Installation d’un serveur local

**Avant de commencer il faut arrêter le serveur mySQL avec l’application services, pour éviter la confusion des ports.**

|  |
| --- |

1-Visualiser la vidéo ci-dessous pour installer le serveur XAMPP si il n’est pas installé.

<https://www.youtube.com/watch?v=-f8N4FEQWyY>

2-Définit le port utilisé par votre serveur XAMPP.

|  |
| --- |

3- Activer votre serveur ainsi que votre base de données mySQL

|  |
| --- |

4- Trouver l’adresse de l’url qui vous affiche la page d'accueil PHPmyAdmin.

|  |
| --- |

5-Chercher l’emplacement du fichier **htdoc** sur votre ordinateur.

|  |
| --- |

6- Mettre votre le site suivant: <https://github.com/Amineham-prog/Burger-HAM>

sur le dossier htdoc

7-Tapez l’url du serveur.<http://localhost/>

8- Placer ce code entre la ligne **124 et 220 du fichier index.html**

| <?php  $images = array("burger2.png", "burger3.png", "burger4.png", "burger2.png", "burger3.png", "burger4.png","burger2.png", "burger3.png", "burger4.png");  ?>  <?php  foreach($images as $image){  echo '  <div class="col-12 col-sm-6 col-md-4">  <!-----------CARD-------------->  <div class="card mycard">  <img class="card-img-top photo" src="./img/'. $image .'" alt="Card image cap">  <div class="card-body bodycard">  <h5 class="card-title">$name</h5>  <p class="card-text contentcard">Some quick example text to build on the card title.</p>  </div>  <div class="card-body bodycard">  <!-------Button ------------------------------------>  <button type="button" class="btn btn btn-success" data-toggle="modal" data-target="#loginModal">  <span class="fa fa-sign-in"></span> Voir </button>  <!----------------------------------------------->  </div>  </div>  <!----------END CARD ------------------>  </div>    ';  }  ?> |
| --- |

9- Renommer votre index.htm par index.php et rafraîchir l’url de votre serveur local.

9-Expliquer l'intérêt et le fonctionnement de ce code.

|  |
| --- |

Syntaxe du langage PHP

## 

## **Exercice 1 Les constantes**

Vous serez parfois amené à utiliser de manière répétitive des informations devant rester constantes dans toutes les pages d’un même site. Il peut s’agir de texte ou de nombres qui reviennent souvent. Pour ne pas risquer l’écrasement accidentel de ces valeurs, qui pourrait se produire si elles étaient contenues dans des variables, vous avez tout intérêt à les enregistrer sous forme de constantes personnalisées. PHP dispose d’un ensemble de constantes prédéfinies utilisables dans tous les scripts.

Découvrir la version PHP sur votre serveur.

| Version PHP |
| --- |
| <?php  phpinfo();  ?> |

| **Exercice 1.php** |
| --- |
| <?php  define("PI",3.1415926535,TRUE);  define("site","http://www.google.com",FALSE);  echo "La constante PI vaut ",PI,"<br />";  echo "La constante PI vaut ",pi,"<br />";  if (defined( "PI")) echo "La constante PI est déjà définie","<br />";  if (defined( "pi")) echo "La constante pi est déjà définie","<br />";  if(defined( "site")) echo  {  echo "<a href=\" " ,site, " \">Lien vers mon site </ a><br>";  echo "<a href=\" " ,SITE, " \">Lien vers mon site </ a>";  }  ?> |

A) Dans un premier temps relever les deux déclarations et déterminer le rôle des deux premiers arguments :

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

|  |
| --- |

B) Déterminer la raison pour laquelle l’on peut faire appel à la constante PI en majuscule et minuscule à contrario de Site :

|  |
| --- |

**Comme pour les variables le php possède un nombre impressionnant de constante prédéfini que nous découvrirons en fonction du besoin.**

**Exercice 2 Type de variable (RAPPEL)**

Avant de manipuler des variables, en utilisant, par exemple, des opérateurs, il peut être utile de connaître leur type. Cela permet de s’assurer que le résultat obtenu est conforme à ce qui est attendu et qu’il n’y a pas d’incompatibilité entre les types de ces variables. L’opérateur d’incrémentation appliqué à une chaîne, par exemple, peut donner des résultats curieux.

La principale fonction permettant de déterminer le type d’une valeur est gettype(), dont la syntaxe est la suivante :

string gettype($mavar)

Elle retourne une chaîne de caractères contenant le type de la variable en clair. Les fonctions suivantes permettent de vérifier si une variable est d’un type précis :

Grâce à vaut recherche déterminer la définition de chacune de ces variables :

| Fonction | Définition : |
| --- | --- |
| is\_integer($var) ou is\_int($var) |  |
| is\_double($var) |  |
| is\_string($var) |  |
| is\_bool($var) |  |
| is\_object($var) | Une variable objet contient un pointeur vers des données stockées ailleurs |
| is\_resource($var) | Un exemple courant d'utilisation du type de données de ressource est un appel à une base de données. |
| is\_null($var) | Null est un type de données spécial qui ne peut avoir qu'une seule valeur: NULL. |
| is\_array($var) |  |

Ces fonctions retournent la valeur booléenne TRUE si la variable est du type recherché et FALSE dans le cas contraire. Nous pouvons donc les utiliser comme condition.

Tester et décrivez le fonctionnement du code suivant :

| **Exercice2.php** |
| --- |
| <?php  $var = 73;  if(is\_int($var))  {  $var++;  echo "La variable vaut $var <br />";  }  $var2="Bonjour ";  if(is\_string($var2))  {  $var2.=" à tous!";  echo $var2;  }  ?> |

Description du fonctionnement :

|  |
| --- |

Supprimer le point de *$var2.=" à tous!";* que ce passe-t-il et comment appel-t-on cette action en informatique ?

|  |
| --- |

## **Exercice 3 Changement de type.**

Lorsque l’on crée une variable, nous avons bien compris que c’est le contenu qui détermine le type. Mais parfois cela peut s’annoncer contraignant dans des calculs.

Debugger et tester le code suivant et déterminer son fonctionnement

| **Exercice3.php** |
| --- |
| <?php  $var="3.52 kilomètres";  $var2 = (double) $var;  echo "\$var2= ",$var2,"<br />";  $var3 = (integer) $var2;  echo "\$var3= ",$var3,"<br />";  $var4 = (boolean) $var3;  echo "\$var4= ",$var4,"<br />";  ?> |

Une autre méthode existe-elle retourne la valeur TRUE si l’opération est réalisée et FALSE dans le cas contraire. Cela ajoute une possibilité de vérifier si la conversion à bien été effectué.

**settype($var,"type \_désiré") ;**

## **Exercice 4 Contrôler l’état d’une variable**

Les fonctions isset() et empty() permettent de contrôler la présence d’une variable donc la valeur est explicité.

La fonction isset(), dont la syntaxe est la suivante :

isset($var)

Retourne la valeur FALSE si la variable $var n’est pas initialisée ou est à la valeur NULL.

Retourne la valeur TRUE si elle a une valeur quelconque.

La fonction empty(), dont la syntaxe est la suivante :

empty($var)

Retourne la valeur TRUE si la variable $var n’est pas initialisée, a la valeur 0 ou NULL ou la chaîne "0", et la valeur FALSE si elle a une quelconque autre valeur.

On souhait affiché vérifier l’existence de $a et afficher sa valeur si elle est non null,

Compléter le code suivant en utilisant vaut connaissant en développement et testez les valeurs NULL,0 et 1 pour $a :

| **Exercice 4.php** |
| --- |
| <?php  $a=5;  if(…………………){  echo "\$a existe déjà<br />";}    else {echo "\$a n'existe pas<br />";}    if(………………..){  echo "\$a est vide <br />";}    else {  echo "\$a a la valeur $a<br />";}  //Affiche "$a n'existe pas" et "$a est vide"  ?> |

Donner un exemple d’utilisation dans le domaine courant du web :

|  |
| --- |