

Développement PHP & MySQL

CLIENT	{	HTML	Code d'affichage (sémantique, contenu, ...)
		CSS	Code de styles graphiques (aspect, couleur, disposition, ...)
		JS	Code d'interactions (validation formulaire, slider, ...)
SERVEUR	{	PHP	Code de traitement (calculs, traitement de données, ...)
		MySQL	Base de données (stockage d'informations)

PHP et MySQL

application serveur :

- Serveur web Apache (technologie linux),
- moteur PHP (fichiers de compilation),
- Serveur de base de données MySQL

Heureusement, des solutions «all-in-one» existent :

WAMP, MAMP, EasyPhp, Xampp, ...

Comment travailler avec PHP et MySQL ?

- Serveur All-In-One
 - Editeur texte : notepad++, textwrangler, ...
 - Navigateur avancé : Google Chrome, Firefox
 - Ressources : Google
-
- Bonne connaissance HTML/CSS indispensable !

Premiers pas...

- Travailler dans l'environnement serveur (repertoire www)
- Travailler avec des fichiers .php (et non .html)
- Déclarer son code php dans une page html en utilisant :

```
<?php  
    [CODE PHP A INSERER]  
?>
```

Les bases du langage PHP

```
<?php
    // ceci est un commentaire
    /*
        Commentaire sur plusieurs
        lignes
    */
    instruction;
    fonction(parametre);
?>
```

\$variable

minuscule, plusieursMots, et simple (pour s'y retrouver)

```
$chiffre1 = 12;  
$chiffre2 = 8;  
$resultat = $chiffre1 + $chiffre2;  
  
echo 'Le résultat est '.$resultat;
```

Les signes de calculs simples : + - / * % (modulo)

EXERCICE

Définir une variable **\$nombre** (qui représente un nombre de secondes) et en définir le nombre d'heures, de minutes et de secondes pour afficher ceci :

73 secondes font : 0 heure(s), 1 minute(s) et 13 seconde(s)

Pour aller plus loin :

- Définir également le nombre de jours
- mettre en gras «73 secondes»
- Définir un chiffre aléatoirement au chargement de la page

LES CONDITIONS

```
if (condition) {  
    instruction  
} else {  
    instruction  
}
```

```
if ($note >= 10) {  
    echo 'Accepté';  
} else {  
    echo 'Redoublement';  
}
```

Opérateurs de conditions : == != > >= < <=

Modifier l'exercice pour que si il y a des heures on les affiche, si il y a des jours on les affiche, etc...

LES STRUCTURES ITERATIVES (CONDITITONS, BOUCLES, CHOIX)

```
while ($compteur < 100) {BOUCLE}
```

```
for($i=1;$i<=100;$i++) {BOUCLE}
```

```
foreach($tableau as $valeur) {echo $valeur;}
```

```
switch($valeur) {case:1;instruction;break;}
```

EXERCICE : la roulette

un chiffre aléatoire entre 1 et 36 est calculé au chargement de la page,
On affiche le chiffre tiré au sort et on affiche les chances simples :
pair ou impair, passe (de 1 à 18) ou manque (de 19 à 36)

17 : impair et passe

Pour aller plus loin :

- de 0 à 36 : Traiter le cas particulier du 0 : écrire en vert et ne pas afficher pair/impair et passe/manque
- Travailler avec des tableaux pour les rouges et les noirs : modification de la couleur du texte affiché
- faire un lien/bouton pour relancer la page

Travail avec des formulaires / passage de données d'une page à une autre

Par un lien

```
<a href='page.php?chiffre=12'>Lien</a>  
$_REQUEST[ 'chiffre' ];
```

Par un formulaire

```
<form action='traitement.php' method='post'></form>  
$_REQUEST[ 'chiffre' ];
```

EXERCICE : la roue de la fortune

un chiffre aléatoire entre 1 et 10. Un champ où l'on défini un chiffre,
Et on affiche 3 possibilités : «C'est plus», «C'est moins», ou «Bravo»;

Pour aller plus loin :

- On traite au rechargement en récupérant les variables et en réaffichant le chiffre donné dans le champ
- On passe la variable du chiffre aléatoire en paramètre pour qu'il ne change pas à chaque fois
- On fait un lien pour relancer complètement le jeu (nouveau chiffre)

Include : permet de charger une page dans une page
`include 'page.php';`

Réalisez un site avec une navigation sous cette forme :

index.php	chargera accueil.php
index.php?page=equipe	chargera equipe.php
index.php?page=contact	chargera contact.php

Pour le formulaire de contact, gérez les champs «nom», «prenom», «mail», «message» et affichez les informations sur une nouvelle page «merci.php» !

Une SESSION est une variable masquée gérée sur/par le serveur permettant de stocker du contenu d'une page à une autre sans formulaire.

Par exemple : panier sur un site marchand, identification de Facebook, ...

`session_start();`

Initialise la session / doit être lancé en php pour utiliser la session

`session_destroy();`

Détruit toutes les données d'une session

Pour définir une variable de session :

`$_SESSION['nom'] = "thomas";`

Exercice : réaliser un formulaire de connection sur l'exercice avec les «include» pour faire une page de connection en testant si le login = «admin» et si le mot de passe = «123456» :

Dans le menu un lien vers la page de connection, et si connecté un lien vers la page `logout.php`

Fonctions utiles pour les chaines de caractères :

`$mon_texte = "Bonjour";`

`$mon_texte .= " tout le monde";`

`echo $mon_texte;`

`strlen($mon_texte);`

`substr($mon_texte, 1, 3);`

`strtolower($mon_texte);`

`strtoupper($mon_texte);`

`strlen($mon_texte);`

Affiche la variable \$mon_texte

Retourne la longueur de la chaîne \$mon_texte

Retourne la chaîne \$mon_texte de la pos. 1 et sur 3 caractères

Retourne la chaîne \$mon_texte en minuscule

Retourne la chaîne \$mon_texte en majuscule

Retourne la longueur de la chaîne \$mon_texte

Pour connaître la date du jour, il faut utiliser la fonction date :

`echo date("Ymd"); // retournera «20171225» (pour le jour de noël)`

- Y** Année sur 4 chiffres (ex : 2017)
 - y** Année sur 2 chiffres (ex : 17)
 - z** Jour de l'année (ex : 241 pour le 241ème jour de l'année)
 - m** Mois sur 2 chiffres (ex : 01 pour janvier)
 - M** Mois en 3 lettres en anglais (ex : Apr pour April)
 - n** Mois sur 1 ou 2 chiffres (ex : 2 pour février)
 - d** Jour du mois sur 2 chiffres (ex : 09)
 - D** Jour de la semaine sur 3 lettres (ex : Fri pour Friday)
- etc... Plus d'infos sur [php.net > dates et heures](http://php.net)

Exercice :

**1. Modifier le mini site (réalisé avec des includes et les sessions) pour stocker en session la date et l'heure de connection sous la forme :
2017/12/31 12:56:22 (année/mois/jour heure:minutes:secondes)**

**2. Afficher dans le footer du site la date de connection (si on est connecté) sous la forme :
connecté depuis le 31 décembre 2017 à 12h56**

**3. Et rajouter ceci en dessous (il va falloir regarder sur Internet pour le calcul) :
Vous êtes connecté depuis XXX secondes**

Exercice e-commerce

A l'aide des sessions, réalisez une page qui affichera 3 articles (html) avec un bouton «ajouter un panier» qui permettra d'ajouter au panier le/les article(s)

- 1. Gérez l'ajout au panier (session)**
- 2. Gérez l'affichage du panier et la suppression des articles dans le panier**
- 3. Gérez la quantité des produits dans le panier**

Découverte de PhpMyAdmin :

- Les utilisateurs et droits associés
- Les tables (Création, modification, suppression)
- Les importations / exportations / sauvegardes
- Les recherches à travers PhpMyAdmin

La structure d'une BDD

Un identifiant unique pour chaque ligne !

- unique (clef primaire)
- auto-increment
- INT(10) -> 9 999 999 999 lignes possibles

Utilisation des bons formats : INT, VARCHAR, TEXT, DATE, ...

Des noms explicites et simples (minuscules, underscore, ...)

EXERCICE : le cas d'école ^"

- 1/ Créer une table pour des **élèves**. Nous avons besoin de stocker le nom, le prénom, la date de naissance, le numéro de téléphone des parents et l'adresse mail des parents [5 lignes à créer]

- 2/ Créer une table pour les **matières enseignées**, avec le nom de la matière et le nom du professeur qui l'enseigne [3 lignes à créer]

- 3/ Créer une table pour les **contrôles**. Il nous faut associer une note (/20) à un élève et à une matière

TABLE eleves

id_eleve

nom

prenom

date_naissance

tel_maison

mail_pere

mail_mere

TABLE matieres

id_matiere

libelle

professeur

TABLE controles

id_controle

eleve

matiere

note

LES REQUETES

Toutes les matières

```
SELECT * FROM matieres;
```

Toutes les matières où le professeur est «Marc»

```
SELECT * FROM matieres WHERE professeur = 'Véronique TORBI';
```

Toutes les matières rangées par ordre alphabétique décroissant

```
SELECT * FROM matieres ORDER BY libelle DESC
```

LES REQUETES SUR PLUSIEURS TABLES

Afficher les notes de l'élève qui est né le 31 décembre 2001

```
SELECT prenom, note FROM controles, eleves WHERE controles.eleve =  
eleves.id_eleve AND date_naissance = '2005-08-11';
```

Tous les élèves qui ont une note inférieure à 10/20

```
SELECT nom, prenom FROM eleves, controles WHERE controles.eleve =  
eleves.id_eleve AND note <= 10 GROUP BY nom, prenom
```

EXERCICE

En une seule requête :

- 1/ Afficher l'élève le plus vieux uniquement
- 2/ Compter le nombre de notes / matière
- 3/ Afficher la matière où Remy est le meilleur

EXERCICE

1/ Afficher l'élève le plus vieux uniquement

```
SELECT nom, prenom FROM eleves ORDER BY date_naissance LIMIT 1
```

2/ Compter le nombre de notes / matière

```
SELECT matieres.libelle, COUNT(note) AS nombre FROM matieres, controles  
WHERE controles.matiere = matieres.id_matiere GROUP BY id_matiere
```

3/ Afficher la matière où Remy est le meilleur

```
SELECT libelle FROM matieres, controles, eleves WHERE id_eleve = eleve  
AND id_matiere = matiere AND eleves.prenom = 'Remy' ORDER BY note LIMIT 1
```

PHP : Se connecter à la base de données

```
$serveur = 'localhost';
$login = 'root';
$password = '123456';
$base = 'demo_mysql';

$connection = mysql_connect($serveur, $login, $password);
mysql_select_db($base, $connection);
```

PHP : Faire une requête

```
$requete = 'SELECT * FROM eleves ORDER BY nom, prenom';  
  
$resultat = mysql_query($requete);  
  
while ($liste = mysql_fetch_array($resultat) {  
  
    // Instructions pour chaque ligne du tableau de résultat  
  
}
```

EXERCICE

Créer une page html complète (html, head, body) qui affichera un tableau récapitulant les infos de chaque élève comme ceci (avec style!!!!):

Nom de l'élève
Dupond Martin

Date de naissance
2 décembre 1980

Téléphone fixe
+33 (0)2.12.34.56.78

Pour aller plus loin

- 1/ Proposer 2 boutons «TRIER PAR» : nom, date de naissance
- 2/ Afficher 2 colonnes en plus : pire note, meilleure note
- 3/ Afficher une colonne pour la moyenne obtenue (toutes les matières)

EXERCICE FINAL : le TO-DO-LIST

- 1/ Définir la structure de base de données
- 2/ Faire le php (une seule page) pour :
 - ajouter une tache (ligne de texte court - 140)
 - la traiter (à faire / fait) (dans les 2 sens)
 - modifier le texte
 - supprimer la tache
- 3/ Mettre du style CSS pour ça soit beau !



EXERCICE FINAL : PARTIE 2

Modifier le code, la base, les styles pour :

- 1/ Mettre un compteur de tâches à faire (dans h1)
- 2/ Rajouter une description à la tâche et l'afficher à l'aide d'un petit + (jquery peut-être ?)
(n'oubliez pas de modifier le code pour l'ajout et la modification d'une tâche)
- 3/ Rajouter une date à la tâche et afficher la tâche différemment si la tâche n'est pas traitée si la date est dépassée !

