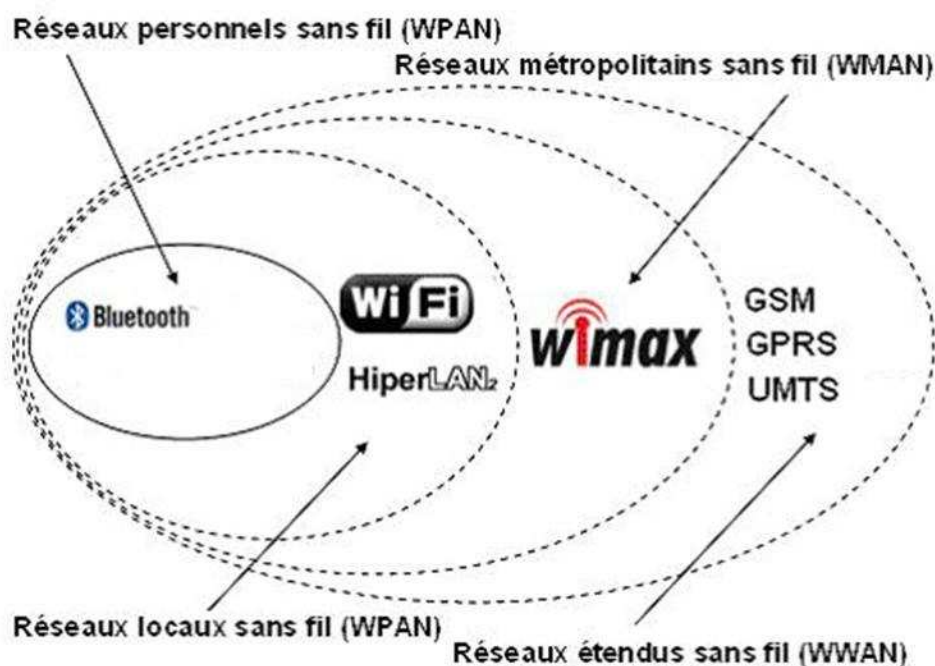
9.3.5 Débits

Les 2 versions de Token-Ring fonctionnent respectivement à 4 et 16 Mbps. Ethernet fonctionne actuellement à 10 Mbps et tend vers 100 Mbps.

9.3.6 Les autres LAN

Les réseaux locaux sans fil WLAN s'utilisent comme les réseaux filaires et couvrent quelques centaines de mètres. Les technologies sans fil évoluant très rapidement, on trouve toute une série de normes physiques, repérées par la lettre suivant le terme générique 802.11. Elles se distinguent par la bande de fréquences utilisée, les débits binaires et la portée dans un environnement dégagé.

Le Wi-Fi est l'une des premières solutions du standard 802.11. Il utilise la bande de fréquences 2,4 GHz sur une portée maximale de 300 mètres. Les débits disponibles pour les réseaux sans fils varient de 11 Mégabit/s pour le 802.11b à 54 Mégabit/s pour le 802.11g. Deux autres solutions plus récentes coexistent dans la bande des 5 GHz (Wi-Fi5 ou 802.11a et Hiperlan2 ou 802.11h). Le protocole d'accès utilisé dans les réseaux locaux sans fil est CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). Ce protocole est destiné à limiter les risques de collisions entre émissions provenant de machines se trouvant hors de portée l'une de l'autre.



9.4 Équipements

9.4.1 Répéteurs et Concentrateurs (Hubs)

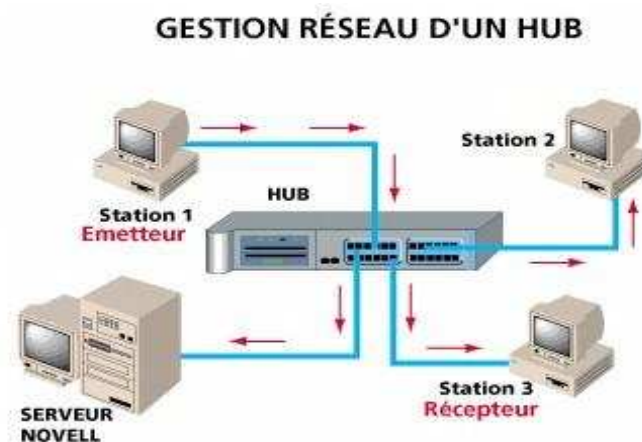
Le concentrateur est un appareil qui régénère les signaux. En effet, le signal émis par la carte Ethernet s'affaiblit en parcourant le câble et, au-delà de cent mètres, il peut devenir trop faible. Cette distance correspond en fait au maximum autorisé par la norme entre un PC et le concentrateur.

Un signal émis par un PC est régénéré sur tous les autres ports du concentrateur (il joue le rôle de répéteur).

Les machines sont reliées individuellement aux entrées ou « port » d'un concentrateur appelé « HUB ». Le Hub simule intérieurement le « bus ».

Les trames envoyées à destination d'une machine sont en fait reçues par toutes les machines même si elles sont seulement traitées par la machine à qui elles sont destinées (principe du BUS). De plus la bande passante théorique de 10 Mbits/s ou de 100 Mb/s n'est disponible pour chaque machine mais partagée entre toutes les autres, puisqu'une seule machine à la fois peut normalement émettre et recevoir sur le réseau.

Si le Hub du réseau ne possède plus de port disponible, il est possible alors de connecter un nouveau Hub avec l'ancien (chaînage de Hub). En norme Ethernet, un maximum de 4 Hubs peut être installé de cette façon et en Fast Ethernet, c'est un maximum de 2. Pour contourner cette contrainte et augmenter la taille du réseau, il faut alors opter pour des Hubs empilables : vous pouvez alors empiler plusieurs Hubs pour augmenter le nombre total de ports, chaque pile de Hubs n'étant vue que comme un seul Hub.



9.4.2 Ponts

Un pont analyse les paquets qui lui parviennent. Si un paquet arrivé sur le port A est destiné à un ordinateur connecté à son autre extrémité, alors, il le fait transiter. L'analyse se fait sur l'adresse MAC du destinataire.

9.4.3 Commutateurs (Switch)

Un commutateur (switch) est un équipement qui offre une bande passante dédiée pour chaque port (10,100 ou 1 000 Mbit/s par port) alors que le concentrateur partage la bande passante entre tous ses ports. Cela revient à créer un segment Ethernet par port.

On distingue les switches cut-through (on the fly, à la volée) et les switches store and forward (les plus courants aujourd'hui). Les premiers se contentent de régénérer la trame (même les trames erronées et les collisions), tandis que les seconds la stockent en mémoire avant de la régénérer. La méthode adaptative cut-through combine les deux principes : dès qu'une trame est en erreur, le commutateur bascule en mode store and forward pendant un certain temps. La méthode fragment free, la plus performante, lit les 64 premiers octets avant de décider du mode de transmission. Avec les concentrateurs-commutateurs, que l'on appelle le plus souvent Switched Hubs ou même tout simplement Switchs, les trames envoyées à une machine particulière sont directement aiguillées vers la machine destinataire, en supprimant toute collision, la topologie logique n'est donc plus le bus mais l'étoile.

De plus, chaque machine connectée à un port du Switch dispose d'une bande passante dédiée. Les machines connectées à un Switch peuvent travailler en full duplex c'est-à-dire qu'elles peuvent émettre et recevoir en même temps.

Le Switch permet des communications simultanées autant de fois que le nombre de ports du Switch divisé par 2. Chaque port est full duplex, ce qui permet d'émettre et de recevoir en même temps.



9.4.4 Routeurs

Les routeurs sont des commutateurs évolués. Ils analysent les paquets à des couches supérieures (Couche réseau : IP, IPX...). Ils permettent de sélectionner le meilleur itinéraire lorsqu'il y en a plusieurs. Ils permettent de passer d'un segment de réseau à un autre dans le cas de réseaux segmentés.

9.4.5 Passerelles

Les passerelles permettent à des architectures réseaux différentes de communiquer entre elles. Par exemple, elles permettent de transférer un paquet d'un réseau Ethernet vers un réseau Token Ring.

9.4.6 Accès Distant

L'accès distant permet de se connecter à un réseau à l'aide de différents périphériques, le plus courant étant le modem. Le Réseau privé virtuel (VPN) est un accès à distance utilisant une technique de cryptage pour garantir la confidentialité des données.

9.4.7 Technologie ADSL

La technologie ADSL (Asymmetric Digital Subscriber Line) permet de transmettre des volumes de données très importants sur des lignes téléphoniques classiques. Comme l'indique le nom, les débits de données sont asymétriques, donc plus importants en réception qu'en envoi.

10

PROTOCOLE RÉSEAUX

Sommaire

10.1	Notion de Protocole	83
10.2	Créer son propre réseau sous Windows XP	85
10.3	L'administration du réseau	86
10.4	Mise en place d'un réseau local	87

10.1 Notion de Protocole

Un protocole est un langage commun utilisé par l'ensemble des acteurs de la communication pour échanger des données. Toutefois son rôle ne s'arrête pas là. Un protocole permet aussi :

- L'initiation de la communication
- L'échange de données
- Le contrôle d'erreur
- Une fin de communication "courtoise"

10.1.1 Qui créent les protocoles ?

Ce sont eux qui créent les normes qui permettent la compatibilité des matériels informatiques. Citons en quelques uns :

AFNOR : Association Française pour la NORmalisation

ECMA : European Computer Manufacturer Association (une trentaine de fabricants de matériel informatique)

ANSI : American National Standard Institute (Equivalent américain de l'AFNOR)

IEEE : Institute of Electrical and Electronics Engineers

EIA Electronic Industries Association (Participe à l'ANSI, il est à l'origine des RS 232,...)

ISO : International Standard Organisation

UIT : Union Internationale des communications

10.1.2 Les principaux protocoles réseaux

Un protocole réseau est un langage que vont utiliser toutes les machines d'un réseau pour communiquer entre elles. Il se présente sous la forme d'un logiciel dans le système d'exploitation. Ils sont parfois composés d'une multitude de protocoles afin de pouvoir acheminer l'information (ex : TCP/IP avec IP,TCP,UDP,ICMP, ...).

10.1.2.1 Modèle de référence OSI

Le modèle OSI est une norme définie par l'ISO (International Organisation for Standardization) qui permet l'interconnexion réseau des systèmes hétérogènes. Il est composé des 7 couches suivantes :

1. **Application** Assure l'interface avec les applications.
2. **Présentation** Définit le formatage des données (représentation, compression, cryptage, ...)

3. **Session** Définit les canaux de communication sur les machines du réseau
4. **Transport** Chargée du transport des données et de la gestion des erreurs.
5. **Réseau** Permet de gérer les adresses et le routage des données.
6. **Liaison Données** Définit l'interface avec la carte réseau.
7. **Physique** Codage des données en signaux numériques.

10.1.2.2 Types de protocoles

Un protocole peut s'intégrer à plusieurs niveaux dans le modèle OSI :

- Protocoles d'application : Echange de données entre application (ex : FTP, IMAP, ...)
- Protocoles de transport : Assure la fiabilité des données transportées (ex : TCP, ...)
- Protocoles réseau : Détermine le chemin d'accès à la destination (ex : IP, ...)

10.1.3 Le protocole TCP/IP

TCP/IP (Transmission Control Protocol / Internet Protocol) est un ensemble de protocoles, développés au début des années 70 par le département américain de la défense afin de permettre l'interconnexion en réseau local de machines hétérogènes.

TCP/IP n'a pas que des avantages. Sa configuration n'est pas automatique : le technicien est obligé de définir, manuellement et individuellement sur chaque machine, une adresse IP (donc logique/réseau et non physique/MAC) qui devra impérativement être unique sur tout le réseau (LAN ou WAN), ainsi que divers paramètres techniques complémentaires, ce qui est laborieux dès que le réseau a quelque importance. On notera que la capacité de codage des adresses IP est actuellement limitée à 32 bits (forme : xxx.xxx.xxx.xxx), ce qui peut se révéler une limite insupportable pour les très grands réseaux comme Internet. Enfin, avec TCP/IP, le fait de passer d'Ethernet 10 Mb/s à 100 Mb/s ne multiplie pas par dix les performances (à cause de TCP).

Une adresse IP se présente sous la forme de quatre champs numériques Champ1.Champ2.Champ3.Champ4 (ex : 195.217.13.121), chaque champ représentant un octet pouvant donc prendre une valeur allant de 0 à 255.

Une adresse IP se décompose aussi en deux parties, l'identificateur réseau et l'identificateur d'hôte. Tous les ordinateurs ayant le même identificateur de réseau peuvent se voir et discuter directement.

10.1.3.1 Protocole IrDA

Le Protocole IrDA permet une communication sans fil entre 2 périphériques infrarouges. Il est intégré en standard à Windows 2000.

10.1.3.2 Protocole IPSec

Garantit la sécurité des transmissions de données sur le réseau en ajoutant une couche de cryptage au cours des communications TCP/IP. Requiert des ressources processeur supplémentaires (excepté dans le cas d'une implémentation matérielle dans la carte réseau) au niveau du client et du serveur.

10.2 Créer son propre réseau sous Windows XP

10.2.1 Matériel nécessaire pour relier 2 PC

Vous devez installer dans chacun des deux postes une carte réseau (également appelée carte Ethernet) puis les relier entre eux par un câble RJ45 dit "croisé". Des kits comprenant ces différents éléments ensemble sont souvent proposés en grande surface ou sur les sites marchands spécialisés dans la vente de matériel informatique.

10.2.2 Matériel nécessaire pour relier plus de 2 PC

Vous devez installer dans chacun des deux postes une carte réseau (également appelée carte Ethernet) puis les relier à un "hub" (boîtier central regroupant les différentes connexions) par un câble RJ45 dit "droit". Des kits comprenant cet ensemble sont souvent proposés en grande surface ou sur les sites marchands spécialisés dans la vente de matériel informatique.

Remarque : il existe d'autres moyens de créer un réseau domestique via les ports USB de vos ordinateurs par exemple. Néanmoins, l'usage de véritables cartes Ethernet vous garantit un réseau performant et rapide. L'USB quant à lui sera plus lent.

10.2.3 Montage des cartes réseau

Le montage de la carte réseau est très simple. Ce montage est bien entendu à réaliser sur chacun des PC que vous souhaitez relier en réseau.

Étape 1 : Éteignez votre PC et débranchez son câble d'alimentation. Débarrassez vous de l'électricité statique accumulée en vous en touchant un objet métallique. Un radiateur en fonte ou le châssis du boîtier de votre ordinateur...

Étape 2 : Ouvrir le capot de l'ordinateur et repérer un connecteur PCI de libre. Ces derniers sont de couleur blanche. Retirez le cache métallique du boîtier face à ce connecteur si besoin est, puis présentez la carte au connecteur.

Étape 3 : Insérez la carte dans le connecteur en faisant correspondre l'équerre de la carte avec le châssis du boîtier, sans forcer jusqu'à ce que l'équerre de la carte épouse le châssis du boîtier.

Étape 4 : Vissez pour maintenir la carte.

Étape 5 : Utilisez les câbles RJ45 fournis pour relier les cartes réseaux au hub si votre réseau comporte plus de deux PC. Si votre réseau ne comporte pas de hub, relier simplement les cartes des deux PC de votre réseau. Chaque carte réseau comporte un branchement de type RJ45 utilisé pour relier les PC entre eux ou les PC au hub dans le cadre d'un réseau de plus de 2 ordinateurs.

Étape 6 Redémarrez le PC afin que la carte soit reconnue par Windows. Sous XP la prise en compte d'une carte réseaux est en général automatique. Installez le cas échéant les pilotes fournis avec la carte si Windows vous le demande.

10.2.4 Attribution des adresses IP sous Windows XP

Maintenant que vos cartes réseau sont installées, il est nécessaire de configurer le réseau. Il est par exemple nécessaire de donner une adresse à chaque ordinateur du réseau... C'est ce que nous allons voir maintenant. Bien sûr cette configuration est à faire sur chaque poste du réseau.

Étape 1 : Cliquez avec le bouton droit de votre souris sur l'icône "Voisinage réseau" ou "Favoris réseau" (selon votre version de Windows) présente sur le bureau Windows et sélectionnez "Propriétés". Cette icône est présente si votre carte réseau a bien été installée, cela se fait en général tout seul sur XP.

Étape 2 : Faites un clic avec le bouton droit de votre souris sur l'icône "Connexion au réseau local" et sélectionnez la ligne "Propriétés".

Étape 3 : La fenêtre de propriétés réseau vous permet de configurer chacun des postes connecté au réseau. La première chose à faire est d'indiquer une adresse pour chaque poste. En informatique, on appelle cette adresse réseau adresse IP. Elle permet à chaque poste d'être reconnu par une adresse unique au sein du réseau. Sélectionnez la ligne "Protocole Internet TCP/IP" puis cliquez sur le bouton "Propriétés".

Étape 4 : Cochez la case "Utiliser l'adresse suivante" si cette dernière ne l'est pas déjà. Indiquez pour chacun des postes du réseau le masque de sous réseau 255.255.255.0 puis l'adresse IP à utiliser : 192.168.0.1 pour le premier poste, 192.168.0.2 pour le deuxième, 192.168.0.x pour le poste x.

Étape 5 : Fermez la fenêtre en cliquant sur le bouton "OK" pour prise en compte de ces paramètres.

10.3 L'administration du réseau

10.3.1 Notion d'administrateur

Administrateur (informatique), personne n'en charge d'un logiciel complexe, de son installation, de sa configuration, de son évolution et du suivi de ses performances.

La fonction d'administrateur se rencontre principalement pour les systèmes d'exploitation (administrateur système), pour les réseaux informatiques (administrateur réseau), pour les systèmes de gestion de bases de données (administrateur de base de données) et pour les sites Web (webmestre ou webmaster).

10.3.2 fonctions d'un administrateur reseau

L'administrateur d'un réseau a un certain nombre de tâches à réaliser de façon régulière :

- La gestion des Comptes d'utilisateurs et des groupes d'utilisateurs,
- La maintenance des ressources matérielles (imprimantes, télécopie, ...),
- Le contrôle de la sécurité du réseau,
- La vérification du bon fonctionnement du réseau,
- La maintenance des serveurs.
- La gestion des sauvegardes et des restaurations de données,
- La maintenance des applications serveurs (messagerie, bases de données, ...),
- Le contrôle de l'intégrité des disques,
- La maintenance des serveurs.

10.4 Mise en place d'un réseau local

10.4.1 Création de câble à paires torsadées

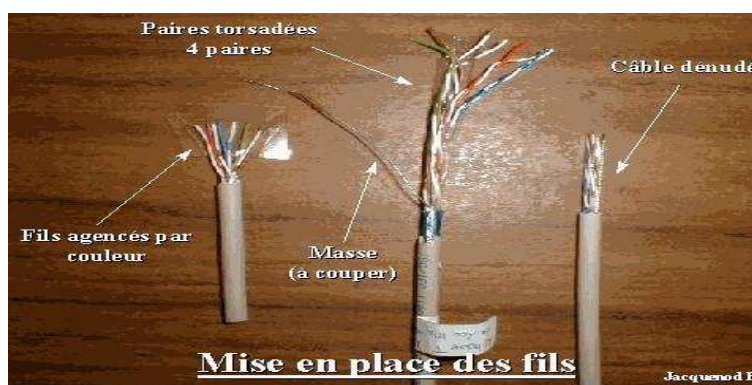
10.4.1.1 Éléments de base



Phase 1 : La gaine PVC est à retirer. Vous pouvez utiliser la pince coupante en faisant très attention à ne pas endommager les fils. Certaines pinces à sertir proposent une partie permettant de dénuder le câble. Enlevez une partie plus grande que celle à entrer dans la prise RJ45, vous aurez comme cela une plus grande latitude pour votre construction.

Phase 2 : La gaine PVC est retirée, vous découvrez alors la partie blindage (petite feuille d'aluminium). Avec la pince, un cutter ou une paire de ciseaux enlevée là.

Phase 3 : Vous découvrez une partie protection en plastique transparent. Enlevez la pour n'avoir que les paires torsadées. Vous avez aussi une masse, sous forme d'un petit fils en acier, ce dernier ne nous sert pas, coupez le. Il est utilisé généralement pour le raccordement à un bornier.



Les fils vont par paires, 4 paires de fils donc 8 fils. Les fils sont associés par couleur :

- (Orange)-(blanche orange),
- (Verte)-(blanche - verte),
- (Bleue)-(blanche - bleue),
- (Marron)-(blanche - marron).

Pour agencer les fils, il faut tout d'abord les détorsader.

Afin de réaliser une connectique fonctionnant en 100Mbps/s il faut ordonner les fils d'une certaine façon avant de les enclencher dans la prise RJ45 dans le même ordre.



Partager les données au sein d'un réseau

À ce stade, votre réseau est configuré. Il reste à définir pour chacun des postes connectés au réseau les répertoires contenant les données à partager.

Étape 1 : Cliquer avec le bouton droit sur le répertoire contenant les données à partager et sélectionner "Partager...". Dans notre exemple le répertoire s'appelle "Micro".



Étape 2 : Dans l'onglet "Partage" et la zone "Partage réseau et sécurité", cochez la case "Partager ce dossier sur le réseau" et indiquez le nom de votre partage (ex : Toto. Ainsi Toto saura que ce dossier contient les données qui lui sont destinées). Par défaut le nom de partage est le nom du répertoire partagé. Cochez enfin la case "Autoriser les utilisateurs du réseau à modifier mes fichiers" si vous souhaitez autoriser la modification et la suppression des données placées dans le répertoire partagé. Cliquez enfin sur le bouton "OK" afin de prendre en compte le partage du répertoire.



Étape 3 : Le répertoire "Micro" est maintenant partagé et tous les membres du réseau peuvent y avoir accès. Un répertoire partagé est caractérisé par cette icône.



11

L'ALTERNATIVE DU SANS FIL : WI-FI



Sommaire

11.1 Introduction	92
11.2 Présentation des réseaux sans-fil	93
11.3 Les réseaux sans-fil étendus	96
11.4 Les réseaux sans-fil spécifiques	96
11.5 Les équipements WiFi	98
11.6 Sécuriser un réseau sans fil	99

11.1 Introduction

Le système de câblage est, en effet, source de complexité et de coût. Il est également synonyme de rigidité : se déplacer d'un bureau vers une salle de réunion ou d'une pièce à l'autre, nécessite de câbler entièrement les bureaux ou sa maison.

Les concepteurs de réseaux Ethernet proposent ainsi une version sans fil de ce protocole, reposant sur les ondes radio. Les applications étant multiples, de nombreux protocoles se disputent le marché. Les réseaux Ethernet sans fil répondent aux inconvénients du câblage, mais posent d'autres problèmes inhérents aux ondes radio, telles que les interférences et la confidentialité des informations.

Confirmant une révolution technologique et culturelle des systèmes d'information initiée dans les années 2000, la mobilité d'une manière générale et de manière plus spécifique son pendant technique, les réseaux sans-fil, ont commencé à véritablement s'imposer dans le paysage informatique ivoirien en 2005.

Ces technologies, porteuses de progrès indéniables, font émerger de nouvelles façons d'accéder aux ressources informatiques et d'échanger des données. Mais cette ouverture des réseaux est à double tranchant car elle peut grandement fragiliser la sécurité du système d'information si elle se fait de manière non maîtrisée ou sans réelle prise en compte des problématiques de sécurité.

Le monde du sans fil est très vaste puisqu'il couvre de nombreuses applications qui nous sont familières : radio, télévision, téléphone et maintenant transmission de données.



D'un point de vue utilisateur, on peut classer le monde du sans fil en quatre catégories qui se distinguent par la taille de la cellule. La cellule désigne la portée de l'onde entre l'émetteur et le récepteur, en d'autres termes la distance maximale entre les deux.

Type de cellule	Large	Macro	Micro	Pico
Taille de la cellule	20 km	1 à 3 km	150 à 100 m	10 à 30 m
Position de l'antenne	Toits, pylônes	Toits, pylônes, façades	Plafonds, murs	Sac, oreillette, PDA
Application	Téléphonie (GSM, UMTS)	Boucle locale radio	Réseau local sans fil	Réseau personnel
Débit des données	Quelques centaines de Kbit/s	Quelques Mbit/s	10 à 50 Mbit/s	Quelques Mbit/s

Les technologies employées sont non seulement adaptées à ces différentes applications, mais également aux conditions réglementaires imposées dans chaque pays. En effet, le domaine des ondes radio couvre un espace de fréquence limité, allant du Kiloherz aux dizaines de Gigahertz. Cet espace doit donc être partagé, et les utilisateurs ne manquent pas : militaires, aviation civile, chaînes de télévision, stations de radio, opérateurs télécoms, entreprises et maintenant les particuliers.

Après une présentation générale des réseaux sans-fil, nous aborderons donc les problèmes de sécurité et les risques associés pour terminer sur une présentation des solutions de sécurisation.

11.2 Présentation des réseaux sans-fil

Un réseau sans-fil substitue aux habituels câbles des connexions aériennes via des ondes radios, infrarouges ou éventuellement des faisceaux laser. Cette définition, assez large, nous amène à considérer plusieurs types de réseaux sans-fil.

11.2.1 Les réseaux ad-hoc

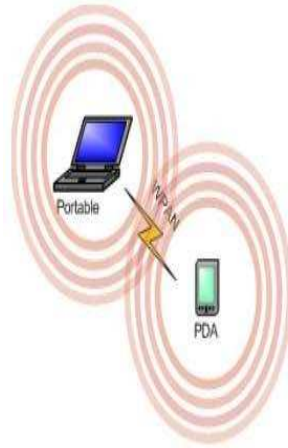
Les réseaux ad-hoc sont connus sous le nom de WPAN (Wireless Personal Area Network) ou de réseaux personnels. L'objectif de ces réseaux est de fournir une connectivité sans infrastructure dédiée. Ils sont donc exclusivement point à point et ne comptent en général que deux participants : Les terminaux mobiles friands de connectivité sans-fil comme les nouveaux téléphones portables et les PDA rassemblent la majeure partie des applications actuelles des WPANs.

11.2.1.1 Exemples :

L'échange de carte de visite ou de fichiers en infrarouge entre deux PDA, la connexion d'un PDA avec un téléphone mobile en Bluetooth pour permettre un accès Internet GPRS, la connexion sans-fil d'un PDA sur une imprimante, ...

Le réseau personnel sans fils (appelé également réseau individuel sans fils ou réseau domotique sans fils concerne les réseaux sans fils d'une faible portée : de l'ordre de quelques

dizaines mètres. Il existe plusieurs technologies utilisées pour les WPANs :



Les WPANs sont exceptionnellement variés et doivent être considérés comme des réseaux sans-fil à part entière, surtout du point de vue de la sécurité.

11.2.2 Les réseaux sans-fil locaux

Les réseaux sans-fil locaux pour terminaux mobiles, et en particulier les réseaux Wi-Fi, sont à la fois les plus répandus et les plus médiatisés à l'heure actuelle. Le réseau local sans fils (WLAN pour Wireless Local Area Network) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre-deux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes :

Le Wi-Fi (ou IEEE 802.11), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres.

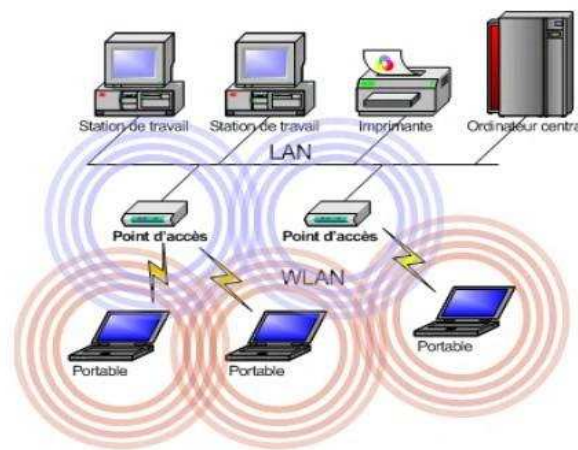
hiperLAN2 (High Performance Radio LAN 2.0), norme européenne élaborée par l'ETSI (European Telecommunications Standards Institute), permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5 150 et 5 300 MHz.

DECT (Digital Enhanced Cordless Telecommunication), norme des téléphones sans fils domestiques. Alcatel et Ascom développent pour les environnements industriels, telles les centrales nucléaires, une solution basée sur cette norme qui limite les interférences. Les points d'accès résistent à la poussière et à l'eau. Ils peuvent surveiller les systèmes de sécurité 24/24h et se connecter directement au réseau téléphonique pour avertir le responsable en cas de problème.

Un WLAN est constitué de points d'accès équipés d'une antenne et d'une interface réseau Ethernet standard. Chaque point d'accès forme une zone de couverture radio appelée cellule. L'ensemble des cellules constitue le WLAN.

Les terminaux mobiles (PC portable, PDA...) équipés d'adaptateur réseau sans-fil naviguent dans la zone de couverture du WLAN et restent connectés en permanence au réseau de l'entreprise sans contrainte physique. Ils accèdent ainsi aux ressources informatiques situées sur la partie câblée des points d'accès de la même façon que les stations de travail standards : le seul changement vient du lien physique utilisé pour la connexion.

Le passage d'un terminal de cellule en cellule sur un WLAN s'effectue via des mécanismes de roaming (itinérance) proche des systèmes utilisés sur le réseau GSM.



Pour rester dans le cadre d'un WLAN, il faut que le réseau respecte deux conditions :

- La zone de couverture utile doit être de l'ordre d'un bâtiment ou d'un site.
- L'infrastructure réseau utilisée doit être contrôlée par l'entreprise. Selon leur vocation les WLAN peuvent être :

Des WLANs privés ou d'entreprise : les terminaux mobiles servent à des employés dans l'enceinte de l'entreprise pour accéder au système d'information traditionnel. Exemple : Dans un hôpital, les médecins vont de chambre en chambre tout en accédant aux dossiers des patients en ligne et aux applications médicales depuis des PC portables.

Des WLANs publics ou hot-spots : les terminaux mobiles appartiennent dans ce cas à des clients accédant à une ressource particulière (le plus souvent un accès à Internet) proposée par le propriétaire du hot-spot. Exemple : Des hôtels, des aéroports ou des cyber-café mettent à la disposition de leurs clients un accès Internet sans-fil.

Des WLANs domestiques : un particulier forme un réseau sans-fil pour connecter un ou plusieurs PC et son accès Internet (routeur ADSL Wi-Fi par exemple).

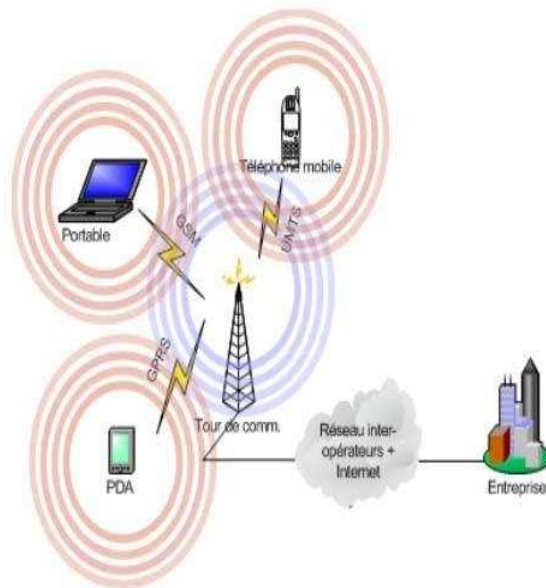
Tous ces WLANs utilisent des technologies semblables mais leur intégration est très différente.

11.3 Les réseaux sans-fil étendus

Les réseaux sans-fil étendus reposent exactement sur le même principe que les WLANs mais avec des zones de couverture nettement plus larges, allant de la ville au monde entier. Les principales technologies sont les suivantes :

- **GSM** (Global System for Mobile Communication ou Groupe Spécial Mobile)
- **GPRS** (General Packet Radio Service)
- **UMTS** (Universal Mobile Telecommunication System)
- **Wimax** (standard de réseau sans fils poussé par Intel avec Nokia, Fujitsu et Pro-wim). Basé sur une bande de fréquence de 2 à 11 GHz, offrant un débit maximum de 70 Mbits/s sur 50km de portée, certains le placent en concurrent de l'UMTS, même si ce dernier est davantage destiné aux utilisateurs itinérants.

On parle de WMANs (Wireless Metropolitan Area Network) ou de WWANs (Wireless Wide Area Network) selon les distances.



11.4 Les réseaux sans-fil spécifiques

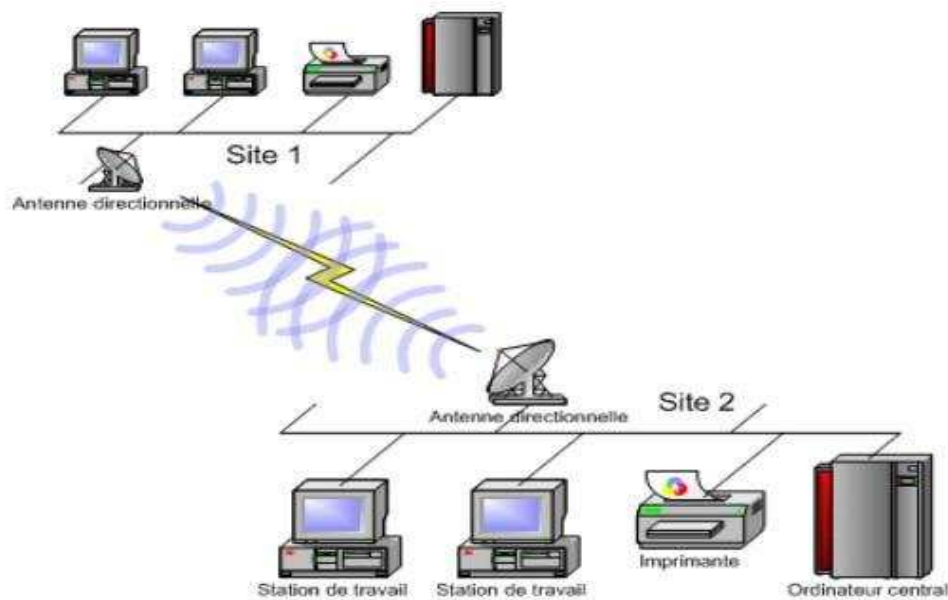
Il existe des WLANs particuliers ne concernant pas directement des utilisateurs : par exemple un réseau de caméras de surveillance sans-fil, un réseau connectant des horodateurs ou des distributeurs de boisson avec un serveur... Beaucoup de ces réseaux sont complètement nouveaux ou prennent un nouvel essor grâce aux WLANs.

On peut également considérer les systèmes de téléphones sans-fil radio basés sur DECT comme des WLANs. Une fois que la qualité de service sera mieux implémentée sur les WLANs, le mariage de la voix sur IP et des technologies Wi-Fi donnera des applications VoIPoWLAN intéressantes.

Les réseaux sans-fil de type infrastructure ne se limitent pas aux seuls réseaux de technologie Wi-Fi : tout composant mobile connecté à un point d'accès via un lien aérien est à prendre en considération.

11.4.1 Les réseaux point à point

Ce type de réseau sans-fil englobe toutes les liaisons point à point longue distance utilisées pour relier des bâtiments ou des sites distants. Ces réseaux utilisent généralement des équipements spécifiques comme des antennes directionnelles ou des technologies plus pointues (liaison laser par exemple) :



11.4.2 Les réseaux point à multipoint

Les réseaux point à multipoints sont semblables à ceux formés par les stations radios ou la télévision hertzienne traditionnelle : un émetteur diffuse une information à un nombre important de récepteurs sur des distances pouvant être étendues. Ces réseaux sans-fil particuliers restent encore très spécialisés.

La montée en puissance des normes pour les accès sans-fil à large bande passant sur de longues distances va probablement ouvrir la voie à de nouvelles applications, informatiques ou non, pour ce type de réseau.

11.4.3 Quelle utilisation pour le Wi-Fi

- Couverture d'une salle difficile d'accès par câble,
- Interconnexion de bâtiments,
- Mise en place de réseau urbain de type communautaire,
- Point d'accès public à Internet de type «Hots Spot»,
- Extension des réseaux filaires de bureau pour les utilisateurs nomades.

11.4.4 Avantages du Wi-Fi

- Internet haut débit avec mobilité garantie au bureau à domicile ou dans les espaces publics,
- Possibilité de déplacer les postes de travail n'importe où dans le bâtiment sans perdre la connectivité,
- Redéploiement facile du réseau en cas de déménagement,
- Intégration facile au réseau existant pour offrir de la mobilité aux utilisateurs,
- Flexibilité d'aménagement des locaux.

11.4.5 Inconvénients du Wi-Fi

- Solution quelque un peu plus onéreuse que le câblé,
- Bande passante partagée ce qui est conséquence des débits encore réduits,
- Sécurisation complexe.

11.5 Les équipements WiFi

Il existe différents types d'équipement pour la mise en place d'un réseau sans fil Wifi :

1. Les adaptateurs sans fil ou cartes d'accès

En anglais wireless adapters ou network interface controller, noté NIC. Il s'agit d'une carte réseau à la norme 802.11 permettant à une machine de se connecter à un réseau sans fil. Les adaptateurs WiFi sont disponibles dans de nombreux formats (carte PCI, carte PCMCIA, adaptateur USB, carte compactflash, ...). On appelle station tout équipement possédant une telle carte. À noter que les composants Wi-Fi deviennent des standards sur les portables (label Centrino d'Intel).

2. Les points d'accès

Notés AP pour Access point, parfois appelés bornes sans fil, permettant de donner un accès au réseau filaire (auquel il est raccordé) aux différentes stations avoisinantes équipées de cartes WiFi. Cette sorte de hub est l'élément nécessaire pour déployer un réseau centralisé en mode infrastructure. Certains modèles proposent des fonctions de modem ADSL et comprennent plus ou moins de fonctions comme un pare-feu.



3. Les autres

- **Smart Display** : écrans mobiles, soutenus par Microsoft.
- **Chaînes WiFi** : offrant la capacité de lire les MP3 directement sur le disque dur d'un ordinateur grâce à l'interface Ethernet sans fil intégrée. Elle préfigure toute une génération de produits, capables de lire, outre les CD audio, les radios qui émettent en MP3 sur Internet.
- **Assistant personnel** : les PDA intégrant le WiFi est parfois plus avantageux qu'un portable pour lire ses mails, importer des documents voir surfer sur le net.
- **Rétroprojecteurs** : pour des présentations avec portables mobiles.
- **Caméra video** : transmettre des images à distance à l'ordinateur qui les enregistre.

Les composants Wi-Fi ne sont pas plus onéreux que ceux des réseaux filaires, bientôt toutes les plates-formes seront vendues avec des modules Wi-Fi intégrés. C'est déjà le cas dans le monde des PC portables, qui, sous l'impulsion d'Intel, fait sa révolution sans fil grâce au Centrino.

11.6 Sécuriser un réseau sans fil

Avec les outils et technologies actuels, il est très simple, très pratique et peu coûteux de créer des réseaux sans fil. Voici quelques conseils pour "sécuriser" un minimum votre réseau sans fil.

Le temps du réseau filaire n'est pas encore révolu mais il est moins d'actualité que le réseau sans fil Wireless Fidelity, plus connu sous le nom de Wi-Fi. Sur le réseau filaire, la sécurité était un problème beaucoup moins important puisqu'il fallait que le Hacker (pirate) se connecte sur le réseau en local physiquement. Maintenant, il suffit au pirate de se trouver à proximité d'une source émettrice, un point d'accès sans fil.

Ces points d'accès émettent un signal Wi-Fi dans un rayon de quelques mètres à plusieurs centaines de mètres, suivant leurs qualités et leurs performances. Il suffit alors d'un ordinateur fixe ou ordinateur portable équipé d'une carte sans fil Wi-Fi pour détecter et se connecter sur ce réseau "invisible".

Actuellement, les différents constructeurs de périphériques réseaux ou même les FAI (Fournisseurs d'Accès Internet) fournissent à leurs clients des solutions sans fil. Bon nombre d'entre eux ne connaissent rien en informatique ou ne savent pas comment configu-

rer correctement un réseau sans fil, alors ils laissent les paramètres par défaut. L'erreur est là ! La majorité de ces réseaux ne sont pas bien configurés et sont "ouverts" au "monde extérieur" : diffusion dans tout le quartier, acceptation de n'importe quel ordinateur et trafic réseau lisible par tout le monde (données reçues et envoyées).

Ce sont quelques conseils qui n'apportent pas la sécurisation à 100% de votre réseau mais va lui permettre de réduire un bon nombre d'accès non autorisés.

Index

ADSL, 81

balise, 10, 12, 14

Bluetooth, 93

Browser, 5

checkbox, 32

commentaire, 10

concentrateur, 79

connecteur, 85

Ethernet, 85

FAI, 99

Firefox, 5

Formulaires, 42

FTP, 84

GSM, 96

HTML, 4–7, 10, 12

Hub, 79

hypertexte, 27

IEEE, 83

interactivité, 21

IP, 84

Konqueror, 5

LAN, 72, 84

Lynx, 5

Mozilla, 5

MySQL, 51, 61

Navigateur, 5

Netscape, 5

Opéra, 5

Passerelles, 81

PDA, 93

PHP, 7, 63

phpMyAdmin, 65

programmation, 4

Requêtes, 42

requêtes, 56

RJ45, 86, 88

Routeurs, 81

scripts, 14

Serveur, 4, 6

SGBD, 39, 51, 56

SQL, 57

Switch, 80

TCP/IP, 84

Topologie, 73

WAN, 73, 84

webmaster, 87

Wi-Fi, 99

Wimax, 96

WPAN, 93

WYSIWYG, 4