# CIR1

# Web dynamique – travaux pratiques 1

Nathalie HELAL

## 1 Préambule

- Créez le répertoire TP1 (sur votre ordinateur portable) dans lequel vous mettrez l'ensemble de vos fichiers php, et autre documents (si besoin) pour ce TP.
- A la fin de ce TP, il faut déposer (sur campus.isen) l'ensemble de vos fichiers **sous forme d'un seul dossier compressé .zip.** Attention, nommez chaque exercice comme indiqué (ou bien le numéro de l'exercice si ce n'est pas précisé, exemple, exercice1.php).
- Installez un serveur web sur votre propre PC qui prend en charge le PHP et MySql (cela sera fait la séance prochaine, dans ce TP vous allez vous concentrer sur le PHP).
    - ✓ EasyPHP sur https://www.easyphp.org/easyphp-devserver.php
    - ✓ Open Dashboard
    - ✓ Start *HTTP Server (patientez pour qu'il démarre)* puis start *Database Server* (si problème redémarrez votre machine).
- Mettez vos fichiers dans le répertoire C:\EasyPHP-Devserver-17\eds-www
- Vos fichiers (exemple HelloWorld.php) seront alors accessible via l'url : http://127.0.0.1/HelloWorld.php

## 2 Objectif de ce TP

Voir et maîtriser les notions de base du PHP afin de bien comprendre son fonctionnement :

- structure d'une page php ;
- les bases du langage php ;
- variables, constantes, tableaux associatifs ;
- boucles et conditions ;
- manipulation des chaines de texte ;
- les fonctions ;
- manipulation des dates ;
- include, require.

## 3 Travail demandé

Dans chacune des sections suivantes, il faut bien lire, exécuter, tester et surtout comprendre le code php donné.

**N.B** : Le site de référence pour une description sur les différentes options et commandes php est : http://www.php.net

## 3.1 PHP syntaxe

- Le code PHP (appelé script PHP) peut être placé n'importe où dans le document.
- L'extension de fichier par défaut pour les fichiers PHP est ".php"
- Un fichier PHP contient normalement balises HTML et du script PHP.
- Ci-dessous, un exemple d'un fichier PHP « bonjour.php », avec du script PHP qui utilise une fonction intégrée à PHP qui s'appelle "echo" dont le rôle est de produire une sortie de texte sur la page web :

```
<!DOCTYPE>
<html lang="fr">
<head>
    <meta charset="utf-8"/>
    <title> première page PHP </title>
</head>
<body>
    <?php echo "<p> Bonjour tout le monde ! Ceci est déjà un
        script PHP </p>"; ?>
</body>
</html>
```

## 3.2 Les variables en PHP

- Une variable commence par le symbole $ suivi par le nom de la variable.
- Une variable peut avoir un nom court ($x, $y, ...). Cependant, on privilégiera un nom plus descriptif ($ageDuCapitaine, $total, $nom, $montantVerse).
- Un nom de variable ne peut pas commencer par un nombre.
- Un nom de variable ne peut contenir que des caractères alphanumériques, minuscule ou majuscule, sans accent et sans espace et le caractère de soulignement (a..z, A..Z, 0..9, _ ).
- Un nom de variable est sensible à la casse !
- Le type d'une variable est dynamique : il est déterminé par la valeur qui lui est affectée.
- L'affectation est réalisée par le signe égal "="
- La concaténation est réalisée avec un point "."

### Affichage des variables

La commande ***echo*** est souvent utilisée pour afficher les variables sur la page web

Exemple d'affichage d'un calcul d'emprunt (emprunt.php)

```
<!DOCTYPE html>
<html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>
<h1>Quelques affichages de variables</h1>

<?php
   // chaine de caractères
   $txt = "les CIR1" ;
   echo "Bienvenue " . $txt ."!" ;
?>

<div align='center'><img src="pics/emprunt.jpg" /></div>

<?php
   // calcul
   echo "<h1>Pret</h1>";
   $capital = 200000.0 ;
   $taux = 1.9 / 100 ;
   $duree = 240 ;
   $mensualite = $capital * ($taux/12) / ( 1 - pow(1+($taux/12),-$duree)) ;
   echo "Pour un capital de " . $capital . " euros<br/>";
   echo "emprunté sur une durée de " . $duree . " mois<br/>";
   echo "à un taux de " . (100 * $taux) . " %<br/>";
   echo "vous payerez une mensualité de " . round($mensualite,2) . " euros<br/>";
   $cout = $duree * $mensualite - $capital ;
   echo "le coût de votre prêt s'élève à " . round($cout,2) . " euros<br/>";
 ?>

</body>
</html>
```

## 3.3 Les commentaires en PHP

Un commentaire dans le code PHP est une ligne qui n'est pas lue (exécutée) dans le moteur PHP du serveur. Son seul but est d'être lu par quelqu'un qui lit le code source.

PHP supporte plusieurs manières de commenter:

/* .... jusqu'au prochain .... */
// ... jusqu'à la fin de la ligne

Exemple

```
<!DOCTYPE html>
<html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>

<h1>Voici ma premi&egrave;re page PHP</h1>
<?php
   // je suis un commentaire sur une seule ligne
   echo "<h3>Bonjour les <b>CIR1</b> !</h3>" ;
   echo "<p>une premi&egrave;re page avec du PHP.</p>" ;
?>

<p>Cette page est une page dynamique avec l'extension .php</p>

<?php
   $nombre = 3 ;
   echo "<p>Je suis encore du texte</p>";
   /* je suis un commentaire sur plusieurs lignes afin d'expliquer longuement ce que fait la suite pour
   me permettre de comprendre dans 10 ans ce que j'avais fait !
   le nombre 10 ci-dessous est commenté lui aussi */
   echo $nombre + /* 10 */ 100 ;
?>

</body>
</html>
```

## 3.4 La casse en PHP (case sensitivity)

En PHP, tous les mots clés (if, else, while, echo, ...) les classes, les fonctions et les fonctions définies par l'utilisateur ne sont pas sensibles à la casse.

Dans l'exemple ci-dessous, seule la première déclaration faisant référence à ***$maCouleur*** sera affichée avec la valeur de la variable **$maCouleur**. Les autres affichages produiront une erreur car les variables ***$MACOULEUR***, ***$macouleur*** et ***$MaCouleur*** ne sont pas définies.

```
<!DOCTYPE html>
<html>
 <body>
 <h1>La casse en PHP</h1>
 <?php
    /* ici pas de problème sur la casse */
    ECHO "<h1>Bonjour les CIR1</h1>";
    EchO "<h1>Bonjour les CIR1</h1>";
    eCHO "<h1>Bonjour les CIR1</h1>";
    /* ici, sur les variables, ATTENTION à la casse ! */
    $maCouleur = "Rouge" ;

   echo "sa voiture est " . $maCouleur . "<br/>" ;   /* syntaxe OK */
   echo "sa maison est " . $MACOULEUR . "<br/>" ;    /* erreur */
   echo "son avion est " . $macouleur . "<br/>" ;    /* erreur */
   echo "sa moto est " . $MaCouleur . "<br/>" ;  /* erreur */
?>
 </body>
 </html>
```

*Remarque :* Dans l'exemple ci-dessus, les trois déclarations de ECHO, Echo et eCHO sont équivalentes. **MAIS** tous les ***noms de variables*** sont sensibles à la casse.

## 3.5 Les chaînes de caractères en PHP (PHP String)

Les guillemets (double quotations marks) et les apostrophes (single quotation marks) sont utilisées pour délimiter les chaines de caractères:

Dans les instructions PHP:

```
<?php
   echo "bonjour" ;
   echo 'Bonjour';
?>
```

Dans les variables PHP :

```
<?php
   $texte = "bonjour" ;
   $titre = 'Vol 714 pour Sydney' ;
?>
```

**La concaténation à l'aide de l'opérateur . (point)**

Concaténer = enchainer, mettre bout à bout au moins deux chaines ou une chaine et autre chose...

```php
<?php

$nb = 12 ;
$ingredient = "oeufs" ;
$texte = "Achetez-moi ". $nb . " " . $ingredient . " frais" ;

// --> $texte contiendra la chaine "Achetez-moi 12 oeufs frais"


$annee = 1 ;
$cycle = "CIR" ;
echo "Bonjour les " . $cycle . $annee ;   // affichera Bonjour les HEI3

echo " Bonjour les <b>" . $cycle . "</b>". $annee ;     // affichera Bonjour les CIR1
                                    // avec CIR en gras

echo ' Bonjour les <b>' . $cycle . "</b>". $annee ;     // affichera Bonjour les CIR1
                                    // avec CIR en gras

?>
```

Remarque sur la **différence entre ' et " :** Les guillemets (double quotations marks) provoquent l'interprétation des variables incluses directement dans la chaine de caractères. Les apostrophes (single quotation marks) non !

Exemple

```php
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple</title>
    <meta charset="UTF-8" />
  </head>
<body>
<h1>Différence entre simple et double quotes</h1>
<?php
  $nom1 = "HARLEY";
  $nom2 = 'DAVIDSON';

  echo "<h1>Utilisation des guillemets</h1>";
  echo "$nom1 $nom2" ;

  echo "<h1>Utilisation des apostrophes</h1>";
  echo '$nom1 $nom2' ;
?>
</body>
</html>
```

**Comment afficher les ' ou les " en PHP ?**

Deux méthodes :

1. Protection des guillemets ou apostrophes par le caractère d'échappement (back-slash) \

```
<?php
   echo "bonjour \"John\" " ;
   echo 'bonjour \'John\' ' ;
?>
```

2. Imbrication des guillemets dans les apostrophes ou inversement

```
<?php
   echo "bonjour 'John' " ;
   echo 'bonjour "John" ' ;
?>
```

## 3.6 Les structures de contrôle en PHP

Les structures de contrôle sont les éléments du langage qui déterminent l'ordre dans lequel les instructions sont exécutées.

La séquence (le fait de mettre des instructions les unes à la suite des autres) et les blocs d'instructions {...} font partie des structures de contrôle.

En PHP, on utilise :

- le test (ou conditionnelle) : exécute des instructions sous certaines conditions ;
- la boucle : répétition d'une partie des instructions.

Le test **if (...) { ... } else { ... }** est souvent utilisé lorsque vous souhaitez effectuer des actions différentes pour différentes décisions. Vous pouvez utiliser des instructions conditionnelles dans votre code pour ce faire. En PHP, les instructions conditionnelles sont :

- **if (...) { ... }** : exécute du code que si une condition spécifiée est vraie ;
- **if (...) { ... } else { ... }** : exécute du code si une condition est vraie et un autre code si la condition est fausse ;
- **if (...) { ... } elseif ( ...) else { ... }** :spécifie une nouvelle condition à tester si la première condition est fausse.

```
<!DOCTYPE html>
 <html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>
<h1>Structures de contrôle</h1>

<?php
  $t = date("H"); // cette fonction avec ce paramètre renvoie l'heure du serveur

  if ($t < "20")
    { echo "Have a good day !"; }
  else
    { echo "Have a good night !"; }
?>
</body>
</html>
```

La boucle **while (...) { ... }** commence à s'exécuter si la condition est vérifiée préalablement. Ensuite, le bloc d'instruction est exécuté tant que la condition est vrai.

*Il est donc ESSENTIEL de mettre dans le bloc d'instructions une instruction qui modifie la condition pour sortir de la boucle à un moment donné; sinon, vous créez une boucle infinie!*

```
<!DOCTYPE html>
 <html>
    <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>
 <h1>Structures de contrôle</h1>
            <?php
            $x = 0.0 ;
            while ( $x <= 1 )
            {
                        echo "cos(" . $x . ") = " . cos($x) . " <br/>";
                        $x = $x + 0.1 ;
            }
            ?>
</body>
</html>
```

La Boucle **do { ... } while (...)** exécutera toujours le bloc de code une fois, la condition sera vérifiée ensuite. La répétition du bloc d'instructions se fera tant que la condition spécifiée est vraie.

*Il est donc ESSENTIEL de mettre dans le bloc d'instructions une instruction qui modifie la condition pour sortir de la boucle à un moment donné; sinon, vous créez une boucle infinie!*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple</title>
    <meta charset="UTF-8" />
  </head>
<body>
<h1>Structures de contrôle</h1>

<?php
        $x = 0.0 ;
        do
                {
                        echo "cos(" . $x . ") = " . cos($x) . "<br/>";
                        $x = $x + 0.1 ;
                }
        while ( $x <= 1 )
?>
</body>
</html>
```

La boucle **for** permet d'exécuter un bloc de code un nombre de fois spécifié. Cette instruction est utilisée lorsque vous savez à l'avance combien de fois le bloc d'instructions doit être exécuté.

Syntaxe générale : **for (init compteur ; test compteur ; incrément compteur) { ... }**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple boucle for</title>
    <meta charset="UTF-8" />
  </head>
<body>
<h1>Structures de contrôle</h1>

<?php
        echo "<h2>Table de multiplication de 7 : </h2>";
        for ($cpt = 1 ; $cpt <= 100 ; $cpt++ )
                {
                        echo "7 x " . $cpt . " = " . ($cpt * 7) ;
                        echo "<br/>";
                }
?>
</body>
</html>
```

- Modifiez le code ci-dessus afin d'afficher chaque ligne de la table de multiplication sous forme d'une liste non ordonnée.

## 3.7 Les tableaux en PHP

Un tableau stocke plusieurs valeurs dans une seule variable... Un tableau (**array**) est donc une variable spéciale qui peut contenir plusieurs valeurs à la fois.

En PHP, il existe 3 sortes de tableaux :

- **tableau indexé** : tableau avec un index numérique. Pour accéder aux valeurs d'un tableau indéxé, on utilise une référence qui est un numéro d'index ;
- **tableau associatif** : tableau avec des clés nommées ;
- **tableau multidimensionnel** : tableau qui contient un ou plusieurs tableaux.

### 3.7.1 Tableau indexé

Il y a deux façons de créer des tableaux indexés: l'indice est attribué automatiquement (index commence toujours à 0) ou l'indice est attribué manuellement comme le montre l'exemple ci-dessous:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>

<body>
<h1>Les tableaux indéxés</h1>
 <?php
// ***************************************************************
// Attribution automatique de l'index dans la déclaration suivante
// Automatiquement le premier élément du tableau a pour index 0
// ***************************************************************
$motos = array("Harley","Yamaha","Honda","Kawazaki") ;

// ***********************************************************
// Attribution manuelle de l'index dans la déclaration suivante :
// ***********************************************************
$voitures[0] = "Aston martin" ;
$voitures[1] = "Bugatti";
$voitures[2] = "Rolls Royce";


// *******************************************
// affichage du contenu du tableau moto
// la fonction count renvoie la taille du tableau
// *******************************************
$taille = count($motos);

for($x = 0; $x < $taille; $x++)
   {
      echo $motos[$x];
      echo "<br/>";
   }
?>
</body>
</html>
```

- Afficher ensuite le contenu du tableau voitures.

### 3.7.2 Tableau associatif

Les tableaux associatifs sont des tableaux qui utilisent des clés nommées que vous attribuez arbitrairement. Il y a deux façons de créer des tableaux associatifs comme le montre l'exemple ci-dessous:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>
<h1>Les tableaux associatifs</h1>
<?php
// ************************************************
// 1ère méthode de déclaration d'un tableau associatif
// ************************************************
$point = array("Dad"=>"305", "Sam"=>"1437", "Gilou"=>"1209");

// ********************************************************************************
// 2ème méthode de déclaration d'un tableau associatif équivalente à la première
// ********************************************************************************
$point["Dad"] = "305" ;
$point["Sam"] = "1437" ;
$point["Gilou"] = "1209";

// ******************************************
// affichage du contenu du tableau associatif
// ******************************************
foreach ( $point as $indice => $valeur )
   {
      echo "clé : " . $indice . " , contenu = ". $valeur ;
      echo "<br/>";
   }
?>
</body>
</html>
```

### 3.7.3 Tableau multidimensionnel

C'est un tableau qui contient un ou plusieurs tableaux, exemple (un tableau à 2 dimensions) :

```
<!DOCTYPE html>
<html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>
<h1>Les tableaux multidimensionnels</h1>
<?php
// on peut imaginer un tableau à deux entrées
// la première colonne contient une liste d'article
// la deuxième colonne contient le prix unitaire
// la troisième colonne contient le stock
// chaque ligne est un article
$article = array(
            array("biscuits",2.30,710),
            array("chocolat",1.90,250),
            array("noisettes",2.70,251),
            array("sucre",2.0,370),
            array("oeufs",0.24,100),
            array("farine",0.95,150),
         );

// affichage ci-dessous (un peu brutal !!!)
// il est préférable d'utiliser des boucles imbriquées et de faire une présentation en table
echo $article[0][0] . ": PU: " . $article[0][1] . "€, stock: ".$article[0][2]. "<br/>";
echo $article[1][0] . ": PU: " . $article[1][1] . "€, stock: ".$article[1][2]. "<br/>";
echo $article[2][0] . ": PU: " . $article[2][1] . "€, stock: ".$article[2][2]. "<br/>";
echo $article[3][0] . ": PU: " . $article[3][1] . "€, stock: ".$article[3][2]. "<br/>";
echo $article[4][0] . ": PU: " . $article[4][1] . "€, stock: ".$article[4][2]. "<br/>";
echo $article[5][0] . ": PU: " . $article[5][1] . "€, stock: ".$article[5][2]. "<br/>";

// ci-dessous l'affichage conseillé
echo "<table border='1' cellpadding='5'>" ;
for ($ligne=0 ; $ligne < count($article) ; $ligne++)
{
      echo "<tr>";
      for ($colonne=0 ; $colonne < count($article[$ligne])/*3*/ ; $colonne++)
      {
         echo "<td>" ;
         echo $article[$ligne][$colonne] ;
         echo "</td>" ;
      }
      echo "</tr>";
}
echo "</table>";
?>
</body>
</html>
```

## 3.8 Les fonctions en PHP

Deux types de fonctions :

- les fonctions intégrées au langage PHP : PHP possède plus de 1000 fonctions intégrées (gestion de la date, gestion des bases de données, gestion des fichiers et des images, gestion des chaines de caractères...) ;
- les fonctions définies par l'utilisateur : l'utilisateur peut créer ses propres fonctions. Une fonction est un bloc d'instructions qui peut être utilisé à plusieurs reprises dans un programme. Une fonction ne sera pas exécutée immédiatement quand une page se charge. Une fonction sera exécutée par un appel à cette fonction.

*Exemple :* La fonction message() ci-dessous est une fonction définie par l'utilisateur.
Elle affiche un message et utilise une fonction date qui est une fonction intégrée au PHP. La fonction coutTTC() admet deux paramètres en entrée, et renvoie (return) une valeur de sortie.
*Pour connaitre les paramètres de la fonction intégrée date, reportez-vous à l'adresse suivante:* *http://php.net/manual/fr/function.date.php*

```
<?php
/* ********************************************** */
/* affichage d'un message de bienvenue avec la date */
/* ********************************************** */
function message() {
   echo "<h3>Good Morning and Welcome</h3>" ;
   echo "<h4>Today is " ;
   echo date("l d F Y") ;
   echo "</h4>";
}

/* ********************************************** */
/* Calcul du cout TTC à partir des valeurs HT et TVA */
/* ********************************************** */
function coutTTC ( $prixHT, $TVA ) {
   return ( $prixHT * (1+$TVA) ) ;
}
?>

<!DOCTYPE html>
<html>
   <head>
      <title>Exemple</title>
      <meta charset="UTF-8" />
   </head>
<body>
   <h1>fonction en PHP</h1>
   <?php
      message() ;  // appel de la fonction
   ?>
   <p> Prix TTC d'un PC  : <?php echo coutTTC(812,0.2) ; ?> euros </p>
</body>
</html>
```

## 3.9 Exercices

### 3.9.1 Exercice 1

Vérifier votre configuration PHP en utilisant la fonction `strftime()` comme vu en cours pour obtenir la date du jour.

```php
<?php
  @setlocale(LC_TIME, 'fr_FR');
  echo "<p> Au moment de l'exécution de ce script PHP, nous sommes le ".strftime('%A %d %B %Y').".</p>";
?>
```

- Créez une page `datedujour.php` valide W3C dont le titre de la page (balise <title>) est : **Date du jour.**
- A incorporant le code php ci-dessus dans le corps du HTML (body) de cette page, affichez la date en français. Un exemple d'affichage de votre page (selon la date du jour) est le suivant :

  Au moment de l'exécution de ce script PHP, nous sommes le jeudi 16 janvier 2020.

- Testez votre page et vérifier l'affichage de la date en français (si ce n'est pas le cas corriger le code).
- Complétez ce code pour afficher également les heures, minutes et secondes.
- Ajoutez dans votre code l'affichage d'un paragraphe affichant "Bonne journée !", "Bon après-midi !", "Bonne soirée !", ou "Bonne nuit." en fonction de l'heure.
- Quelle est la différence entre fonction date et la fonction strftime ? donnez votre réponse dans une nouvelle balise <p> dans le code php.
- Modifiez votre code et créez une fonction qui s'appelle **laDateDuJour()** qui s'occupera de l'affichage "Bonne journée !", "Bon après-midi !", "Bonne soirée !", ou "Bonne nuit." en fonction de l'heure.

Aide :

- documentation sur http://www.php.net , précisément sur https://www.php.net/manual/fr/function.strftime.php et sur https://www.php.net/manual/fr/function.date.php
- la variable $h étant correctement initialisée

```
...
if($h < 12){
    echo "<p> Bonne journée !</p>";
}else if
...
```

### 3.9.2 Exercice 2 (Inclusion de pages PHP : factorisation du code / modularité)

On reprend le premier site réalisé au TP n°3 du Web Statique. Il comporte 3 pages HTML avec le code suivant pour chaque page i, i allant de 1 à 3.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="site.css">
<title> Mon Site </title>
</head>
<body>
<div id="titre"> <p> Le titre du Site </p> </div>
<div id="soustitre"> <p> Quelque chose sous le titre </p> </div>
<div id="navgauche">
<ul>
<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>
</ul>
</div>
<div id="navcorps"> <p> Contenu lié à la page n°i </p> </div>
<div id="pieddepage"> <p> Blabla en bas de page </p> </div>
</body>
</html>
```

**Question 2.1 :** Pourquoi faut-il factoriser ce code ?
**Question 2.2 :** Écrire le contenu de toutes les pages HTML et PHP permettant de factoriser ce site. Il vous faut donc créer des pages contenant du code commun à plusieurs pages et les inclure avec la fonction PHP include quand nécessaire.

**Question 2.3 :** Toutes les pages créées ont-elles vocation à s'afficher dans le navigateur via leur url ?

### 3.9.3 Exercice 3

Écrire le code PHP/HTML/CSS permettant de réaliser le tableau suivant (il s'agit de la table de multiplication des nombres de 1 à 10).

Table de Pythagore

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Aide :

- Utilisez les boucles php (for par exemple) pour créer votre tableau.
- De combien de boucles vous aurez besoin pour afficher ce tableau à deux dimensions ?
- Il ne faut pas taper tout le code html (toutes les balises <tr> et <td>) pour ce grand tableau. Les boucles php devraient créer les balises <tr> (les lignes du tableau) et les balises <td> (les cellules du tableau) ainsi que les valeurs affichées dans ce tableau.
- N'oubliez pas le fichier CSS pour habiller votre tableau, ou bien d'employer du style en ligne.

### 3.9.4 Exercice 4

Créez une fonction qui s'appelle **plusGrand()**. Elle prendra deux arguments de type int. Elle devra retourner le plus grand des deux et l'afficher.

### 3.9.5 Exercice 5 (en utilisant des tableaux indexés)

5.a) Créez une fonction qui s'appelle **premierElementTableau()**. Elle prendra un argument de type array. Elle devra retourner le premier élément du tableau. Si l'array est vide, il faudra retourner null.

5.b) Ensuite Créez une fonction qui s'appelle **dernierElementTableau()**. Elle prendra un argument de type array. Elle devra retourner le dernier élément du tableau. Si l'array est vide, il faudra retourner null;

### 3.9.6 Exercice 6

Créez une fonction qui s'appelle **remplacerLettres()**. Elle prendra un argument de type string. Elle devra retourner ce même string mais en remplaçant les e par des 3, les i par des 1 et les o par des 0 Exemple :

input : "Bonjour les amis" ==> Output : B0nj0ur l3s am1s

input : "Les cours de programmation Web sont trops cools" ==> Output : L3s c0urs d3 pr0grammat10n W3b s0nt tr0ps c00ls

### 3.9.7 Quiz PHP débutant

Pour finaliser ce TP, Complétez vos connaissances PHP avec le Quiz PHP débutant Alsacréations. Refaites le quiz pour avoir le maximum de point au besoin.