

Le **protocole HTTP** (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. La version initiale était uniquement destinée à transférer des données. Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format (X)HTML) localisés grâce à **une chaîne de caractères** appelée **URL** entre un **navigateur** (le **client**) et un **serveur Web**.

- <http://www.news-de-stars.com>
- <file:///C:/wamp/www/Herge/index.html>

Le **HTML** (« **Hypertext Mark-Up Language**») est un **langage** dit de « **marquage** » (de "structuration" ou de "**balisage**") dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage.

Les **balises** permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

Quelques règles simples sont à respecter lors de l'écriture des balises html.

- **Règle 1** : Ecrire les balises en **minuscule**.
- **Règle 2** : Toutes les balises **ouvertes** par `<nom_balise>` doivent être **fermées** par `</nom_balise>` sauf s'il s'agit d'une balise unique `<balise_unique />`.
- **Règle 3** : Les balises doivent être correctement **imbriquées** et **indentées**

Exemple : Dans le code ci-dessous, `<h1></h1>` est **imbriquée** avec `<td></td>`, `<tr>` et `<td>` sont **indentées**.

Le secret de la licorne

- **Règle 4** : Les valeurs des attributs doivent toujours figurer entre des **guillemets**.

Exemple : `<table border = "1">`

I. Définition du HTML

HTML est l'abréviation de **HyperText Markup Language**, c'est à dire « langage de mise en forme de documents hypertextes ». « Hypertexte » signifie que les textes contiennent des références croisées (des liens), et « mise en forme » renvoie aux propriétés de formatage du texte (titre, liste, gras, cellule de tableau, etc.)

II. Propriétés du HTML

- les pages HTML sont des fichiers texte simples portant l'extension .htm ou .html ;
- chaque page HTML se compose d'une balise `<html></html>` qui contient l'ensemble de la page et qui n'est contenu dans aucune balise;
- la balise `<html>` contient deux balises :
 - `<head>`, l'en-tête permet de configurer les propriétés de la page.
 - `<body>`, le corps contient la page affichée.
- à chaque balise de début doit correspondre une balise de fin qui est différencié par un slash. (exemple : `<p>mon paragraphe</p>`)
- Attention : les balises ne doivent pas être entrelacées entre elles : `<p><i>...</i></p>` est correct, cependant `<p><i>....</p></i>` n'est pas correct ;
- quand les balises sont vides on peut compresser la balise de début et la balise de fin. exemple : `<hr />`
- il est possible d'insérer un commentaire entre "`<!--`" et "`-->`"

III. Les accents

Il est vivement recommandé de remplacer les caractères accentués et les caractères spéciaux par le code HTML correspondant.

Caractère	Codage	Caractère	Codage
à	à ;	î	î ;
â	â ;	ô	ô ;
é	é ;	ù	ù ;
è	è ;	û	û ;
ê	ê ;	"espace"	

HTML : Les Balises

IV. Quelques exemples de balises

Balise	Description	Exemple
<title>	Cette balise se place dans <head> et correspond au titre de la page. Il sera affiché dans l'onglet de votre fenêtre.	<title>ma page</title>
<h1>, <h2>...	Les balises <h1> à <h6> constituent des balises de titre, permettant jusqu'à six niveau de titres.	Grand titre Sous-titre
<p>	Les paragraphes sont introduits par et s'achèvent par . Il est possible de changer l'alignement du paragraphe avec l'attribut « align ».	<p align= "right">mon paragraphe</p>
 	Cette balise permet le renvoi à la ligne.	
<hr />	La balise <hr /> insère dans votre page une barre horizontale. Il est possible de changer la taille de la ligne avec l'attribut « width ».	<hr width="30%" align="left" />
<a>	La balise <a> permet la création de lien hypertexte. Cette balise permet entre autre de : Faire un lien vers une autre page du site Faire un lien vers un autre site Se déplacer rapidement dans la page. (ancrage) Permettre à l'utilisateur de télécharger un fichier Faire un lien vers une adresse mail	Suite Le site du zero soustitre3 ... <h2 id="bas">soustitre3</h2> fichier Contact
	La balise permet d'insérer une image.	
, et	Les listes numérotées sont introduites par et conclues par . Les listes à puces commencent par et se terminent par . Les différents éléments d'une liste sont placés entre des balises et .	 - Partie 1 - Partie 2 - Partie 3
<table>	Les tableaux sont introduits par <table> et conclus par </table>. Chaque ligne du tableau (table row) commence par <tr> et s'achève par </tr>. Une cellule (table data) est définie à l'aide des balises <td></td>	<table><tr> <td>Nom</td><td>Licences</td> </tr> <tr> <td>Word</td><td>10</td> </tr> <tr> <td>Excel</td><td>5</td> </tr> <tr> <td>Photoshop</td><td>2</td> </tr></table>

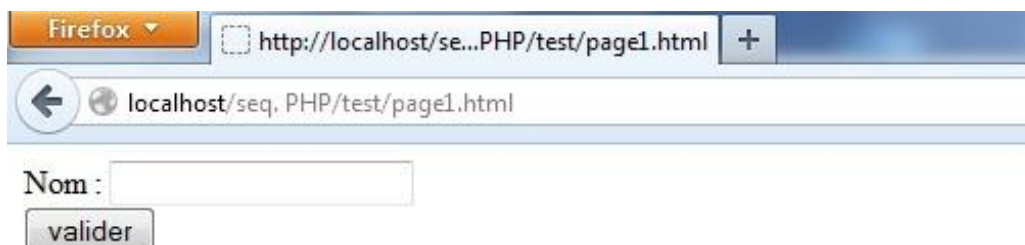
<input> La balise *INPUT* est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments "interactifs", de type :

- **text**: il s'agit d'un *champ de saisie* permettant la saisie d'une ligne de texte. La taille du champ peut être définie à l'aide de l'attribut *size* et la taille maximale du texte saisi grâce à l'attribut *maxlength*
- **checkbox**: il s'agit de *cases à cocher* pouvant admettre deux états : *checked* (coché) et *unchecked* (non coché).
- **hidden**: il s'agit d'un *champ caché*. Ce champ non visible sur le formulaire permet de préciser un paramètre fixe qui sera envoyé.
- **file**: il s'agit d'un *champ* permettant à l'utilisateur de préciser l'emplacement d'un fichier qui sera envoyé avec le formulaire. Il faut dans ce cas préciser le type de données pouvant être envoyées grâce à l'attribut *ACCEPT* de la balise *FORM*
- **image**: il s'agit d'un *bouton de soumission personnalisé*, dont l'apparence est l'image situé à l'emplacement précisé par son attribut *SRC*
- **password**: il s'agit d'un *champ de saisie*, dans lequel les caractères saisis apparaissent sous forme d'astérisques afin de camoufler la saisie de l'utilisateur
- **radio**: il s'agit d'un *bouton* permettant un choix parmi plusieurs proposés (l'ensemble des boutons radios devant porter le même attribut *name*. La paire nom/valeur du bouton radio sélectionné sera envoyé. Un attribut *checked* pour un des boutons permet de préciser le bouton sélectionné par défaut
- **submit**: il s'agit du *bouton de soumission* permettant l'envoi du formulaire. Le texte du bouton peut être précisé grâce à l'attribut *value*
- **reset**: il s'agit d'un *bouton de remise à zéro* permettant uniquement de rétablir l'ensemble des éléments du formulaire à leurs valeurs par défaut

Exemple:

```
<form action="page2.php" method="GET">
    <label for="nom">Nom :</label>
    <input type="text" name="nom" id="nom"></input><br />
    <input type="submit" name="valider"
value="valider"></input>
</form>
```

Affichage :



<textarea> La balise *TEXTAREA* permet de définir une zone de saisie plus vaste par rapport à la simple ligne de saisie que propose la balise *INPUT*.

Affichage :

Commentaires :

valider

<select> La balise *SELECT* permet de créer une liste déroulante d'éléments (précisés par des balises *OPTION* à l'intérieur de celle-ci).

Les attributs de cette balise sont :

- **name**: représente le nom associé au champ, c'est le nom qui permettra d'identifier le champ dans la paire nom/valeur
- **disabled**: permet de créer une liste désactivée, c'est-à-dire affichée en grisée
- **size**: représente le nombre de lignes dans la liste (cette valeur peut être plus grande que le nombre d'éléments effectifs dans la liste)
- **multiple**: marque la possibilité pour l'utilisateur de choisir plusieurs champs dans la liste

Exemple:

```
<form action="page2.php" method="GET">
  <select name="couleur">
    <option value="white">blanc</option>
    <option value="red">rouge</option>
    <option value="green">vert</option>
  </select>
  <input type="submit" name="valider" value="valider"></input>
</form>
```

Affichage :

blanc ▼

blanc
rouge
vert

valider

Récapitulatif :

Balise	Attribut	Valeur	Résultat	Effet Visuel
<FORM> ... </FORM>	METHOD	POST GET		
	ACTION		envoi à l'adresse indiquée	
	ENCTYPE		spécifie le type de codage utilisé	
<INPUT>	TYPE	submit	effectue l'ACTION dans le balise <FORM>	<input type="submit" value="Envoyer"/>
		text	simple ligne de texte dont la longueur est donnée par l'attribut size	
		reset	efface le contenu du formulaire	<input type="reset" value="Rétablir"/>
		radio	bouton radio	☒
		checkbox	case à cocher	☐
		...	...	
	NAME		Nom	
	SIZE		Taille du texte	
<TEXTAREA> ... </TEXTAREA>	NAME		Zone de texte	
	ROWS			
	COLS			
<SELECT> <OPTION> ... </OPTION> </SELECT>	NAME			Choix 1
	MULTIPLE		Plusieurs choix possibles	Choix 1
<OPTION> ... </OPTION>	SELECTED	Option par défaut		Choix 1 Choix 2 Choix 3
	VALUE	Valeur forcée		

I. Définition

CSS est l'abréviation de « Cascading Style Sheets », c'est à dire, en français, les feuilles de style en cascade. Elles permettent de modifier le style et les propriétés de chaque balise offrant ainsi à l'utilisateur une infinité de possibilité pour décorer sa page.

Il y a deux manières d'utiliser le CSS :

- La méthode la plus simple est de l'intégrer directement dans la page html avec la balise `<style>`.

Exemple :

```
<style type= "text/css">
<!--
placez ici les feuilles de style
// -- !>
< /style>
```

- La deuxième méthode consiste à séparer le CSS de la page HTML. Cela permet de **pouvoir utiliser la même feuille de style pour plusieurs pages HTML**. Pour cela il faut importer le CSS avec la balise `<link>` dans `<head>`.

Exemple :

```
<head>
<link rel="stylesheet" type="text/css" href="office.css" />
</head>
```

II. Propriétés du CSS

- Le CSS n'est pas un langage de balise. Il se présente de cette manière :

```
nom_de_la_balise
{
    // pour attribuer le style à toutes les balises de ce nom
    attribut_1 : valeur ;
    attribut_2 : valeur1, valeur2, valeur3 ; // commentaire
}

.ma_class
{
    // pour attribuer le style à toutes les balises de cette class
    attribut : valeur ;
}

$mon_identifiant
{
    //pour attribuer le style à la balise ayant cet identifiant
    attribut : valeur ;
}
```

- Pour attribuer un style à plusieurs balises, classe ou identifiant en même temps, il faut les séparer par une virgule.

Exemple :

```
#bas , .maiclass , #haut {  
    text-align : center;  
} //s'applique aux identifiants bas et haut et la classe maiclass
```

- Pour faciliter la « mise en style » d'une page HTML on la divise en différent bloc grâce à la balise <div>.

Exemple :

```
<html>  
<head>...</head>  
<body>  
    <div id="entete" />  
    <div id="container">  
        <div id="menu">  
            <h3>Menu</h3>  
            <ol>  
                <li><a href="page1.html">page 1</a></li>  
                <li><a href="page2.html">page 2</a></li>  
                <li><a href="page3.html">page 3</a></li>  
            </ol>  
        </div>  
        <div id="corps">  
            <h1>Grand titre</h1>  
            <h2>Sous-titre</h2>  
            <p>  
                ...  
            </p>  
        </div>  
    </div>  
</body>  
</html>
```

- Il est possible de modifier le style d'une balise en fonction des balises dans lequel elle est imbriquée. Pour cela, il faut les écrire dans l'ordre hiérarchique en les séparant par un espace.

Exemple :

```
#container #corps h1 {  
    ...  
} //modifie uniquement "h1" contenu dans "corps" contenu dans  
"container"
```

- On peut également modifier le style d'une balise lors du survol de la souris en rajoutant ":hover" au nom de la balise.

Exemple :

```
h1:hover {  
  ...  
}
```

nota : dans la même optique, il existe pour la balise <a> :

a:link (état normal du lien), **a:hover**, **a:active** (état actif du lien), **a:visited** (état visité)

- Il est possible d'appliquer un style à un mot ou une partie d'un paragraphe avec la balise

Propriété de Police

Attribut	Valeur	Élément contrôlé	Exemple
font-family	nom de police concret	police de caractères	font-family :Arial;
	serif	famille de polices avec serif	font-family : Times, serif;
	sans-serif	famille de polices sans serif	font-family :Arial, sans-serif;
	cursive	famille de polices d'apparence italique	font-family :cursive;
	monospace	famille de polices non proportionnelles	font-family :monospace;
	fantasy	famille de polices fantaisie	font-family :Western, fantasy;
font-size	pt, cm, etc.	taille de police	font-size : 12pt ;
font-style		style de police	
	normal	sans propriétés	
	italic	italique	font-style : italic ;
	oblique	polices à corps creux	font-style : oblique;
font-variant		variantes de polices	
	normal	sans propriétés	
	small-caps	petites majuscules	font-variant : small-caps ;
font-weight	de 100 à 900	épaisseur (mesurée par pas de cent)	font-weight : 500 ;
	normal	correspond à 400	font-weight : normal ;
	bold	correspond à 700	font-weight : bold ;

Mise en forme du texte

Attribut	Valeur	Élément contrôlé	Exemple
word-spacing	pt, cm	espacement des mots	word-spacing : 5pt ;
letter-spacing		espacement des caractères	letter-spacing : 0,2cm ;
text-decoration		mise en avant	
	none	désactive la mise en valeur	text-decoration : none ;
	underline	souligné	text-decoration : underline;
	overline	trait au-dessus	text-decoration : overline ;
	line-through	barré	text-decoration : line-through ;
	blink	texte clignotant	text-decoration : blink ;
text-transform		majuscules et minuscules	
	none	aucun changement	
	uppercase	majuscules	text-transform : uppercase ;
	lowercase	minuscules	text-transform : lowercase ;
	capitalize	première lettre du mot en majuscule	text-transform : capitalize ;

Agencement du texte

Attribut	Valeur	Élément contrôlé	Exemple
display	block	affichage sous forme de bloc	<code>display : block ;</code>
	inline	affichage dans le texte	<code>display : inline ;</code>
text-align		alignement de paragraphes	
	left	aligné à gauche	<code>text-align : left ;</code>
	center	centré	<code>text-align : center ;</code>
	right	aligné à droite	<code>text-align : right ;</code>
	justify	justifié	<code>text-align : justify ;</code>
text-indent	pt, cm, etc.	retrait de première ligne	<code>text-indent : 1cm ;</code>
line-height	pt, cm, etc.	hauteur de ligne	<code>line-height : 14cm ;</code>
	valeur relative	interligne en fonction de la taille des caractères	<code>line-height : 1.2 ;</code>
vertical-align		alignement vertical par rapport à la ligne de base sur laquelle est aligné le texte	
	baseline	sur la ligne	<code>vertical-align : baseline ;</code>
	super	en exposant	<code>vertical-align : super ;</code>
	top	le long de la limite supérieure de la ligne	<code>vertical-align : top ;</code>
	text-top	le long de la limite supérieure du texte	<code>vertical-align : text-top ;</code>
	middle	centré	<code>vertical-align : middle ;</code>
	text-bottom	limite inférieure de la ligne	<code>vertical-align : text-bottom ;</code>
	sub	mise en indice	<code>vertical-align : sub ;</code>

Couleurs et arrière-plans

Attribut	Valeur	Élément contrôlé	Exemple
color		couleur du contenu d'un élément, en général du texte	<code>body{ color :blue ; }</code>
background		couleur d'arrière-plan, par exemple pour créer un effet d'ombrage	<code>h1, h2 {background : silver ;}</code>
background-image		incorpore une image d'arrière-plan	<code>body { background-image : url(image.gif) ;}</code>
background-repeat		indique comment l'image doit être alignée	
	repeat	paramètre par défaut, mosaïque	<code>background-repeat : repeat ;</code>
	repeat-x	répétition horizontale, sur l'axe des «x»	<code>background-repeat : repeat-x ;</code>
	repeat-y	répétition verticale, sur l'axe des «y»	<code>background-repeat : repeat-y ;</code>
	no-repeat	pas de répétition	<code>background-repeat : no-repeat ;</code>

CSS : Les attributs

background-attachement		indique si l'image d'arrière-plan se déplace avec le contenu de la page lorsque ce dernier défile	
	scroll	paramètre par défaut, l'image d'arrière-plan se déplace avec la page	background-attachement : scroll ;
	fixed	l'image d'arrière-plan est fixe	background-attachement : fixed ;
background-position	px, cm, etc.	position de l'image d'arrière-plan, d'abord à gauche puis en haut	background-position : 100px 10px ;

Retraits extérieur et intérieur

Attribut	Valeur	Élément contrôlé	Exemple
margin	pt, px, cm, etc.	retrait extérieur, marge	body{ margin :1cm ;}
margin-top		marge supérieure	p {margin-top : 10px ;}
margin-bottom		marge inférieure	h3 {margin-bottom: 3pt;}
margin-left		marge de gauche	img { margin-left : 50px ;}
margin-right		marge de droite	p.citation {margin-right : 10pt ;}
padding	pt, px, cm, etc.	marge intérieure	p { padding : 2cm ;}
padding-top		marge intérieure du haut	body {padding-top :15px ;}
padding-bottom		marge intérieure du bas	h3 {padding-bottom : 1.3cm ;}
padding-left		marge intérieure de gauche	p.nouveau {padding-left :5px}
padding-right		marge intérieure de droite	img {padding-right : 5cm ;}

Cadres

Attribut	Valeur	Élément contrôlé	Exemple
border	px, cm, etc.	épaisseur du cadre	p {border : 5px ;}
border-top		épaisseur du cadre haut	h1 {border-top : 0.2cm ; }
border-bottom		épaisseur du cadre bas	h2 {border-bottom : 4px ; }
border-left		épaisseur du cadre à gauche	p.citation {border-left : 5px ;}
border-right		épaisseur du cadre à droite	pre {border-right : 12px ;}
border-style		style de cadre	
	solid	continu	p{border-style : solid ;}
	double	ligne double	h1{border-style : double ;}
	groove	effet 3D avec rainure	p{border-style : groove ;}
	ridge	effet 3D avec rebord agrandi	p.nouveau{border-style : ridge ;}
	inset	effet 3D avec contenu	div{ border-style : inset ;}

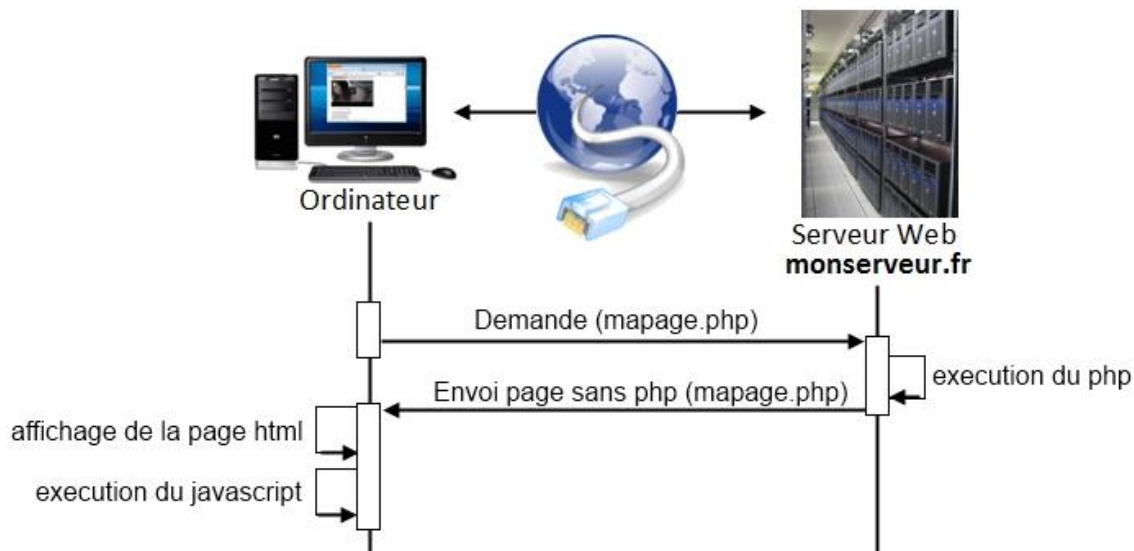
CSS : Les attributs

		renfoncé	
	outset	effet 3D avec contenu rehaussé	<code>span {border-style : outset ;}</code>
	dotted	contour pointillé	<code>h2, h3{border-style : dotted ;}</code>
	dashed	contour à tirets	<code>pre{border-style : dashed ;}</code>
border-style-top	voir ci-dessus	style de cadre pour le haut	<code>border-style-top : ridge ;</code>
border-style-bottom	voir ci-dessus	style de cadre pour le bas	
border-style-left	voir ci-dessus	style de cadre pour la gauche	
border-style-right	voir ci-dessus	style de cadre pour la droite	
border-width	px, cm, etc.	espace entre les lignes du cadre, là aussi border-width-top, border-width-bottom, etc. sont possibles	<code><div style="border style : outset; border-width: 10px;"></code>
border-color	valeur hex., nom, rgb	couleur de cadre, là aussi border-color-top, etc. sont possibles	<code>p{border : 7px ; border-color : rgb(255, 0, 0) ;}</code>

Positionnement absolu des éléments

Attribut	Valeur	Élément contrôlé	Exemple
position	positionnement sur la page		
	static	paramètre par défaut	
	absolute	positionnement absolu par rapport à l'angle supérieur gauche, la position est conservée lorsque la page défile	<code>position : absolute ; top : 10px ; left : 20px ;</code>
	relative	positionnement relatif à l'élément précédent	<code>position : relative ; top : 20px ;</code>
top	px, pt, etc.	retrait du bord supérieur	
left	px, pt, etc.	retrait du bord gauche	
width	px, pt, etc.	largeur de l'élément	<code>width : 150 px ;</code>
height	px, pt, etc.	hauteur de l'élément	<code>height : 80px ;</code>
visibility		visibilité	
	visible	visible	
	hidden	masqué	<code>visibility : hidden ;</code>
z-index	1, 2, 3, etc.	définit l'ordre des éléments	<code>z-index : 2 ;</code>

Fonctionnement global



Pour pouvoir communiquer entre eux, l'ordinateur client et le serveur ont besoin d'un logiciel appelé navigateur pour le client et serveur web pour le serveur.

Le navigateur seul ne pouvant pas interpréter le langage PHP nous utiliserons le logiciel WampServer.

Ce logiciel est composé entre autre de :

- Php my admin, permettant la création et la gestion de bases de données ;
- Un dossier « www » permettant de stocker notre site web. Il sera lu par le navigateur à l'adresse « localhost ».

Le PHP à quoi ça sert ?

Le PHP permet par exemple:

- De créer un site web qui interagit avec une base de donnée. (en lecture et écriture)
- De permettre à l'utilisateur d'interagir avec le site web. (accéder à une page spécifique, choix du design etc.)
- De permettre au site web de lire ou d'écrire dans un fichier (Texte, XML, CSV etc.)

En combinant ces 3 utilisations, on peut faire une infinité de choses (recherche par mot clé dans une Foire aux Questions, choix de la langue du site, jeu vidéo etc.)

Comment s'en servir ?

Pour que le serveur interprète le PHP il faut lui préciser que le fichier est un fichier contenant ce langage de programmation. Celui-ci est donc nommé sous la forme « monfichier.php ».

Concernant le code PHP, il est inséré dans le code HTML entre les caractères « <?php » et « ?> ».

On peut également afficher du HTML sans quitter le PHP en utilisant « echo »

Exemple : `echo "<h1>Bonjour Monsieur</h1>";`

Comme beaucoup de langage de programmation, une ligne de code PHP fini toujours par le caractère « ; ».

Pour stocker une valeur numérique (le temps du programme) pour pouvoir faire des calculs, l'incrémenter etc. on utilise des variables. On peut également y stocker du texte.

Les variables

Dans le langage PHP, on utilise une variable sans avoir déclaré son type auparavant.

Exemple : `$maVariable1 = 10 ;`

Ou : `$maVariable2 = "Bonjour";`

On peut de la même façon les afficher grâce à « écho ».

Exemple : `$monNom = "PARADE" ;`

`echo "Bonjour Monsieur $monNom" ;`

Comment interagir entre le programme côté serveur et le client ?

Pour que le client puisse communiquer avec le serveur, on utilise généralement un formulaire de saisie.

Exemple ->

login	
mdp	
Se connecter	

Les formulaires

Les formulaires sont créés en HTML avec la balise <form> (voir index).

Il existe deux méthodes pour envoyer les données d'un formulaire :

- La méthode GET affiche les variables en clair dans la barre d'adresse.
Exemple : <http://www.monsite.fr/?page=1>
Faire un lien <a> en renseignant les variables dans l'adresse permet de ne pas passer par le formulaire.
- La méthode POST permet d'envoyer des variables de façon plus discrète.

Réception d'un formulaire

Méthode GET :

`$_GET['nom']` est utilisé comme une variable pour utilisé la valeur de 'nom' envoyé dans le formulaire.

Méthode POST :

La méthode POST est similaire à la méthode GET : `$_POST['nom']`.

Exemple :

```
if(isset($_GET['envoyerEntreprise']))
{
    //si le bouton "envoyerEntreprise" a été validé

    echo "<font color=\"red\">".$_GET['entreprise']</font>";
    //afficher le nom de l'entreprise en rouge
}
```

Interagir avec une Base De Données

Se connecter :

```
$pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
$dbdd = new PDO('mysql:host=localhost;dbname=simu_ventilo',
                'root', '', $pdo_options);
```

Il existe deux méthodes en PHP pour envoyer une requête SQL :

- La fonction `exec()`
- La fonction `prepare()` puis `execute()`

Lire dans une BDD :

```
$reponse = $bdd->prepare("
    SELECT DAY(mp_date) AS jour FROM marche_partielle
    WHERE MONTH(mp_date) = ? AND YEAR(mp_date) = ? AND mp_idPiece = ?
");
$reponse->execute(array($mois, $annee, $_GET['idPiece']));
```

Ecrire dans une BDD :

```
$bdd->exec("
    INSERT INTO marche_complete(mc_date, mc_idPiece)
    VALUES ('".$annee."-".$mois."-".$i."', '".$_GET['idPiece']."' )
");
```

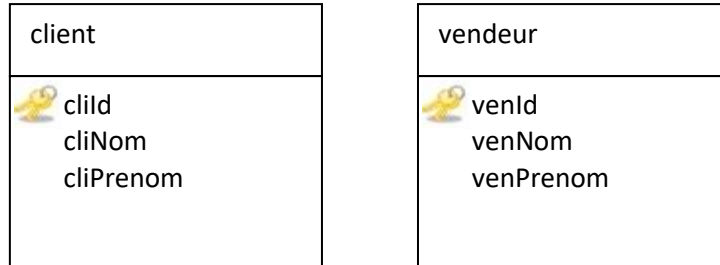
Effacer des données :

```
$bdd->exec("
    DELETE FROM marche_complete
    WHERE mc_idPiece='".$_GET['idPiece']."'
");
```


Base de donnée

Une base de données est composée de tables elles-mêmes composées de données réparties dans différentes colonnes.

Les tables



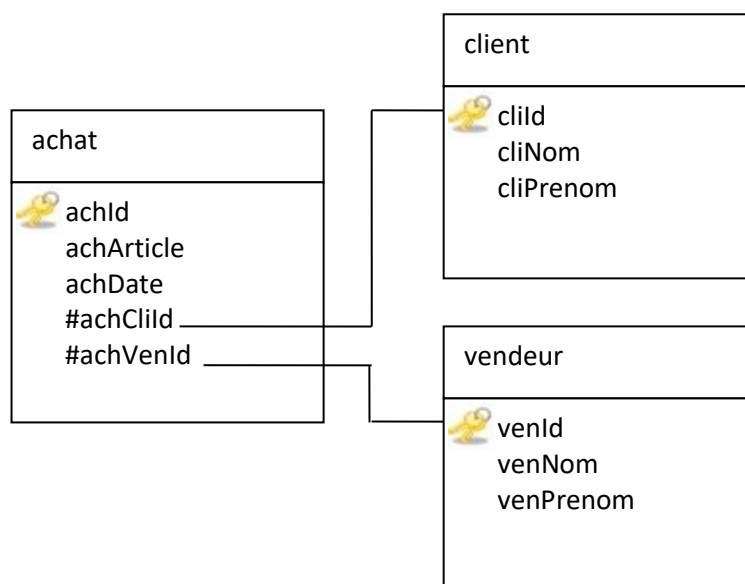
- Pour éviter la confusion lorsqu'il y a plusieurs tables, le nom de chaque colonne commence par les 3 premières lettres du nom de la table, le caractère suivant étant une majuscule.

Exemple : cliNom est une colonne de la table client.

- Afin de mieux gérer les doublons d'une même table, la première colonne de celle-ci est **systématiquement un identifiant** (id). On l'appelle la **clé primaire**. Elle est généralement un nombre qui s'incrémente automatiquement à chaque nouvelle donnée.

- Lorsqu'on veut faire correspondre des données d'une table (ex : un achat) avec les données d'une ou plusieurs autres tables (ex : un client ou client et vendeur), on renseigne les identifiants des tables dans la table concernée en rajoutant un # à leur nom. On parle alors de **clé secondaire**.

Exemple : #achVenId est une des clés secondaires de la table achat. Pour chaque achat, elle correspondra à l'identifiant d'un des vendeurs.



Le langage SQL

Pour pouvoir lire ou écrire on utilise le langage SQL.

Lire dans une BDD :

Syntaxe :

```
SELECT ma_colonne_01, ma_colonne_02, ma_colonne_03  
      FROM ma_table_01, ma_table_02  
      WHERE <condition>
```

Exemple :

```
SELECT DAY(mp_date) AS jour  
      FROM marche_partielle  
      WHERE MONTH(mp_date) = 2  
            AND YEAR(mp_date) = 2012  
            AND mp_idPiece = 512
```

Description :

```
Afficher les jours de la colonne mp_date dans une "variable" nommée "jour"  
      Venant de la table marche_partielle  
      Quand le mois de la colonne mp_date = 2  
            ET quand l'année de la colonne mp_date = 2012  
            ET quand la clé secondaire représentant l'identifiant de la pièce = 512
```

Ecrire dans une BDD :

```
INSERT INTO marche_complete(mc_date, mc_idPiece)  
      VALUES('2012-01-30', '200')
```

Suppression de données :

```
DELETE FROM marche_complete WHERE mc_idPiece=21
```

Introduction à WampServer

Lancement

Quand vous démarrez WampServer une icône apparaît dans votre barre des tâches.

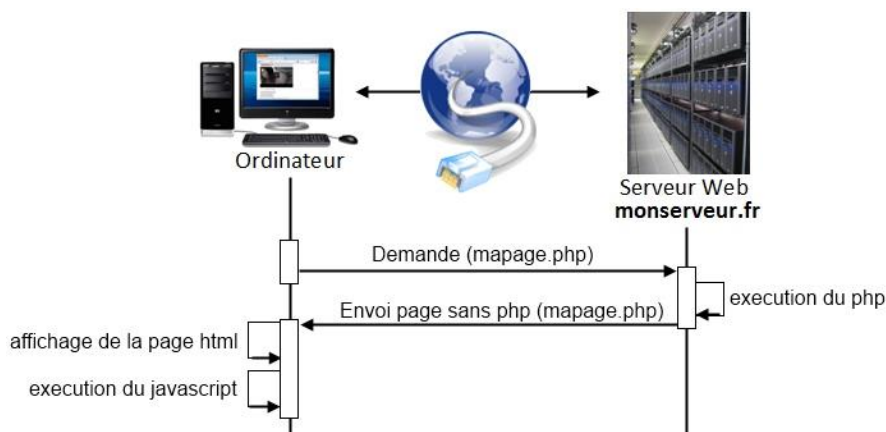
 : Tous les services sont activés

 : Aucun service activé

En cliquant sur cet icône vous pourrez accéder à différents services.



Fonctionnement



Un navigateur seul ne peut pas lire le PHP, seul le serveur le peut. Il faudra donc déposer le site web dans un dossier spécial du serveur, le **répertoire www** ("www directory") accessible via Wamp.

Une fois le site web dans ce dossier, vous pourrez alors visualiser votre œuvre en vous connectant au "localhost". Celui-ci est accessible via le navigateur **en tapant** "localhost" dans la barre de navigation, mais également via WampServer **en cliquant** sur "localhost".

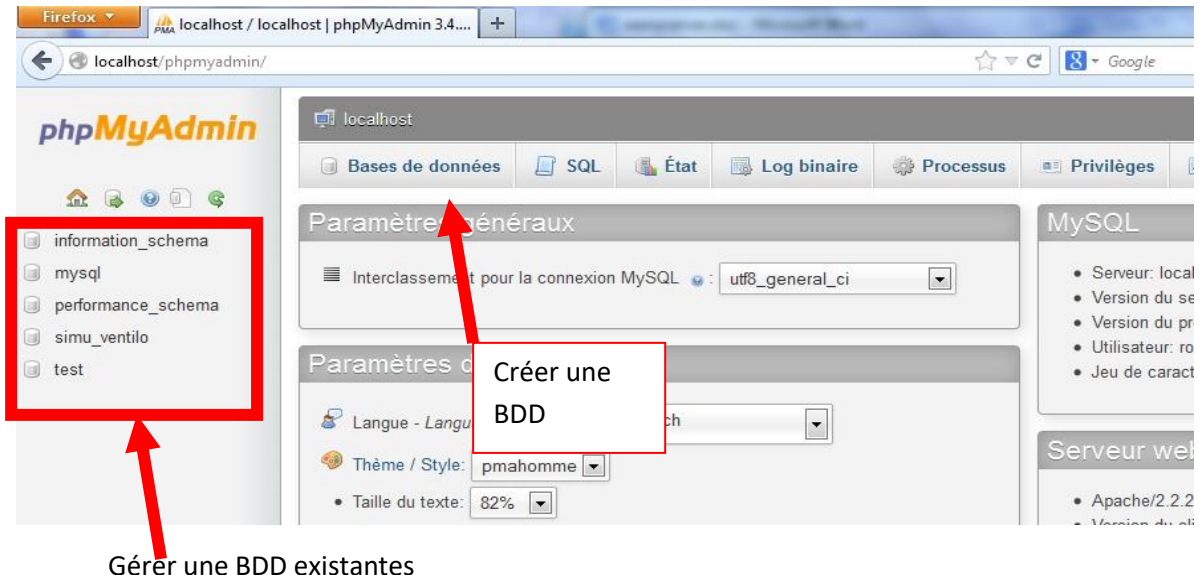
Introduction à WampServer

phpMyAdmin



Ce service permet de créer et gérer une Base De Données (BDD). Une BDD va, par exemple, permettre à un magasin d'avoir la liste de ses clients en liens avec leurs achats.

phpMyAdmin n'est autre qu'un site en PHP hébergé par WampServer permettant d'interagir avec vos BDD.

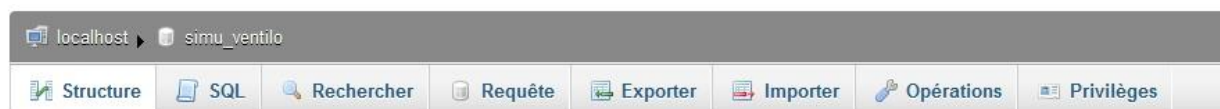


Pour créer une BDD, cliquez sur l'onglet "Bases de données". Cliquez sur "créer" après avoir renseigné le nom et l'interclassement de votre base.

Nota : un caractère est codé sur 7 à 8 bits. Les 7 premiers bits sont codés à partir de la table ascii. Cependant l'interprétation du 8iem bit différent en fonction de l'interclassement.

Pour gérer une base de données existante sélectionnez-la dans le menu de gauche.

Une fois la BDD sélectionnée :



Structure : Permet de créer et gérer les tables de la BDD. (Voir le tutoriel sur les BDD)

SQL : Permet de tester des requêtes SQL.

Requête : Permet "d'auto-concevoir" une requête.

Exporter : Permet d'exporter les tables et les données dans un fichier. (Sous forme de ligne de code SQL)

Importer : Permet d'importer des tables et données d'une base précédemment exportée.