

# R3.1 - PHP

## TP1 Intro

### Environnement de travail

Vérifiez que votre environnement est fonctionnel :

- Installation **sur votre machine personnelle**
  - **Laragon**, XAMPP, EasyPHP ou WampServer (Windows)
  - MAMP (Mac)
  - LAMP ou installation manuelle (Linux)
- Réglage du **document root** :
  - Configurer dans Laragon : **document\_root** sur Z:\public\_html (machine IUT) ou sur le répertoire de travail de votre choix
  - Créer les dossiers **data, images, fichiers, includes** dans le document root
  - Lancer Laragon
  - Démarrer le serveur apache (bouton *start*)
  - Test sur **le localhost** (bouton *web*)



Code

**Créer le dossier TP1, y créer un script **test.php** avec une balise php et un affichage pour voir si tout fonctionne.**

```
<?php
    echo " Ça fonctionne ";
?>
```

**Observer le résultat, vérifier le code source dans le navigateur et observer l'en-tête.**



Liste des fichiers à produire :

- tp1.php contenant chacun des exercices, titrés avec une balise <h1>

# Exercice 1

## Génération de HTML conditionnelle.

La syntaxe du langage PHP est assez similaire à celle du C, Java, Javascript, C#.

Les conditions et boucles permettent de rendre dynamique le contenu de la page, en fonction de certaines données, de générer ou non certaines portions de code html, etc.

```
<html>
<body>
<?php
    if( $macondition ) {
?>
    <code html si la condition est vraie/>
<?php
    }else{
?>
    <code html si la condition est fausse.../>
<?php
    }
?>
</body>
</html>
```



Code

**1. Créer une variable initialisée avec une valeur aléatoire**

**2. Ecrire un test sur cette condition : est-ce que la valeur générée est paire,**

- si condition vraie, afficher un *paragraphe* avec la valeur donnée et le texte **PAIR**
- si condition fausse, afficher un *div* avec la valeur donnée et le texte **IMPAIR**

**Rafraichir la page et observer le code source dans le navigateur.**



Aide

L'opérateur modulo (%) permet de connaître si un nombre x est pair (  $x \% 2 == 0$  ) ou impair (  $x \% 2 == 1$  )

La fonction *rand(x,y)* renvoyer une valeur aléatoire entre x et y

## Exercice 2

### Génération HTML à partir d'une boucle

```
<html>
<body>
<?php
    for( paramètres ) {
?>
    <code html répété en boucle>
<?php
    }
?>
</body>
</html>
```

#### Code

1. Créer une variable  $n$  contenant un nombre entier  $> 5$
2. Créer une balise `<select>` contenant  $n$  options
3. Ajouter l'attribut `selected` sur l'option dont la valeur est 2

Modifier la valeur de la variable, rafraichir la page et observer la liste déroulante générée.

#### Aide

Le code généré doit ressembler à :

```
<html>
<body>
    <select>
        <option>1</option>
        <option selected>2</option>
        <option>3</option>
        ...
    </select>
</body>
</html>
```

## Exercice 3

### Génération HTML à partir d'une boucle

#### Code

1. Créer une variable  $n$  contenant un nombre entier
  2. Créer une table HTML contenant  $n$  lignes de deux colonnes : dans la première la valeur de  $i < n$ , dans la seconde si  $i$  est paire ou impaire
  3. Si  $n = 0$ , ne pas afficher la table, mais la chaîne « aucune donnée à afficher »
- Modifier la valeur de  $n$ , rafraichir la page et observer le contenu généré.

```
<html>
<body>
<?php
    for( paramètres ) {
?>
    <code html répété en boucle>
<?php
    }
?>
</body>
</html>
```

#### Aide

Le code généré doit ressembler à :

```
<html>
<body>
    <table>
        <tr>
            <td>1</td><td>IMPAIR</td>
        </tr>
        <tr>
            <td>2</td><td>PAIR</td>
        </tr>
        ...
    </table>
    OU
    <div>Aucune donnée à afficher</div>
</body>
</html>
```

## Exercice 4

---

### Guillemets simples, doubles, interprétations de variables

Voici quelques définitions de variables:

```
titre = PHP "avancé"  
vignette = Z:\news\image\oreilly.png  
edition = O'Reilly  
prix = $US 15
```



Code

**Créer un script qui initialise ces chaînes à l'aide d'un tableau associatif, puis générer l'affichage du tableau dans une table HTML**

titre	PHP "avancé"
vignette	Z:\docs\image\oreilly.png
edition	O'Reilly
prix	15 \$US

## Exercice 5

---

### Manipulation simple de chaînes



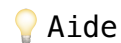
Code

**Créer un script qui, au format texte,**

- affiche la longueur de l'adresse mail
- sépare l'utilisateur et le nom de domaine d'une variable contenant une adresse mail.

**On suppose que l'adresse mail est valide, une vérification sera faite à l'aide des expressions régulières ou de fonctions spécifiques dans un prochain TD.**

david.durand@u-picardie.fr      →      david.durand      u-picardie.fr



Aide

`strlen($chaîne)` : nombre de caractères contenus dans `$chaîne`

`substr($chaîne, $position, $nb)` : extrait `$nb` caractères à partir de `$position`

La fonction `strpos(chaîne, recherche)` renvoie :

- la position de la chaîne `recherche` au sein de la chaîne `chaîne`, ou
- `FALSE` si celle-ci n'est pas trouvée.

## Exercice 6

---

### Fichiers, tableaux et boucles.

Un fichier est un ensemble de caractères stocké sur un support. Dans le cas d'un fichier texte, on peut le considérer comme une chaîne ou bien comme un tableau de lignes.

#### Code

**1. Créer dans votre dossier `fichiers` un fichier `.txt` contenant une liste d'URL, par exemple**

```
http://www.u-picardie.fr
http://duckduckgo.com
http://codepen.io/ge1doot/
```

**2. A partir du fichier, générer un fichier HTML qui affiche ces liens web sous forme cliquable.**

#### Aide

La fonction `file($filename)` retourne un tableau contenant chaque ligne du fichier `$filename` (une ligne est un ensemble de caractères terminés par `\r`, `\n` ou les deux)

La balise pour les liens est *anchor* : `<a href='url'>texte affiché</a>`

## Exercice 7

---

### Tableaux, fichiers, et découpage de chaîne

#### Code

- 1. Compléter le fichier texte précédent en y ajoutant une description à la suite de chaque lien. Choisir un caractère approprié qui permet de séparer le lien de sa description**
- 2. Générer un fichier HTML qui affiche les descriptions sous forme de liens cliquables vers l'url.**
- 3. Alternier en clair/foncé chacune des lignes affichées**

#### Aide

La fonction `explode ( $separateur, $chaîne)` retourne un tableau comprenant chaque sous chaîne de `$chaîne` séparées par le caractère `$separateur`.

Une `classe CSS` permet d'appliquer un style (couleur de fond, de texte...) à un ensemble de balises.

```
<style>
.maclasse{ background: #FF3388}
</style>
...
<p class='maclasse'>Lorem ipsum dolor sit amet</p>
```

## Exercice 8

### Tableau associatif

PHP propose diverses variables vous permettant d'obtenir des renseignements sur l'environnement et la requête en cours : `_SERVER`, `_GET`, `_POST`, `_FILES`, `_COOKIE`, `_SESSION`, `_REQUEST`, `_ENV`.

Exemple : voici un extrait des infos de la variable `$_SERVER`

Nom de la clé	Description
<code>\$_SERVER["SERVER_NAME"]</code>	nom du serveur
<code>\$_SERVER["DOCUMENT_ROOT"]</code>	racine du serveur
<code>\$_SERVER["HTTP_HOST"]</code>	hôte du client
<code>\$_SERVER["HTTP_REFERER"]</code>	url de la page précédente (si disponible)
<code>\$_SERVER["HTTP_USER_AGENT"]</code>	infos du navigateur
<code>\$_SERVER["REMOTE_ADDR"]</code>	adresse IP client
<code>\$_SERVER["SCRIPT_FILENAME"]</code>	nom du fichier PHP
<code>\$_SERVER["REQUEST_URI"]</code>	chemin d'accès utilisé
...	...

### Code

**Créer un script qui affiche toutes les clés/valeurs du tableau associatif `_SERVER` dans un tableau HTML.**

### Aide

Vous pouvez aussi avoir accès à toutes ces informations à l'aide de la fonction `phpinfo()`.

La boucle `foreach` permet de parcourir les tableaux associatifs

```
<?php
$tab=array("fr"=>"français","en"=>"anglais","de"=>"allemand")
foreach ($tab as $cle => valeur)
...
?>
```

## Exercice 9

---

### Générer d'autres type de sorties

#### Code

1. **Modifier l'en-tête du fichier pour passer en texte brut. Quel résultat observez-vous ?**
2. **Créer un fichier json-gen.php, convertir le tableau \$\_SERVER en json, puis afficher le contenu obtenu. Observez le résultat.**
3. **Modifier l'en-tête pour passer en mode JSON. Quel résultat obtient-on cette fois ?**

#### Aide

- Le **type mime** est utilisé pour identifier le type de données contenu dans une ressource (fichier, flux...), par exemple :

text/html, text/plain, application/json, image/jpeg, etc.

- La fonction php **header()** permet de modifier le type mime, réglé par défaut à text/html
- l'entête à modifier est : **Content-type**

<https://www.php.net/manual/fr/function.header.php>

[https://developer.mozilla.org/fr/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_Types](https://developer.mozilla.org/fr/docs/Web/HTTP/Basics_of_HTTP/MIME_Types)

## Exercice 10

---

### Recherche de chaine : test du navigateur

On veut afficher dans une page HTML les informations qui indiquent à l'utilisateur son système et son navigateur. Ces informations permettent d'établir des statistiques, inviter l'utilisateur à télécharger une application pour la bonne plateforme, etc.

#### Code

1. **Chercher sur le web les icônes des principaux navigateurs (Linux, Mac, Windows) et OS Chrome, Firefox, Safari...), et placez-les dans votre dossier [images](#).**
2. **Créer un script qui affiche les icônes correspondant au navigateur et au système du client.**

#### Aide

Le navigateur envoie lors de chaque requête une chaîne d'identification appelée [user agent](#). Cette information est accessible en PHP à travers la variable d'environnement HTTP\_USER\_AGENT

```
$ua = $_SERVER['HTTP_USER_AGENT'];
```



## Frise chronologique simplifiée des différents navigateurs.

