Projet découverte du PHP – Découverte de PHP & Tables de multiplication.

Ressources :

https://github.com/michelonlineformapro/Projet-2-PHP-Operations

Projet individuel – temps estimés : 5 - 10 jours

Description du projet :

Le but du projet est de découvrir le syntaxe et les base du langage PHP et de réaliser une mini application de tables de multiplication avec le langage PHP.

Au cours de ce projet, les apprenants vont pouvoir se familiariser progressivement avec le langage PHP et apprendre à installer et à utiliser un serveur de test local (wamp, Xamp).

Individuellement, chaque apprenant doit réaliser les tâches suivantes :

- Un fichier pour afficher la table de multiplication de 3.
- Sélectionner une table de multiplication entre 1 et 10 dans une liste déroulante. Une fois validée, la table de multiplication sélectionnée s'affiche en dessous.
- Choisir la ou les table(s) de multiplication à afficher par check box et par un bouton "valider". La ou les table(s) de multiplication sélectionnée(s) s'affiche(nt) en dessous.
- Intégrer un mode révision : l'utilisateur doit répondre à une multiplication tirée au hasard dans la table sélectionnée.

OBJECTIF :

- Comprendre la différence entre site statique et dynamique
- Les sites **statiques** : réalisés en HTML et CSS, leur contenu ne peut être mis à jour que par le webmaster ;
- Les sites **dynamiques** : réalisés avec d'autres outils comme PHP et MySQL en plus de HTML et CSS, ils permettent aux visiteurs de participer à la vie du site, de poster des messages… bref, de rendre le site vivant !
- Le fonctionnement client – server : Les visiteurs du site sont appelés les *clients*. Ils demandent au serveur qui héberge le site de leur transmettre les pages web.
- PHP est un langage exécuté par le serveur. Il permet de personnaliser la page en fonction du visiteur, de traiter ses messages, d'effectuer des calculs, etc. Il génère une page HTML.
- Wamp et Xamp = Apache (Server Web locale de test) + langage PHP (permet au serveur d'exécuter des pages PHP et de retourner de HTML pour le navigateur) + MySQL (SGBDR Système de base de données relationnelle)
- La syntaxe PHP + afficher du texte (echo) + comprendre le debug (var_dump) + afficher des commentaires
- Comprendre les variables scalaires ($) + affecter des valeurs aux variables + afficher et concaténer des variables + faire des calculs simples

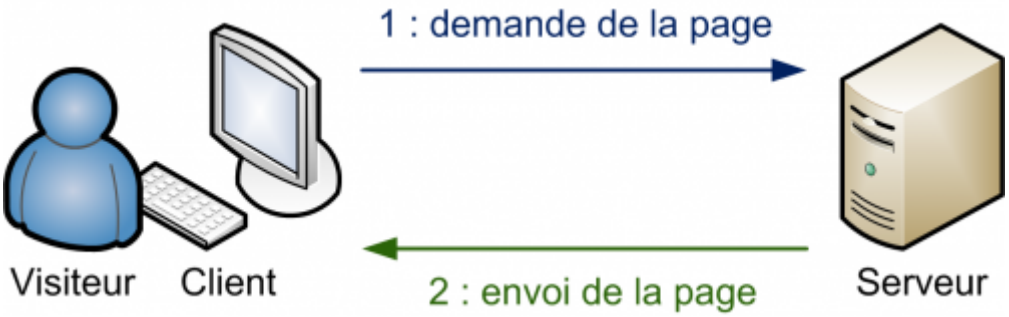
- Comprendre et appliquer des conditions (if…elseif…else, switch case et les ternaire)
- Comprendre et créer des boucles (while et for)
- Comprendre et créer des tableaux (les 2 types : numérotés et associatif)
- Parcourir et rechercher dans des tableaux
- Comprendre et créer des fonctions
- Les fonctions prête à l'emploi PHP
- Inclure des potions pages
- Les bonnes pratiques (nom claires, indentation, commentaires + standard)

LES BASES (Ressources) :

https://github.com/michelonlineformapro/Remise-niveau-PHP

https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql

UN SITE SATATIQUE :



UN SITE DYNAMIQUE :

PHP

La concurrence :

- **ASP.NET** : conçu par Microsoft, il exploite le framework (c'est-à-dire un ensemble de bibliothèques qui fournissent des services pour les développeurs) .NET bien connu des développeurs C#. Ce langage peut être intéressant si vous avez l'habitude de développer en C#.NET et que vous ne voulez pas être dépaysé ;
- **Ruby on Rails** : ce framework s'utilise avec le langage Ruby et permet de réaliser des sites dynamiques rapidement en suivant certaines conventions ;
- **Django** : il est similaire à Ruby on Rails, mais il s'utilise en langage Python ;

- **Java et les JSP (Java Server Pages)** : plus couramment appelé « **JEE** » ou « **Java EE** », il est particulièrement utilisé dans le monde professionnel. Il demande une certaine rigueur. La mise en place d'un projet JEE est traditionnellement un peu plus longue et plus lourde, mais le système est apprécié des professionnels et des institutions. C'est ce qui est utilisé sur le site des impôts français, par exemple.

Les concurrents de MySQL

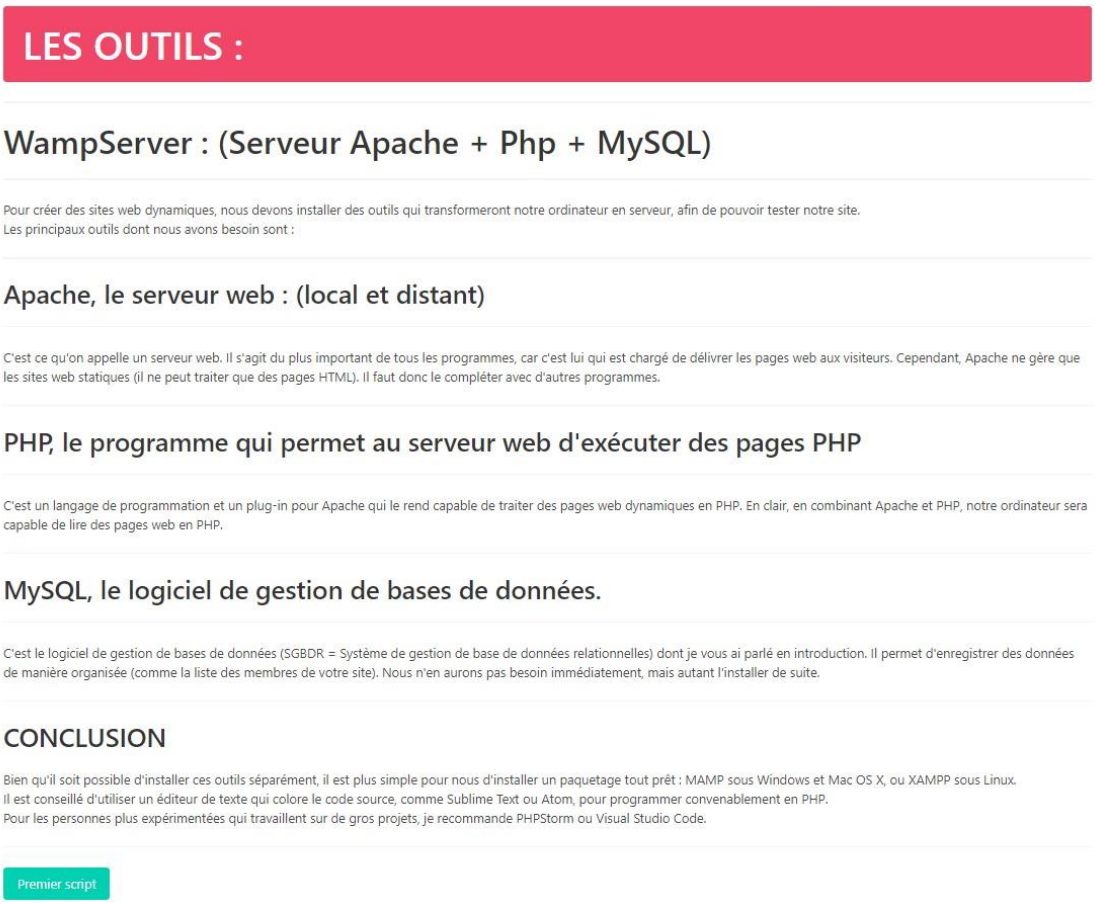
En ce qui concerne les bases de données, le choix est là encore très vaste.
Cependant, alors que PHP et ses concurrents sont la plupart du temps libres et gratuits, ce n'est pas le cas de la plupart des SGBD.

Parmi les concurrents de MySQL, je vous conseille de connaître (au moins de nom) les suivants :

- **Oracle** : c'est le SGBD le plus célèbre, le plus complet et le plus puissant. Il est malheureusement payant (et cher), ce qui le réserve plutôt aux entreprises qui l'utilisent déjà massivement. Il existe cependant des versions gratuites d'Oracle, notamment pour ceux qui veulent apprendre à s'en servir ;
- **MariaDB** : variante libre de MySQL, qui a été créée depuis que ce dernier a été racheté par... Oracle. Oui, je sais, on peut penser que ce monde est fou ! MySQL appartient à Oracle, mais reste bien une base de données à part. MariaDB est une copie (on dit un *fork*) de MySQL qui a la volonté de rester libre et indépendante ;
- **Microsoft SQL Server** : édité par Microsoft, on l'utilise souvent en combinaison avec ASP.NET, bien qu'on puisse l'utiliser avec n'importe quel autre langage. Il est payant, mais il existe des versions gratuites limitées ;
- **PostgreSQL** : il s'agit d'un SGBD libre et gratuit comme MySQL, qui propose des fonctionnalités plus avancées. Parfois comparé à Oracle, il lui reste cependant du chemin à parcourir.
- Il existe deux types de sites web :
    - Les sites **statiques** : réalisés en HTML et CSS, leur contenu ne peut être mis à jour que par le webmaster ;
    - Les sites **dynamiques** : réalisés avec d'autres outils comme PHP et MySQL en plus de HTML et CSS, ils permettent aux visiteurs de participer à la vie du site, de poster des messages… bref, de rendre le site vivant !
- Les visiteurs du site sont appelés les *clients*. Ils demandent au serveur qui héberge le site de leur transmettre les pages web.
- PHP est un langage exécuté par le serveur. Il permet de personnaliser la page en fonction du visiteur, de traiter ses messages, d'effectuer des calculs, etc. Il génère une page HTML.
- MySQL est un système de gestion de bases de données. Il se charge du stockage des informations (liste des messages, des membres…).

L’environnement de travail Wamp Server ou Xamp (Mamp pour MAC et Lamp pour les distribution Linux)

- Pour créer des sites web dynamiques, nous devons installer des outils qui transformeront notre ordinateur en serveur, afin de pouvoir tester notre site.
- Les principaux outils dont nous avons besoin sont :
    - Apache, le serveur web ;
    - PHP, le programme qui permet au serveur web d'exécuter des pages PHP ;
    - MySQL, le logiciel de gestion de bases de données.
- Bien qu'il soit possible d'installer ces outils séparément, il est plus simple pour nous d'installer un paquetage tout prêt : MAMP sous Windows et Mac OS X, ou XAMPP sous Linux.
- Il est conseillé d'utiliser un éditeur de texte qui colore le code source, comme Sublime Text ou Atom, pour programmer convenablement en PHP.
  Pour les personnes plus expérimentées qui travaillent sur de gros projets, je recommande PHPStorm ou Visual Studio Code.



LES BALISES PHP DE BASE :

php

```php
<?php
/* Le code PHP se met ici
Et ici
Et encore ici */
?>
```

LES BALISES:

Le code PHP vient s'insérer au milieu du code HTML. On va progressivement placer dans nos pages web des morceaux de code PHP à l'intérieur du HTML.
Ces bouts de code PHP seront les parties dynamiques de la page, c'est-à-dire les parties qui peuvent changer toutes seules (c'est pour cela qu'on dit qu'elles sont dynamiques).

```php
    <p>La balise PHP ouvrante / fermante est la suivate :</p>
    <?php

    ?>
<p>Injecter votre contenu dans l'HTML
    <?php echo 'Du texte' ?> ou <?= 'Du texte' ?>
</p>
<p>Il est possible d'ajouter des balise HTML dans les balises PHP, exemple :</p>
    <?php echo '<p>Un paragraphe</p>' ?>
<p>
```

PHP langage coté serveur :

Sur la figure suivante, vous découvrez concrètement ce qu'il se passe avec notre code source.
Le code PHP est exécuté en premier et l'ordinateur fait ce qu'on lui demande. Ici, on lui a dit « Affiche ce texte ici ».
Une fois toutes les instructions PHP exécutées (ici c'était simple, il n'y en avait qu'une !), la page qui sort est une page qui ne contient que du HTML ! Le PHP est intrerprété par le navigateur et transformé en HTML. C'est cette page de « résultat » qui est envoyée au visiteur, car celui-ci ne sait lire que le HTML.

En résumé

- Les pages web contenant du PHP ont l'extension .php
- Une page PHP est en fait une simple page HTML qui contient des instructions en langage PHP.
- Les instructions PHP sont placées dans une balise ouvrante et fermante :
- Pour afficher du texte en PHP, on utilise l'instruction echo
- Il est possible d'ajouter des commentaires en PHP pour décrire le fonctionnement du code. On utilise pour cela les symboles // ou /* */ .

Les variables

LES VARIABLES :

Une variable, c'est une petite information stockée en mémoire temporairement. Elle n'a pas une grande durée de vie. En PHP, la variable (l'information) existe tant que la page est en cours de génération. Dès que la page PHP est générée, toutes les variables sont supprimées de la mémoire car elles ne servent plus à rien. Ce n'est donc pas un fichier qui reste stocké sur le disque dur, mais une petite information temporaire présente en mémoire vive.
Une variable en PHP commence par le symbole $ entre deux balise PHP

Elles sont de différent types :
String = chaine de caractères : $age = "Du texte";
Integer = chiffre ou nombre entier : $age = 1; ou $age = 25; (et entier relatif $longitude = -15;)
Float = chiffre ou nombre à virgules : $pi = 3.14
Booléens = Valeur qui permet de stocker soit vrai soit faux. Cela permet de retenir si une information est vraie ou fausse : $estMajeur = true;
Rien (null) = Parfois on a besoin de déclaré des variables nuls (absence de typage) : $pasDeValeu = null;

Résumé : une variable PHP = $ + nom (type) = valeur + ;
On a donc un système de type clé/valeur

Afficher des varaiables

C'est trés simple
```
$prenom = "Michael";
echo $prenom;
```

La concaténation

Il faut écrire la variable en dehors des guillemets et séparer les éléments les uns des autres à l'aide d'un point
Cette methode permet de mélanger des variable PHP et de l'HTML.
```
$prenom = "michael";
$age = 152;
echo 'Bonjour' . $prenom .' tu as : ' . $age . ' ans. ';
```

Les opréartions (+ - * /)

On peut réalisé des opérations sur les varaiables
```
$addition = 4 + 3;
$a = 10;
$b = 15;
$c = $a + $b;
echo $c;
```

Les conditions: if else elseif switch

Les opérateurs (== != > < >= <=)

La structure est la suivante :
Si la condition est juste = on affiche un résultat SINON autres résultats :
if(condition){ résultat } else { résultat }
$age = 18;
if($age >= 18){
echo "Vous êtes majeur";
}else{
echo "Vous êtes mineur";
}
La structure peut être plus complexe à l'aide de if elseif else

---

La condition peut prendre tous type de varaiable :
$peutEntrer = true;
if($peutEnter == true){ echo "Fait ceci";} else {echo "Fait cela";}
Un booléen peut prendre la valeur false comme ceci :
if(!$peutEntrer) = à if($peutEnter == false)

Des conditions multiples

Les mot clés

Et (AND) = && (deux esperluettes)
Ou (OR) = || (Alt Gr + 6)
Exemple :
if($age > 18 && $nom == "Marc"){ résultat }

Le Swtich Case

Pour évité des cas de condition trop longue, uutilse le switch case
Une variable $age = 3;
La structute :
switch(condition){ les différents cas }
Exemple :
switch($age){
case 1:
echo "Ceci s'affiche si $age = 0"; sinon
break; Ici si le cas 1 n'est pas valide on stop la consition
Si $age = null
On utilise le cas par defaut
default: echo "Ceci s'affiche si aucun cas ne correspond";

```php
<?php
$note = 10;

switch ($note) // on indique sur quelle variable on travaille
{
    case 0: // dans le cas où $note vaut 0
        echo "Tu es vraiment un gros nul !!!";
    break;

    case 5: // dans le cas où $note vaut 5
        echo "Tu es très mauvais";
    break;

    case 7: // dans le cas où $note vaut 7
        echo "Tu es mauvais";
    break;

    case 10: // etc. etc.
        echo "Tu as pile poil la moyenne, c'est un peu juste…";
    break;

    case 12:
        echo "Tu es assez bon";
    break;

    case 16:
        echo "Tu te débrouilles très bien !";
    break;

    case 20:
        echo "Excellent travail, c'est parfait !";
    break;

    default:
        echo "Désolé, je n'ai pas de message à afficher pour cette note";
}
?>
```

Les cas if else et le ternaire

Un version racourcie d'un if else est possible :
$majeur = ($age >= 18) ? true : false;

Les boucles

LES BOUCLES :

Les boucles simples Tant Que :

Les instructions sont répétées tant que la conditions est vraie
Exemple :
$ligne = 1;
while($ligne <= 100){
echo "On incrémente le nombre de ligne de 1 à 100, rappel pour incrémenté on utilise la variable ++, c'est à dire $ligne++; ou $ligne += 1;
$ligne++;
}
Une fois à 100 : la boucle s'arrête.

Les boucles for :

C'est une forme condensée des boucles
for(Initialiser; condition; incrementation){ résultat }
Exemple :
for($repetition = 0; $repetition <= 100; $repetion++){
echo "On va ecrire cette phrase : " .$repetiton. " fois !";

Les tableaux

LES TABLEAUX :

2 Types : numéroté et associatif :

Les tableaux numérotés sont de types clés + valeurs, ils sont initialisé à l'aide du mot clé array()
Exemple :
$prenom = array("michael", "bob", "sophie", etc...)
On récupère les valeurs ainsi
$prenom[0]; $prenom[1]; etc...
Ici le premier élément du tableau prend l'index (la clé) 0 (et non 1) qui est égale à "michael".

Les tableaux Associatif :

On utilise toujours le mot clé array() mais ici on ^récise la clé et la valeur.
Exemple :

```
$livre = array(
'titre' => 'Tintin',
'auteur' => 'Hergé',
'nbrPage' => 25
);
```

Ensuite pour afficher un résultats :

```
echo $livre['auteur'];
```

Pour parcourir un tableau, on utilise une boucle for ou foceach
Avec notre exemple de livre

```
for($nombre = 0; $nombre <= nombre d'élément 2 (ici 3 éléments donc index 0, 1, 2); $nombre++){
echo "Lecture des éléments : " .$nombre[$livre];
AVEC FOREACH
foreach($livre as $nbrLivre){
echo $nrbLivre;
```

On constate que pour lire les données d'un tableau la boucle foreach est beaucoup plus simple.

Vérifier si une clé existe

```
if(array_key_exists('nom', $nomtablau)){
echo "la clé 'nom' existe dans le tableau";
} else {
echo "Cette clé n'existe pas !";
C'est identique pour une valeur :
if(in_array("Tintin", $livre){
echo "Ce livre" . $livre[$titre] . " existe dans le tableau";
} else {
echo "Ce livre n'existe pas dans le tableau";
```

Pour récupérer la clé d'une valeur **array_search**

Les fonctions

LES FONCTIONS :

Une fonction est une série d'instructions qui effectue des actions et qui retourne une valeur.
En général, dès que vous avez besoin d'effectuer des opérations un peu longues dont vous aurez à nouveau besoin plus tard, il est conseillé de vérifier s'il n'existe pas déjà une fonction qui fait cela pour vous.
Et si la fonction n'existe pas, vous avez la possibilité de la créer.

Les fonctions native PHP

strlen : longueur d'une chaine
str_replace: rechercher et remplacer
str_shuffle : réalise un mélange
strtolower : ecrire en minuscule
LES DATES
Les dates sont au format H = heures, i = minutes, d = jour, m = mois, Y = année
Exemple : $jour = date("d");

Les fonctions : function MaFonction(){}
Exemple

```
$nom = "TRUMP";
function DireBonjour($nom){
echo "Bonjour Donald " .$nom;
}
```

On utlise la fonction comme ceci : DireBonjour("prenom");

Une fonctions et des paramètres

```
function CalculCarre($valeur1, $valeur2){
$resultat = $valeur1 * $valeur2;
return $resultat;
}
Pour utilisé la fonction : CalculCarre(5,5);
echo "Résultat de l'opération = " .$resultat;
```

Des portions de pages

ACCUEIL EXERCICES Documentation Php.net

LES INCLUDES, REQUIRE et REQUIRE_ONCE :

Eviter les répétition (DRY Dont repeat Yourself)

En PHP, nous pouvons facilement insérer d'autres pages (on peut aussi insérer seulement des morceaux de pages) à l'intérieur d'une page.
Le principe de fonctionnement des inclusions en PHP est plutôt simple à comprendre. Vous avez un site web composé de, disons, vingt pages. Sur chaque page, il y a un menu, toujours le même. Pourquoi ne pas écrire ce menu (et seulement lui) une seule fois dans une page menus.php ?

---

include : inclus une portion de page, si elle n'existe pas le site va l'ignoré.
require : inclus une portion de page, si elle n'est pas trouvé le site retourne une erreur 404.
require_once : inclus une portion de page une seule fois, si elle n'est pas trouvé le site retourne une erreur 404.

Dans un site on utlise une navbar commune à toutes les pages

1 - On créer un fichier .php : ex: menu.php
2 - Cette page va contenir le code HTML de la barre de navigation.
3 - Dans chaque page, on va utilisé :

```
require_once "views/navbar.php";
```

Des données dans URL

ACCUEIL EXERCICES Documentation Php.net

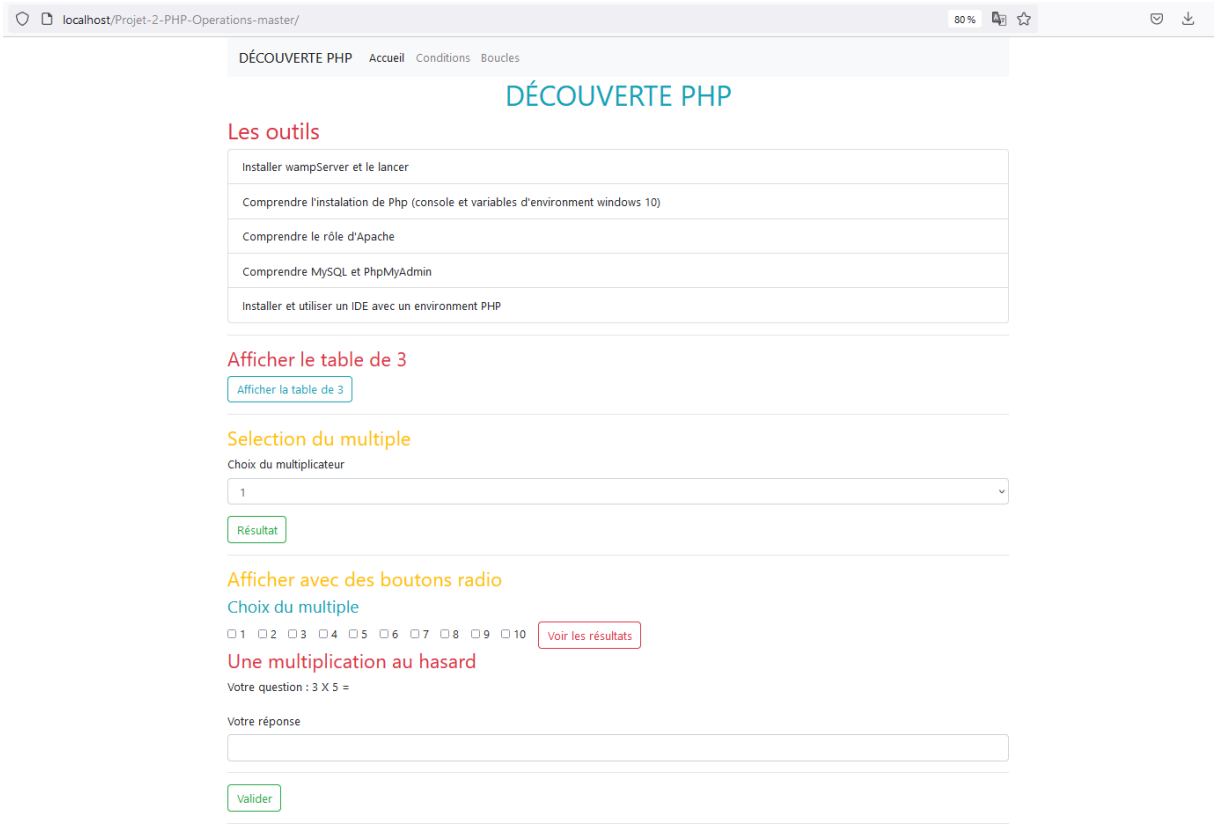
Passer des variables dans une URL

Dans cette exemple: on passe des variables dans un lien href
Dans une balise anchor href="getpost.php?nom=ALAIN&prenom=DELON"

Passer des variables dans url

Des données dans URL

LE PROJET :



1. Pour chaque etape : séparer l'affichage du traitement
2. Realisé un router simple à l'aide du passage de données dans l'URL avec la super globale $_GET[" "]
3. Comprendre la notion de Templating avec ob_start() et ob_getclean()

index.php

```php
<?php

ob_start();

if(isset($_GET['url'])){
    $url = $_GET['url'];
}else{
    $url = "home";
}


if($url == 'home'){
    require "views/home.php";
}elseif ($url == 'table3'){
    require "views/table3.php";
}elseif ($url == 'choixtable'){
    require "views/choixtable.php";
}elseif ($url == 'checkbox'){
    require "views/checkbox.php";
}elseif ($url == 'random'){
    require "views/random.php";
}

$content = ob_get_clean();

require "views/template.php";
```

4. https://www.php.net/manual/fr/function.ob-start.php
5. https://www.php.net/manual/fr/function.ob-get-clean.php

Pour les apprenants qui ont terminé la première partie :

- Super Mode révision : Poser une série de 5 questions puis afficher le score.
- L'ensemble du site est responsive.

Objectifs opérationnels :

- Savoir installer WampServer.
- Savoir réaliser une application PHP.

Objectifs d'apprentissage :

- Apprendre les bases du langage PHP.
- Comprendre le fonctionnement d'une architecture client-serveur.
- Apprendre à installer et à utiliser un serveur de test local.
- La pratique de quelques concepts algorithmiques.

Thèmes :

- PHP.
- WampServer.
- GitHub.
- Algorithmique.

Ressources :

Modules Onlineformapro :

- PHP

Vidéos :

- https://www.youtube.com/watch?v=4Rdh3GSVlKQ&list=PLxhnp_qsD8EN0rIZUvNFVHb8FQhl2d10B&ab_channel=PhilippeKoumi

Cours web :

- https://g-rossolini.developpez.com/tutoriels/php/cours/
- https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/4237646-decouvrez-le-fonctionnement-dun-site-ecrit-en-php
- https://www.pierre-giraud.com/php-mysql-apprendre-coder-cours/
- https://grafikart.fr/formations/php

Sites web :

- https://www.php.net/manual/en/index.php
- https://www.w3schools.com/php/
- https://www.wampserver.com/