

# TD1-2 : Premiers pas en PHP

## V2.2.0

---



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Document en ligne : [www.mickael-martin-nevot.com](http://www.mickael-martin-nevot.com)

---

## 1 Généralités

Vous visualiserez systématiquement votre travail dans différents navigateurs Web.

N'oubliez pas de faire des recherches sur le Web à chaque fois que cela est nécessaire en prenant soin de vérifier que les informations trouvées soient correctes.

Vous trouverez la boîte à outils ainsi que l'ensemble des documents nécessaires à la réalisation de ce TD sur le site Web de l'enseignant.

## 2 Découverte du PHP

Créez une page PHP, décrivez une structure HTML et dans le corps de la page Web, introduisez le code suivant :

```
<?php
    echo '<strong> Voici mon premier programme PHP </strong><br/>' . "\n";
    echo 'C\'est pas mal non ? <br/>';
?>
```

Ajoutez une déclaration de variable (par exemple au début du fichier) :

```
<?php
    $var1 = 6;
    $var2 = 1.3;
    $var3 = 'Variable 3';
?>
```

À la suite du code précédent, ajoutez l'affichage suivant (déterminez bien la différence entre ces deux lignes) :

```
<?php echo "$var1 + $var2"; ?>
<?php echo $var1 + $var2; ?>
```

Essayez enfin l'affichage de cette ligne (comprendre ce qu'il se passe) :

```
<?php echo $var1 + $var3; ?>
```

### 3 Mutualisation du code

Les fonctions en PHP permettent de ne pas réécrire toujours le même code, par exemple on peut faire une fonction PHP pour écrire le début et la fin d'une page Web.

Au début du fichier :

```
<?php
function start_page($title)
{
    echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"><head><title>' . "\n"
. $title . '</title></head><body>' . "\n";
};
?>
```

Puis :

```
<?php
start_page('titre');
?>
```

Créez un fichier PHP qui contiendra deux fonctions : `start_page(...)` et `end_page()`. Cette page Web devra avoir comme titre « Test » et un corps HTML du genre :

```
<hr/><br/><strong>Test</strong><br/><hr/>
```

### 4 Date

Dans le fichier précédent, ajoutez l'affichage de la date du jour de la manière suivante :

```
<?php
echo date('l F d, Y');
?>
```

Essayez ensuite de mémoriser la date dans une variable puis de l'afficher :

```
<?php
$jour = date('l F d, Y');
?>
```

Traduisez une date d'un format anglais vers un format français :

```
<?php
$jour = date('d/m/Y', strtotime('2020-04-01'));
?>
```

#### Remarque

La fonction `strtotime(...)` transforme une chaîne de caractères en date et `date(...)` met en forme une date dans une chaîne de caractères.

Affichez la date du jour sous le format suivant :

12/03/2001

March 12, 2001, 10:16 pm

## 5 Calculatrice

### 5.1 Machine à calculer

Créez le fichier `calculator.php` qui utilisera les fonctions `start_page(...)` et `end_page()`. Ajoutez un titre à votre page et créez à l'intérieur un formulaire contenant deux zones de saisie que vous nommerez `op1` et `op2` puis une série de quatre boutons radio du même groupe (portant le même nom `op` par exemple) et ayant pour valeurs respectives `+`, `*`, `/` et `-` :

```
<input checked="checked" type="radio" name="op" value="*" />* <br/>
<input type="radio" name="op" value="+" />+ <br/>
<input type="radio" name="op" value="-" />- <br/>
<input type="radio" name="op" value="/" />/ <br/>
```

Ajoutez un bouton de soumission ainsi qu'un bouton d'effacement du formulaire. Le formulaire doit commencer de la manière suivante :

```
<form action="calcul.php" method="post">
```

Cela signifie que lorsque l'utilisateur cliquera sur le bouton `submit` le fichier `calcul.php` sera appelé et les valeurs du formulaire lui seront « postées ».

### 5.2 Calcul

Créez le fichier `calcul.php` en utilisant toujours les fonctions `start_page(...)` et `end_page()`. Pour récupérer les valeurs « postées », il suffit d'utiliser le tableau `$_POST` dans ce fichier :

```
<?php
    $op1 = $_POST['op1'];
    $op2 = $_POST['op2'];
    $op  = $_POST['op'];
?>
```

À partir de là, les variables `$op1`, `$op2` et `$op` contiennent les valeurs renseignées dans le formulaire. Pour l'instant, afin de tester votre formulaire, contentez-vous d'afficher dans cette page Web (et dans le titre) l'opération sous la forme (sans écrire le résultat pour l'instant) : `op1 * op2 =`

Pour réaliser l'opération, testez quel est l'opérateur saisi, par exemple :

```
<?php
    if('*' == $op)
    {

    }
    elseif('+ ' == $op)
    {

    }
    else
    {
        echo '<br/><strong>opérateur ' . $op . ' non géré </strong>';
    }
?>
```

## 6 Factorisation du code

Créez une variable qui contient la chaîne de caractères "\*+/" (sans espace) :

```
<?php
    $operateurs = '*+/' ;
?>
```

On peut atteindre chaque caractère de la chaîne comme avec un tableau, de la manière suivante : `$operateurs[0]` est égal à '\*', `$operateurs[1]` est égal à '+', etc.

Ensuite, à la place des boutons radio, écrivez une boucle du genre :

```
<?php
    for($cpt = 0 ; $cpt <= 3 ; ++$cpt)
    {
        echo '<input ' ;
        if($cpt == 0)
        {
            echo 'checked="checked" ' ;
        }
        echo 'type="radio" name="op" value="' . $operateurs[$cpt] . '"/>' . $op
erateurs[$cpt] . ' <br/>' . "\n";
    }
?>
```

Testez à nouveau votre formulaire.

## 7 Calculatrice améliorée (optionnel)

La calculatrice peut également être composée de quatre boutons différents. Pour ce faire :

- dans `calculator.php` : mettez un nom ("action") au bouton submit du formulaire ;
- dans `calcul.php` : récupérez la variable `$action` grâce au tableau `$_POST` et affichez-la.

Supprimez les lignes du formulaire déclarant les boutons radio et créez quatre boutons submit du même nom ("action") et portant les valeurs respectives "\*", "+", "/", "-" (testez ensuite la valeur de `$action` dans `calcul.php` afin de déterminer quel est l'opérateur saisi).

Vous pouvez utiliser la méthode HTTP GET à la place de POST. Pour ce faire :

- dans `calculator.php` : modifiez le formulaire en remplaçant "POST" par "GET" ;
- dans `calcul.php` : utilisez le tableau `$_GET` au lieu de `$_POST`.

Constatez la différence d'utilisation des deux méthodes en observant l'URL de la page Web lancée lorsque vous cliquez sur un bouton submit. Essayez de lancer la page Web en entrant directement une URL de la forme : `...calcul.php?op1=12&op2=22&action=* ;` puis déterminez ce qu'il faut en conclure.