



# Technologie Web Avancée

## PHP/MYSQL

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).



# Plan du Chapitre

- **1. PHP**
- **2. MYSQL**
- **3. PHP/MYSQL**



# Technologie Web Avancée

## PARTIE 1 : PHP

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).



# Plan du Chapitre

- 1. Introduction
- 2. Variable
- 3. Operateur
- 4. Affichage
- 5. Commentaire
- 6. Structure de Contrôle
- 7. Tableau
- 8. Fonction
- 9. Inclusion
- 10. Redirection

# INTRODUCTION

PHP est un langage de programmation qui s'intègre dans vos pages HTML. Il permet entre autres de rendre automatiques des tâches répétitives, notamment grâce à la communication avec une base de données (utilisation la plus courante de PHP).

Intégration d'un script dans une page :

- ❖ Les pages web sont au format HTML. Les pages web dynamiques générées avec PHP sont au format PHP. Le code source PHP est directement insérer dans le fichier html entre cette syntaxe :

**<?php ... ?>**

## 1. Variables

- Les identificateurs de variable sont précédés du symbole « \$ ».

Exemple : **\$toto**.

## 2. TYPE D' ENTIER

- Les variables peuvent être de type entier (**integer**), réel (**double**), chaîne de caractères (**string**), tableau (**array**), objet (**object**), booléen (**boolean**).

### 2.1 Constant

L'utilisateur peut définir des constantes dont la valeur est fixée une fois pour toute. Les constantes ne portent pas le symbole \$ (dollars) en début d'identificateur et ne sont pas modifiables.

**define("var",valeur)** : définit la constante **var** (sans \$) de valeur **valeur**

*Exemple :*

```
define("author","Djama");  
echo author;
```

```
// affiche Djama
```

## 3. OPERATEUR

Opérateurs arithmétiques :

**+** (addition), **-** (soustraction), **\*** (multiplié), **/** (divisé), **%** (modulo), **++** (incrément), **--** (décrément). Ces deux derniers peuvent être pré ou post fixés

Opérateurs d'assignement :

**=** (affectation), **\*=** (**\$x\*=\$y** équivalent à **\$x=\$x\*\$y**), **/=**, **+=**, **-=**, **%=**

Opérateurs logiques :

**and**, **&&** (et), **or**, **||** (ou), **xor** (ou exclusif), **!** (non)

Opérateurs de comparaison :

**==** (égalité), **<** (inférieur strict), **<=** (inférieur large), **>**, **>=**, **!=** (différence)

Opérateurs de concaténation :

## 4. Affichage

Les fonctions d'affichage :

**echo()** : écriture dans le navigateur ou **print()** : écriture dans le navigateur

Exemple :

`<?php echo 'Bonjour le monde !' ; ?>` **ou** `< ?php print 'Bonjour le monde !' ; ?>`

# Commentaire

Comme dans tout autre langage de programmation, PHP dispose sa propre syntaxe de commentaires.

Les commentaires permettent de bien expliquer une partie du code afin qu'on puisse déchiffrer en cas de maintenance ou de travail collaboratif (plusieurs développeurs travaillant sur le même programme).

Il existe deux sortes de commentaire. Le commentaire sur une seule ligne et le commentaire multilignes.

Une seule ligne	Multilignes
<pre>&lt;?php // Ceci est un premier commentaire sur une ligne echo 'Hello World !'; ?&gt;</pre>	<pre>&lt;?php /* Ce programme a été écrit par Dr. Kadar Il affiche la chaîne 'Hello Word !' à l'écran */ echo 'Hello World !'; ?&gt;</pre>



# 3. STRUCTURE DE CONTROLE (1/2)

- Structures conditionnelles

```
if(condition1) {  
    # instructions à exécuter si la  
    condition1 est vraie...  
}  
elseif (condition2) {  
    instructions à exécuter si la  
    condition2 est vraie...  
}  
elseif (condition3) {  
    instructions à exécuter si la  
    condition3 est vraie...  
}  
else {  
    instructions à exécuter si  
    aucune des conditions n'est  
    vraie...  
}
```

```
switch(expression) {  
    case resultat1 :
```

```
    # instructions à exécuter si  
    l'expression vaut resultat1...  
    break ;
```

```
    case resultat2 :
```

```
    # instructions à exécuter si  
    l'expression vaut resultat2...  
    break ;
```

```
    .....
```

```
    default :
```

```
    # instructions à exécuter en  
    dernier recours...  
}
```

# 3. STRUCTURE DE CONTROLE (2/2)

## ○ Structures de boucles

**for**(expression1 ; condition ; expression2) {  
# instructions à exécuter tant que la condition est vraie...

}

**Exemple : compter jusqu'à 10**

```
for ($i = 1 ; $i <= 10 ; $i++) {  
echo "$i " ;  
}
```

Affiche 1 2 3 4 5 6 7 8 9 10.

**While** (condition){

# instructions à exécuter tant que la condition est vraie...

}

**Exemple : compter jusqu'à 10**

```
$i = 1 ;  
while ($i <= 10) {  
echo "$i " ;  
$i++ ;  
}
```

Affiche 1 2 3 4 5 6 7 8 9 10.

**do** {

# instructions à exécuter tant que la condition est vraie...

} **while**(condition);

**Exemple : compter jusqu'à 10**

```
$i = 1 ;  
do {  
echo "$i " ;  
$i++ ;  
} while ($i <= 10) ;
```

Affiche 1 2 3 4 5 6 7 8 9 10.

## 7. TABLEAU (1/2)

- × Une variable tableau est de type **array**. Un tableau accepte des éléments de tout type.
  - + Exemple :

- × `$tab_colors = array('red', 'yellow', 'blue', 'white');`
  - × `$tab = array('djama', 2002, 20.5, $name);`

- × Indice initiale d'un tableau est 0
- × Parcours d'un tableau :
  - + *Exemple 1 :*

```
$i=0;
while($i <count($tab)) {           // count() retourne le nombre d'éléments
    echo $tab[$i].'\n';
    $i++;
}
```

+ *Exemple 2 :*

```
foreach($tab as $i) {
    echo $i."\n";
}
```

- × La variable **\$elem** prend pour valeurs successives tous les éléments du tableau **\$tab**.

## 7. TABLEAU (2/2)

- × Quelques fonctions utiles:
  - + **count(\$tab)** : retournent le nombre d'éléments du tableau
  - × **foreach(\$tab as \$elem)**: permet a la variable **\$elem** de prendre pour valeurs successives tous les éléments du tableau **\$tab**.
  - + **in\_array(\$var,\$tab)** : dit si la valeur de **\$var** existe dans le tableau **\$tab**
  - + **sort(\$tab)** : trie alphanumérique les éléments du tableau
  - + **strlen(\$tab)** : Calcule la taille d'une chaîne
  - + **empty(\$tab)** : Détermine si une variable est vide ou non
  - + **isset(\$tab)** : Déterminer si une variable est bien définie

## 8. FONCTION

- Les fonctions peuvent prendre des arguments dont il n'est pas besoin de spécifier le type. Elles peuvent de façon optionnelle retourner une valeur.

### *Exemple :*

```
<?php function calculerSomme($ a, $ b)
{ // Déclaration de la variable locale
    $somme = $a + $b;
return $somme;
}
// Affichage du résultat
echo calculerSomme($ a, $ b);
?>
```

```
calculerSomme(8,10);
```

## 5. INCLUSION

- × On peut inclure dans un script php le contenu d'un autre fichier.
  - + **include** évalue et insert à chaque appel (même dans une boucle) le contenu du fichier passé en argument.
    - × *Exemple :*
      - \* **include**("fichier.php");
  - + **require** insert dans le code le contenu du fichier spécifié même si ce n'est pas du code php. Est équivalent au préprocesseur *#include* du C.
    - × *Exemple :*
      - \* **require**("fichier.php");

## 20. REDIRECTION

- `<?php`  
`header("Location: home2.php");`  
`exit();`
- `?>`
- Ce script effectue une redirection vers une autre page. C'est-à-dire que le navigateur du client en voyant cet entête *Location* va charger directement la page indiquée sans regarder la suite du script. La fonction **exit** est là pour parer au cas impossible où le script continuerait son exécution.



# Technologie Web Avancée

## Partie 2 : MYSQL

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).





# Plan du Chapitre

- **1. Présentation**
- **2. Création/sélection d'une base de données**
- **3. Gestion de la base de données**
- **4. Affichage d'une table**

# INTRODUCTION

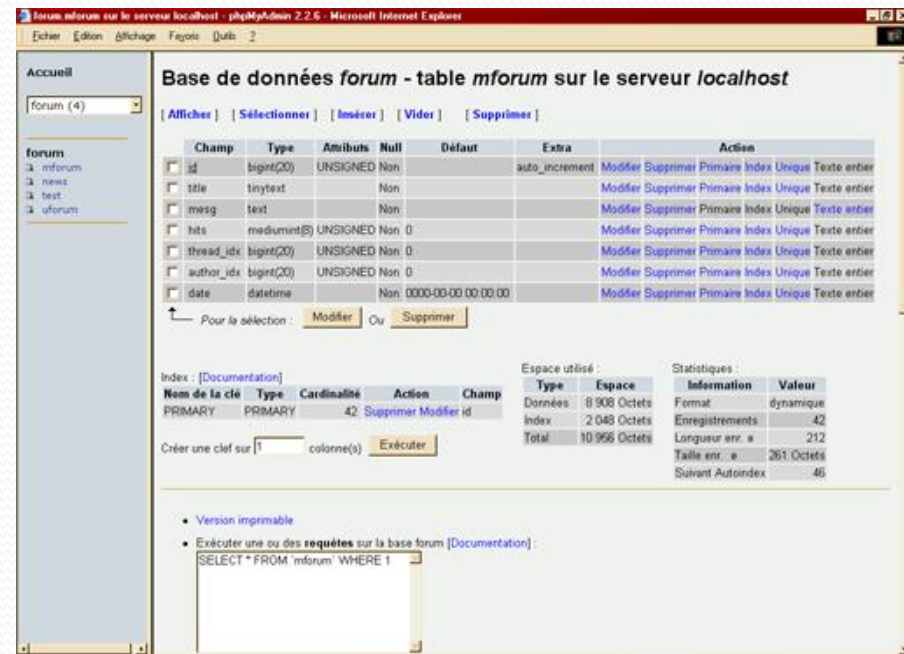
- MySQL est une base de données implémentant le langage de requête SQL un langage relationnel très connu.
- Il existe un outil libre et gratuit développé par la communauté des programmeurs libres : phpMyAdmin qui permet l'administration aisée des bases de données MySQL avec php. Il est disponible sur :  
<http://sourceforge.net/projects/phpmyadmin/> et  
<http://www.phpmyadmin.net>

# I. Presentation

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

- Cet outil permet de :

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



## 2. Création/sélection d'une base de données

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



ETAPE 2 : sélectionnez le nom de la base à manipuler (le nombre de tables de la base apparaît entre parenthèses)

ETAPE 1 : Écrivez le nom de la base de donnée à créer. Puis cliquez sur « Créer »

## 3. Gestion de la base de données

forum sur le serveur localhost - phpMyAdmin 2.2.6 - Microsoft Internet Explorer

Echier Edition Affichage Favoris Outils 2

Accueil

forum (4)

forum

- mforum
- news
- test
- uforum

Base de données *forum* sur le serveur *localhost*

Table	Action	Enregistrements	Type	Taille
<input type="checkbox"/> mforum	Afficher Sélectionner Insérer Propriétés Supprimer Vider	42	MyISAM	10,7 Ko
<input type="checkbox"/> news	Afficher Sélectionner Insérer Propriétés Supprimer Vider	4	MyISAM	2,6 Ko
<input type="checkbox"/> test	Afficher Sélectionner Insérer Propriétés Supprimer Vider	7	MyISAM	2,1 Ko
<input type="checkbox"/> uforum	Afficher Sélectionner Insérer Propriétés Supprimer Vider	5	MyISAM	2,7 Ko
4 table(s)	Somme	58	--	18,1 Ko

Tout cocher / Tout décocher Pour la sélection :

Version imprimable

Exécuter une ou des requêtes sur la base forum [Documentation]

☒ Réafficher la requête après exécution

Où Emplacement du fichier texte :

Parcourir...

Exécuter

- 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



# Affichage d'une table

Base de données *forum* - table *mforum* sur le serveur *localhost*

Affichage des enregistrements 3 - 5 (42 total)

requête SQL : [Modifier]

SELECT \* FROM 'mforum' LIMIT 3,3

Rappel de la requête

Rappel de la base, de la table et du serveur

<< < Afficher : 3 lignes à partir de 5 > >>  
en mode horizontal et répéter les en-têtes à chaque groupe de 100

Colonnes = noms des attributs de la table

		id	title	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

Liste des enregistrements de la table par pages de X lignes

<< < Afficher : 3 lignes à partir de 5 > >>  
en mode horizontal et répéter les en-têtes à chaque groupe de 100

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

Insérer un nouvel enregistrement



# Technologie Web Avancée

## Partie 3 : PHP/MYSQL

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).



# Plan du Chapitre

- **1. Connexion**
- **2. Envoi de données**
- **3. Récupération des données**



# 2. CONNEXION

- × Pour se connecter à une base depuis PHP, il faut spécifier un nom de serveur, un nom d'utilisateur, un mot de passe et un nom de base de données.
  
- × Les fonctions de connexion :
  - + `mysql_pconnect($server,$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
  
  - + `mysql_select_db($base)` : permet de choisir la base `$base`, retourne `TRUE` en cas de succès, sinon `FALSE`

# CONNEXION

Exemple :

```
<html>
<head>
</head>
<title> exemple d'application sur php </title>
```

```
<?php
```

```
$serveur    = "localhost";
$utilisateur = "root";
$motDePasse = "";
$base       = "mesn";
```

```
$sum=mysql_pconnect($serveur, $utilisateur , $motDePasse)
    or die("Impossible de se connecter au serveur de bases de données.");
mysql_select_db($base)
    or die("Base de données non trouvée.");
```

## 2. ENVOI DES DONNEES

- ❖ Pour envoyer des données dans 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.

### *Exemple :*

```
/* récupération des données envoyés par le formulaire */
```

```
$email = $_POST["email_visiteur"];
```

```
$avis = $_POST["avis_visiteur"];
```

```
/* Création de la requête */
```

```
mysql_query("INSERT INTO commentaire ( mail,opinion)  
VALUES ('$email','$avis')");
```

```
?>
```

# 2. RECUPERATION DES DONNEES

- Pour récupérer des données à partir d'une base de donnée, il existe la fonction **mysql\_fetch\_row(\$result)** qui retourne une ligne de résultat sous la forme d'un tableau. Les éléments du tableau étant les valeurs des attributs de la ligne. Retourne FALSE s'il n'y a plus aucune ligne.

## *Exemple :*

```
/* Création de la requête */  
$result = mysql_query("SELECT * FROM commentaire");  
  
/* Parcours de la requête */  
while($ligne = mysql_fetch_row($result)) {  
    $email = $ligne[0];  
    $avis = $ligne[1];  
  
/* affichage de la requête */  
    echo "$email, $avis <br />";  
}  
?>
```

- Ici, on accède aux valeurs de la ligne par leur indice dans le tableau.

# Session

- Une session est un mécanisme technique permettant de sauvegarder **temporairement sur le serveur** des informations relatives à un internaute.
- La sécurité et l'intégrité des données sont améliorées tout au long de la session.
- Contrairement à une base de données ou un système de fichiers, la session conserve les informations pendant quelques minutes.
- Les sessions sont particulièrement utilisées pour ce type d'applications :
  - Les espaces membres et accès sécurisés avec authentification.
  - Gestion d'un caddie sur un site de vente en ligne.

- Pour utiliser les sessions, différentes fonctions PHP s'offrent à nous. Voici déjà un petit tableau vous permettant de vous familiariser avec ces différentes fonctions (que nous détaillerons bien sur dans la suite de ce tutorial):

Fonction	Signification
<code>session_start</code>	Démarre une session
<code>session_register</code>	Enregistre une variable de session
<code>session_unregister</code>	Efface une variable de session
<code>session_id</code>	Retourne l'id de la session en cours
<code>session_name</code>	Retourne le nom de la session en cours
<code>session_unset</code>	Detruit toutes les variables de la session en cours
<code>session_destroy</code>	Detruit la session en cours

# Exemple d'Application

## Etape 1: Connexion

### Connexion

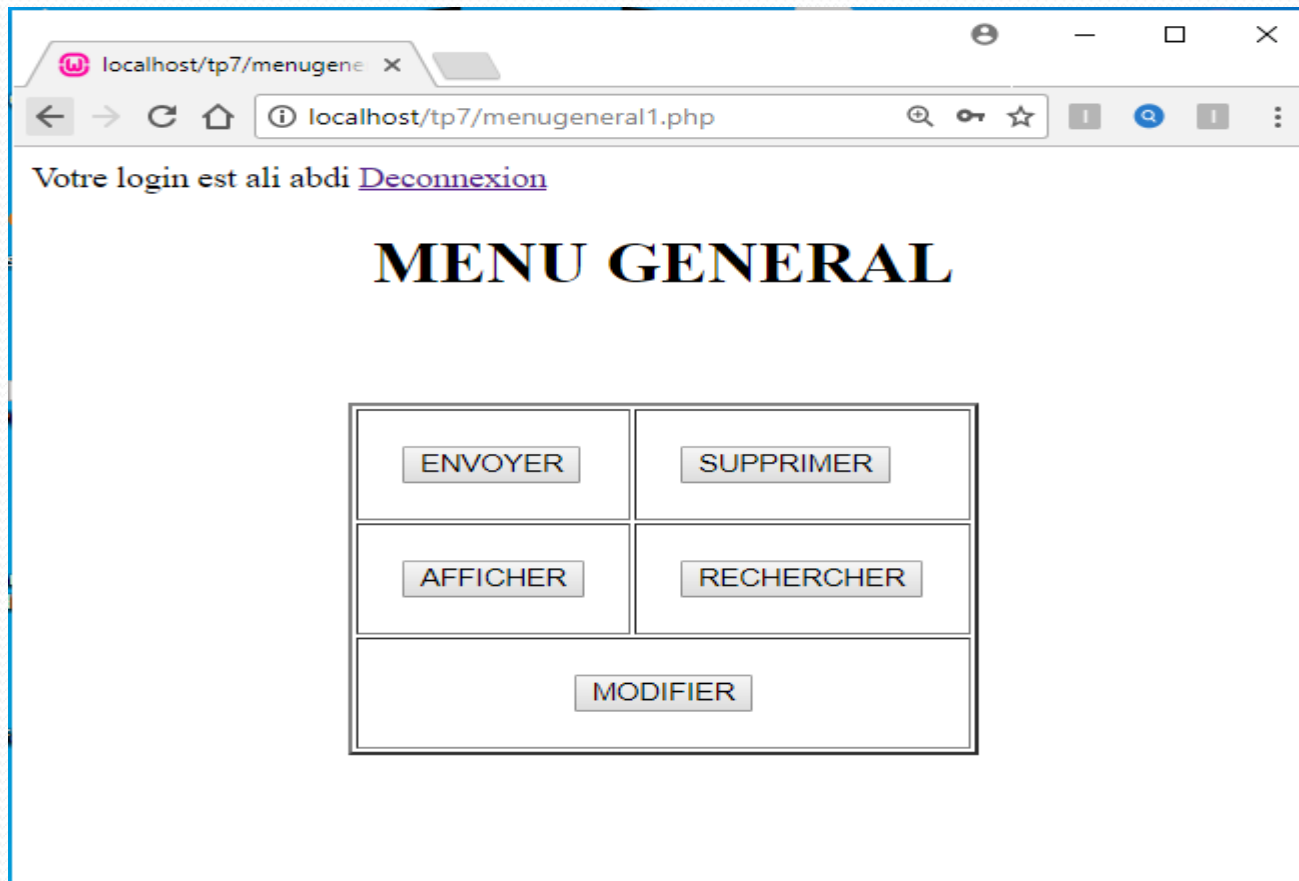
Nom :	<input type="text"/>
password :	<input type="password"/>
	<input type="button" value="Annuler"/> <input type="button" value="Connexion"/>

# Étape 1: Connexion

- // on démarre la session  
`session_start ();`
- // on enregistre les paramètres de visiteur comme variables de session (\$login et \$pwd) (notez bien que l'on utilise pas le \$ pour enregistrer ces variables)  
`$_SESSION['login'] = $_POST['nom'];`  
`$_SESSION['pwd'] = $_POST['password'];`
- // on redirige le visiteur vers une page du section membre  
`header("Location: menugeneral1.php");`



# Etape 2: Menu General



# Etape 2: Menu General

// On démarre la session (ceci est indispensable dans toutes les pages de notre section membre)

```
session_start ();
```

echo ‘

// On récupère nos variables de session

```
Votre login est '$_SESSION['login'].
```

# Etape 3: DECONNEXION

```
<?php
// On démarre la session
    session_start ();

// On détruit les variables de notre session
    session_unset ();

// On détruit notre session
    session_destroy ();

// On redirige le visiteur vers la page d'accueil
    header ('location: connexion.html');

?>
```



# CHAPITRE 2

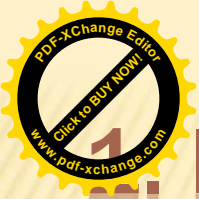
# LANGAGE HTML



# PLAN DU CHAPITRE

1. INTRODUCTION
2. TEXTE
3. TITRE ET LISTES
4. LIENS
5. IMAGES
6. ARRIERE PLAN
7. TABLEAU
8. FORMULAIRE
9. CADRES





# 1 INTRODUCTION

HTML est le langage universel utilisé pour communiquer sur le Web. Votre information sera ainsi transportée sur cette gigantesque toile de réseaux interconnectés qu'est Internet, pour aboutir sur l'ordinateur de votre lecteur grâce à un programme appelé navigateur ou browser.

Ce logiciel, que l'on appelle un browser ou navigateur, vous permet de surfer sur le Net et d'afficher sur votre écran les pages qu'il a interceptées.

Il y a, hélas, beaucoup de marques et de types de browsers différents. Des simples, des archaïques ou des sophistiqués... Les plus connus sont FireFox de Mozilla, Internet Explorer de Microsoft et Netscape dans sa version 2, 3 et 4.

# 1.1 QU'EST CE QUE LE LANGAGE HTML

Le langage Html étant un ensemble de balises et d'attributs, il nous paraît utile sinon indispensable de les passer en revue et surtout de les visionner au moins une fois car :

- ✗ si les éditeurs Html vous faciliteront grandement la tâche, ils ne sont pas toujours parfaits surtout lors des modifications, annulations ou suppressions en cours de travail. Il vous faudra bien alors vous plonger dans le code source pour corriger les dysfonctionnements.
- ✗ les codes source de vos pages préférées sont disponibles (et sans copyright). Il est alors possible de s'en inspirer pour reprendre le procédé sans avoir à réinventer.
- ✗ ces mêmes éditeurs Html vous proposeront des termes comme " En-tête, Heading, Cell spacing, Numered List..." qui sont propres au langage Html.
- ✗ vous aurez besoin d'une connaissance pointue du Html pour inclure les codes du PHP, Javascript ou VBscript dans vos pages.



## 4.2 PREMIÈRES BALISES

`<HTML>` Ceci est le début d'un document de type HTML.

`</HTML>` Ceci est la fin d'un document de type HTML.

`<HEAD>` Ceci est le début de la zone d'en-tête (prologue au document proprement dit contenant des informations destinées au browser).

`</HEAD>` Ceci est la fin de la zone d'en-tête.

`<TITLE>` Ceci est le début du titre de la page.

`</TITLE>` Ceci est la fin du titre de la page.

`<BODY>` Ceci est le début du document proprement dit.

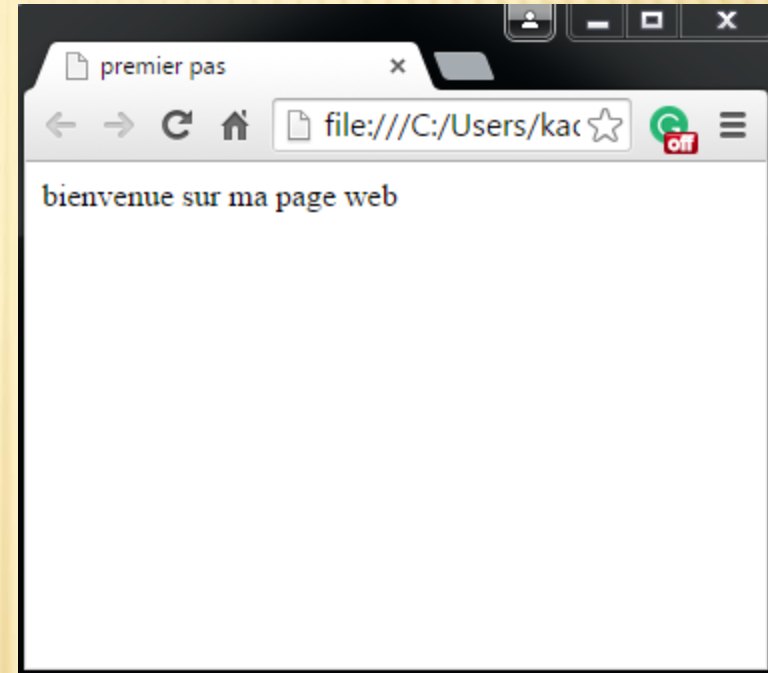
`</ BODY>` Ceci est la fin du document proprement dit.

- ✗ Vous aurez remarqué qu'à chaque balise de début d'une action, soit `<...>`, correspond (en toute logique) une balise de fin d'une action `</...>`.
- ✗ Vous noterez aussi que les balises ne sont pas "case sensitive". Il est donc équivalent d'écrire `<HTML>`, `<html>`, `<Html>`, etc



# 1.3 PREMIER PAS HTML:

- Ouvrir l'éditeur de texte.
- Ecrire les codes Html suivants:  
<HTML>  
<HEAD>  
<TITLE>premier pas </TITLE>  
</HEAD>  
<BODY> bienvenue sur ma page web  
</BODY>  
</HTML>
- Enregistrer le document avec l'extension .html ou .htm.
- cliquer sur la page web pour visualiser



# TEXTE

Tout document Html contiendra en majorité du texte. Voyons comment l'agrémenter par quelques balises élémentaires.

- ✗ Gras [Bold] `<B>...</B><STRONG>...</STRONG>` Début et fin de zone en gras
- ✗ Italique [Italic] `<I>...</I>` Début et fin de zone en italique
- ✗ Souligner [Underline] `<U>.....</U>` Début et fin de zone souligné
- ✗ Police [Font face] `<FONT face="Times New Roman"></FONT>` Début et fin de zone en police Times New Roman
- ✗ Taille de caractère [Font size] `<FONT SIZE=?>...</FONT>` Début et fin de zone avec cette taille
- ✗ Couleur de caractère [Font color] `<FONT COLOR="#0000FF"></FONT>` Début et fin de zone avec cette couleur
- ✗ A la ligne [Line break] `<BR>` Aller à la ligne
- ✗ Commentaires [Comments] `<!-- *** -->` Ne pas afficher
- ✗ Centrage [Center] `<CENTER></CENTER>` Centrer
- ✗ Pour aligner du texte `<DIV align=left>...</DIV> <DIV align=center>...</DIV> <DIV align=right>...</DIV>`

Voici les codes de quelques couleurs basiques.

- ✗ Bleu #0000FF Vert #00FF00
- ✗ Blanc #FFFFFF Violet #8000FF
- ✗ Rouge #FF0000 Jaune #FFFF00
- ✗ Gris clair #COCOCO Noir #000000

# LES TITRES ET LES LISTES

Tout document d'une certaine consistance se doit de présenter, par exemple dans la table des matières, les différents niveaux de son exposé. Html a disposé dès son origine d'outils spécialement conçus à cet effet.

- ✗ En-têtes [Heading] `<Hn></Hn>` avec  $n=1$  à 6 Afficher une en-tête de niveau  $n$  et sauter une ligne
- ✗ Liste non-ordonnée [Bullet list] `<UL></UL>` Afficher le texte sous forme d'une liste non-ordonnée  
`<UL type="square"></UL>` = ■ ; `<UL type="disc"></UL>` = ● ; `<UL type="circle"></UL>` = ○
- ✗ Liste ordonnée [Numbered list] `<OL></OL>` Afficher le texte sous forme d'une liste ordonnée.  
`<OL type="1"></OL>` ; `<OL type="I"></OL>` ; `<OL type="A"></OL>`
- ✗ Élément de liste [List items] `<LI>` Voici un élément de la liste
- ✗ Paragraphe [Paragraph] `<P></P>` Saut de ligne, insérer une ligne vierge et commencer un paragraphe

Exemple:

- ✗ `<H1>Les mois du printemps</H1>`
- ✗ `<UL type="disc"><LI>avril`
- ✗ `<LI>mai`
- ✗ `<LI>juin</UL>`
- ✗ `<P>`
- ✗ `<H3>Les mois d'automne</H3>`
- ✗ `<OL type="1"><LI>janvier`
- ✗ `<LI>fevrier`
- ✗ `<LI>mars</OL>`





# LES LIENS

Html (Hyper Text Markup Language) est un langage hypertexte (et hypergraphique) qui vous permet en cliquant sur un mot, généralement souligné (ou une image) de vous Transporter;

- ✗ vers un autre endroit du document.
- ✗ vers un autre fichier Html situé sur votre ordinateur.
- ✗ vers un autre ordinateur situé sur le Web.

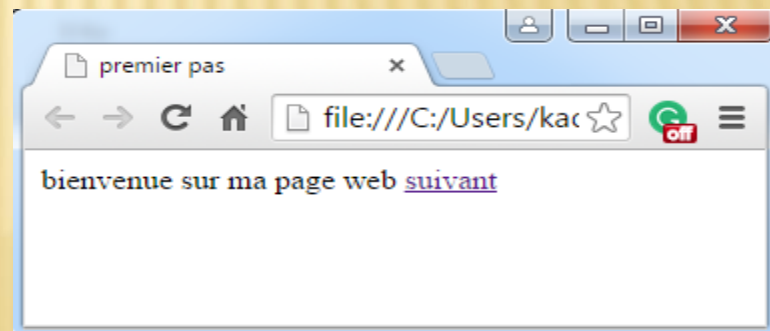
Ce système d'hypertexte vous est familier car il est également utilisé par les fichiers d'aide de Windows. Ce sont ces liens qui vous permettent de surfer de page en page et qui constituent l'essence des documents Html.

- ✗ La syntaxe de ces liens entre plusieurs pages, est simple mais entraînera de nombreux commentaires :

`<A HREF="URL ou adresse">...</A>`

Exemple:

`<A HREF="exempleliste.html">suivant</A>`



# LES IMAGES

## GIF ou JPEG?

Avant de passer aux balises, il faut savoir ce qui suit.

- ❖ Les deux formats d'image (maximum 256 couleurs) reconnus sur le Web, sont le format GIF (version 89a) et le format JPEG. Pour le format GIF, on retiendra la caractéristique "entrelacé" qui permet de charger progressivement l'image lors de l'ouverture de la page. Ces deux formats donnent des résultats assez équivalents bien que JPG soit plutôt recommandé pour des images avec des tons nuancés ou dégradés.
- ❖ · L'ennemi sur le Web, c'est la taille des images! Plus l'image sera grande, plus le temps de chargement sera long... au risque de décourager vos visiteurs. Si cela est possible, une image en 16couleurs peut très bien faire l'affaire.
- ❖ · Il n'est pas inutile de prévoir dans votre panoplie de compositeur Web, un logiciel de retouche d'images. A ce propos « photoshop" a la triple qualité d'être disponible en shareware, d'être très intuitif et d'être performant. En outre, un tel logiciel vous permettra de composer vos propres images et ainsi de personnaliser vos pages.

# LES IMAGES

Le code Html est :

- ✗ `<IMG SRC="Adresse de l'image">` Afficher l'image qui se trouve à l'adresse...
- ✗ La balise image possède de nombreux attributs.
- ✗ Dimensions `width=? height=?` Hauteur et largeur (en pixels)
- ✗ `border=?` (en pixels) Bordure
- ✗ `align=top`
- ✗ `align=middle`
- ✗ `align=bottom`
- ✗ `align=left`
- ✗ `align=right`

Exemple :

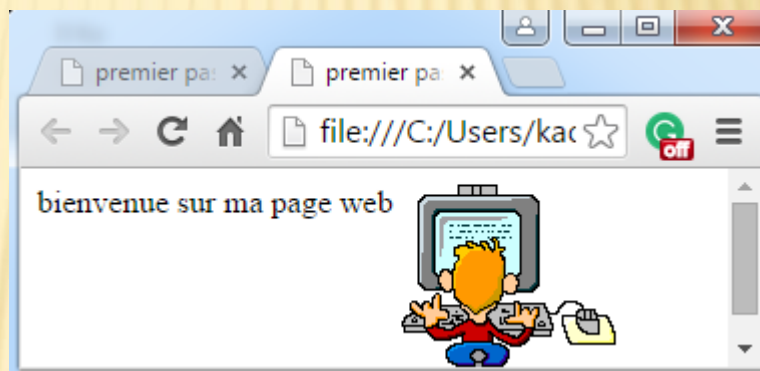
`<IMG SRC="pirate.gif" align=TOP>`

Lien sur image

Les balises sont :

- ✗ `<A HREF="fichier.html"><IMG SRC="image.gif"></A>`

Remarquons que les images cliquables sont entourées d'une bordure ■





# LES ARRIÈRE-PLANS

✖ Le langage Html permet d'agrémenter la présentation du document d'un arrière-plan [background] coloré ou composé d'une image. Ce qui apporte un élément "artistique" à votre page. La balise à utiliser ne pose pas de problème : Couleur d'arrière-plan `<BODY BGCOLOR="#$$$$$$">`.

Il existe d' autres balises qui sont prevus pour modifier les couleurs :

- ✖ Couleur de texte `<BODY TEXT="#$$$$$$">`
- ✖ Couleur de lien `<BODY LINK="#$$$$$$">`
- ✖ Lien visité `<BODY VLINK="#$$$$$$">`
- ✖ Lien actif `<BODY ALINK="#$$$$$$">`

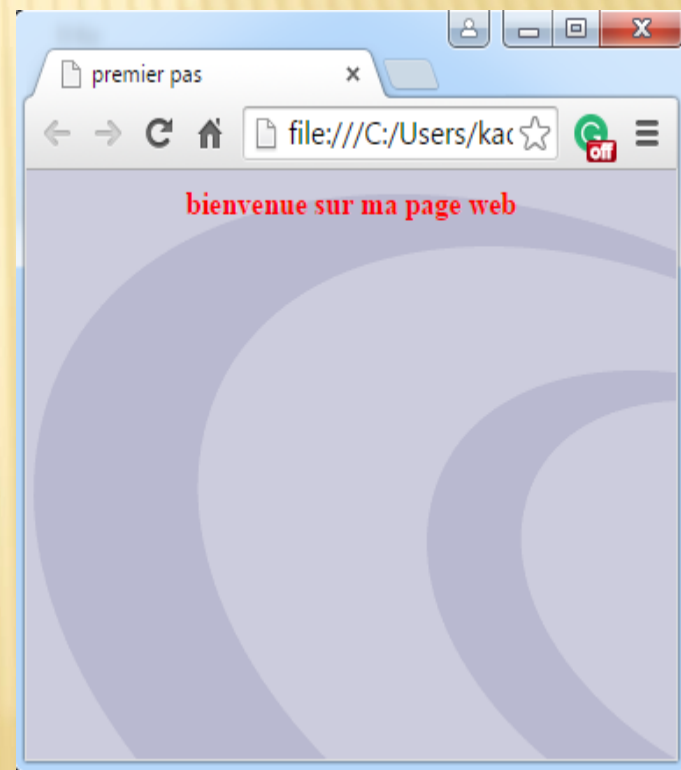
Reprenons notre exemple en modifiant la balise body

- ✖ `<BODY BGCOLOR="#000088" TEXT="#FFFF00">`

On peut aussi prévoir un fond en image.

Exemple :

Texture d'arrière-plan `<BODY BACKGROUND="Adresse de l' image">`



# 7. LES TABLEAUX

En Html, les tableaux servent non seulement à aligner des chiffres mais surtout à placer des éléments à l'emplacement que vous souhaitez. L'usage des tableaux est donc très fréquent.

Un tableau est composé de lignes et de colonnes qui forment les cellules du tableau. Les balises de base sont donc :

Définition du tableau	[Table]	<TABLE></TABLE>	Début et fin de tableau
Définition d'une ligne	[Table Row]	<TR></TR>	Début et fin de ligne
Définition d'une cellule	[Table Data]	<TD></TD>	Début et fin de cellule

Exemple d'application : Si vous souhaitez y adjoindre des bordures :

Bordure de cadre [Border] <TABLE border=?></TABLE>

<TABLE border=1>

<TR><TD>1</TD><TD>2</TD></TR>

<TR><TD>3</TD><TD>4</TD></TR>

</TABLE>



1	2
3	4



# LES TABLEAUX

Il y a encore trois éléments (définis par défaut mais modifiables) :

L'espace entre les cellules ou

l'épaisseur des lignes du quadrillage

L'enrobage des cellules ou

l'espace entre le bord et le contenu

La largeur de la table

<TABLE cellpadding=?>

<TABLE cellspacing=?>

<TABLE width=?> <TABLE width=%>

## Exemple d'application :

```
<TABLE border=2 cellpadding=10>
```

```
<TR><TD>1</TD><TD>2</TD></TR>
```

```
<TR><TD>3</TD><TD>4</TD></TR>
```

```
</TABLE>
```

1	2
3	4

## Les cellules des tableaux

Avant toutes choses, les cellules peuvent contenir tous les éléments Html déjà passés en revue soit :

- du texte
- des images
- des liens
- des arrière-plans

# LES TABLEAUX

Bien que l'allure de votre tableaux soit déjà déterminée, chaque cellule est en quelque sorte un petit univers à part qui a ses propres spécifications. Découvrons les balises.

Largeur d'une cellule	<code>&lt;TD width=?&gt;</code> en pixels <code>&lt;TD width=?&gt;</code> en pourcentage
Fusion de lignes	<code>&lt;TD rowspan=?&gt;</code>
Fusion de colonnes	<code>&lt;TD colspan=?&gt;</code>

## Exemple d'application :

```

<CENTER><TABLE width=60% border=1>
<TR>
<TD colspan=3 align=center>Tarif </TD>
</TR>
<TR>
<TD width=33% >Article 1</TD>
<TD width=33%>Ref 002 </TD>
<TD width=34% align=right>30fr</TD>
</TR>
</TABLE></CENTER>

```

Tarif		
Article 1	Ref 002	30fr

# 9. Les Formulaires

Un formulaire *HTML* est une section d'un document pouvant contenir des balises et des contrôles. Les contrôles sont des éléments spéciaux qui permettent à l'utilisateur de fournir des données. Les contrôles tels que les cases à cocher, les boutons radio, les labels, les menus, etc. Lorsque l'utilisateur a complété le formulaire, il clique sur un bouton pour envoyer les données qui seront traitées par le serveur.

Insérer un formulaire commence toujours par le placement de :

```
<FORM NAME=" nom du formulaire" METHOD="post" ACTION="mailto:votrepseudo@votreprovider.com" >  
Ici vos champs de choix.  
</FORM>
```

Les différents champs (à mettre entre les balises <FORM> et </FORM> sont (par exemple) :

Champ de ligne simple : <INPUT TYPE="text" NAME="sonnom" SIZE="25" MAXLENGTH="50" VALUE="Entrez ici votre nom">

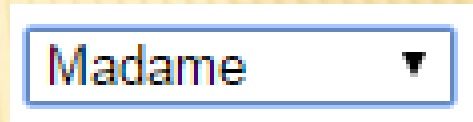
Champ de ligne multiple: <textarea rows="6" name="avis\_visiteur" cols="29"></textarea>



# Les Formulaires

Une liste déroulante :

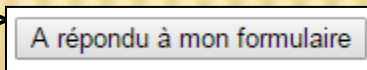
```
<SELECT NAME="genre" SIZE="1">  
<OPTION VALUE="Madame" selected>Madame</OPTION>  
<OPTION VALUE="Mademoiselle">Mademoiselle</OPTION>  
<OPTION VALUE="Monsieur">Monsieur</OPTION>  
</SELECT>
```



Des boutons radio : `<INPUT TYPE="radio" NAME="Bouton choisi" VALUE="Contact téléphonique">` ☐

Des cases à cocher : `<INPUT TYPE="checkbox" NAME="Case coché" VALUE="Création de site">` ☐

Un bouton d'envoi : `<INPUT TYPE="submit" NAME="Submit" VALUE="A répondu à mon formulaire">`



# Les Cadres ( Frames)

L'utilisation de cadres (les "frames") permet de diviser l'écran du navigateur en plusieurs zones. Chaque zone peut contenir une page html.

## 1.L'élément **FRAMESET**

Il permet de définir le nombre de zones et leurs dimensions. Les deux principaux attributs sont rows et cols.

**ROWS** = "h1,h2,...hn"

permet de faire des zones horizontales.

**COLS** = "l1,l2,...ln"

permet de faire zones verticales

Les hauteurs peuvent être données en pixels ou en proportions. Ces deux attributs peuvent être employés conjointement pour faire une grille.

<FRAMESET rows="25%,25%,50%"> ... </FRAMESET> en pourcentage

<FRAMESET cols="25,25,50"> ... </FRAMESET> en pixels

### 1.1 Les éléments **FRAMESET** peuvent être imbriqués :

Exemple:

```
<FRAMESET cols="100,3*,*">
```

```
...
```

```
<FRAMESET rows="30%,*">
```

```
...
```

```
</FRAMESET> </FRAMESET>
```

# Les Cadres ( Frames)

## 2.L'ÉLÉMENT *FRAME*

L'élément FRAME permet de définir le contenu de chacune des zones. Pour cela, on utilise deux attributs :

**SRC=***url* pour indiquer le fichier à placer dans la zone

**NAME=***NomDeZone* permet de nommer la zone afin qu'elle puisse devenir la cible d'un lien (on utilisera alors **l'attribut target** de la balise <A> pour y accéder.

Exemple :

```
<FRAMESET ROWS="100,3*,*">  
<FRAME SRC=FICHIER1.HTML NAME=DEBUT ...>  
<FRAME SRC=FICHIER2.HTML NAME=SOMMAIRE ...>  
<FRAME SRC=FICHIER3.HTML NAME=COURS ...>  
</FRAMESET>
```

## 3. TYPE DE DOCUMENT

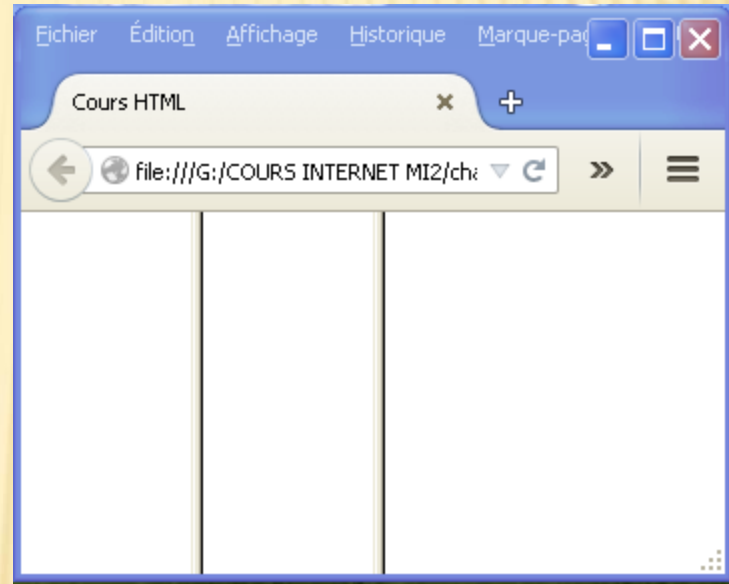
Un document contenant des frames doit tout d'abord être déclaré comme tel (élément **DOCTYPE** de type FRAMESET). Voici la forme d'un document contenant des frames.



# Les Cadres ( Frames)

Exemple d'application:

```
<!DOCTYPE HTML PUBLIC >
<HTML>
<HEAD>
<TITLE>COURS HTML </TITLE>
</HEAD>
<FRAMESET COLS="25,25,50">
<FRAME>
<FRAME>
<FRAME>
</FRAMESET>
</HTML>
```



## 4. Envoyer un lien vers un cadre

L'attribut `target` de l'élément `A` permet de diriger un document vers un cadre autre que celui qui contient le lien activé. Sinon, lorsqu'on active un lien, par défaut la nouvelle page s'affiche dans le cadre contenant le lien en question.

Exemple :

```
<A href="introduction.html" target=cours>Introduction</A>
```