PHP – Formation Ultime

CONCATENER DES VARIABLES :

$direBonjour = ‘Hello ‘  
$destination = ‘World’  
🡪 echo $direBonjour . ‘ ‘ . $destination // On associe les variable avec des points.

LES TABLEAUX :

Permet de stocker des données.

$identitePersonne1 = array (  
 ‘id ‘ => 4, // l’ID est la key = valeur unique pour communiquer avec une BDD  
 ‘prenom’ => ‘Nicolas’,  
 ‘nom’ => Dupont,  
 ‘age’ => 20  
) ;  
🡪 Bonjour $identite[‘prenom’] ! donnera Bonjour Nicolas  
Le tableau permet de ne pas refaire une requête auprès de la BDD pour les infos. Si on a pas nos clés dans le couple ‘clé’=>’valeur’ alors le tableau crée des index en commençant à 0.

LES CONDITIONS TERNAIRES :

(CONDITION) ? succes : false ;   
echo ($number > 10) ? ‘true’ : ‘false’ ; // on utilises les conditions ternaires principalement pour de l’affectation.

LES SWITCH : Lorsque l’on veut tester des égalités. Ici on pourrait avoir un message personnalisé pour chacun des notes de 0 à 20. On n’oublie pas de break pour sorti des accolades et d’avoir un default.

Switch ($notes){

case 0 : echo ‘Vous êtes naze’ ;  
 break ;  
 case 1 : echo ‘vous êtes très nul’ ;  
 break ;  
 default : echo ‘Vous êtes magnifiquement claqué au sol’ ;

}

LES CONDITIONS SIMPLES : pour seulement une ligne de code si vrai et une si fausse. On peut se passer des accolades.   
 $number =10 ;   
 if($number ==10)  
 echo ‘true’ ;  
else  
 echo ‘false’ ;

LES BOUCLES : while = tant que condition est vraie on fait le code, donc on incrémente dans le code

for ($i=0 ; $i<10 ; $i++) = pour i de 0 à 10 fait le code 🡪 Nombre précis de fois ou il faut faire les instructions. .

foreach($user as $name) {echo $name ;} 🡪 pratique pour les tableaux ! Surtout si on a des index personalisé !

do while = Faire ceci puis tant que ça c’est vrai continuer. Mais on y rentre une fois dans tous les cas.   
do { le code } while (condition) ;

On peut imbriqué des boucles c’est-à-dire faire une boucle dans une boucle.

**PS : Quand on code que du PHP on ne met pas de balise fermante.**

LES FONCTIONS : convention PHP, on écrit la fonction avec une majuscule :   
ex : function HelloWorld($name) {   
 echo ‘hey ’ . $name . ‘<br>’ ;  
 } ;

Pour retourner une valeur il faut avoir un return. Si on return $temp dans la fonction Formule, on est pas obligé d’appeler $temp qqs part. si on fait $résultat = Formule(52,74) ; le fait de retourner $temps nous permet de lire la variable résultat.

QQS Fonctions pour les strings :

* strlen($variables) 🡪 nombre de caractères dans la string
* str\_replace(‘mot à remplacer’, ‘mot qui remplace’, $variables) 🡪 remplace un mot dans une string
* strtolower($variables) 🡪 mettre en minuscule
* strtoupper($variables) 🡪 mettre en maj
* substr($variables , n°du premier caractères à garder, n° du dernier caractères à garder) 🡪 couper une partie d’une string
* <https://www.php.net/manual/fr/ref.strings.php> 🡪 Pour en avoir plus.

QQS Fonctions pour les Maths :

* abs($nombre) 🡪 donne la valeur absolue de la variable.
* max($notes ou suite de nombre) 🡪 valeur maximale de la suite de nombre   
  existe avec min() ;
* rand() ; génère un nombre entier au hasard, rand(0,1) ; génère un nombre entier entre 0 et 10 au hasard.
* round($nombre, rien ou nombre de chiffres après la virgule à garder) ; 🡪 arrondis la valeur de $nombres
* <https://www.php.net/manual/fr/book.math.php> 🡪 Pour en avoir plus

QQS Fonctions pour les tableaux :

* array\_flip($array) ; 🡪 Inverse clé et valeur dans un tableaux
* if(array\_key\_exists(clé, $array) ){echo ‘yes’} ; 🡪 Savoir si la clé (index) existe
* count($array) 🡪 compte le nombre d’items
* sort($array) ; 🡪 ordre de tri croissan

QQS Fonction obligatoire à plus de deux pages :

* include(« chemindufichier/fichier.php ») ; \_once 🡪 l’affiche que une fois sur la page
* require(« idem ») ; 🡪 Si fichier existe pas ça ne charge pas la page \_once 🡪 l’affiche que une fois sur la page

FORMULAIRE : Principe 1 : Récupérer les infos entrées par l’utilisateurs.

* On peut mettre le form dans un <table></table> pour que ce soit plus propre.

**ATTENTION AUX DONNEES !!! 🡪 On doit toujours utiliser des variables protégées !   
1) Rentrer nos infos dans des variables : $prenom = $\_POST[‘prenom’] ;  
2) Utiliser htmlspecialchars(REQUETE) ; 🡪 Capture le code html comme du texte  
3) + créer une fonction vérifier : que caractère alpha ou nombre de caractères etc**

ENVOYER UN FICHIER VIA UN FORMULAIRE :

1. Créer un dossier upload
2. enctype="multipart/form-data" 🡪 dans la balise form
3. APPELER L’IMAGE :

$\_FILES['image'] --> Ref à l'input image.

$\_FILES['image']['name'] --> renvoie le nom de l'image

['type'] --> renvoie le type (.png, .jpeg . docx etc)

['tmp\_name'] -->emplacementtemporaire du fichier

['error'] --> Y a t-il eu une erreur ? si == 0 alors par d’erreurs

1. 1mo = 1 000 000 000 octets par défaut php = au max 8mo. Et on parle en octet en php pour préciser la taille.
2. Donc on vérifie la taille
3. On choppe le chemin : $informationImage = pathinfo($\_FILES[‘image’][‘name’]) ;
4. On prends l’extensions : $extensionImage = $informationsImage[‘extension’] ;
5. On donne les extensions ok : $extensionsArray= array(‘png’,’gif’,’jpg’,jpeg’) ;
6. If(in\_array($extensionImage, $extensionsArray) ) {  
    move\_uploaded\_file($\_FILES[‘image’][‘tmp\_name’], ‘NomDuFichierUpload/’.time().basename($\_FILES[‘image’[‘name’])) ; plutôt que .basename on va utiliser rand(). ‘ . ‘ . $extensionImage) ;  
    🡪 Ici on accepte l’envoie car l’extension est bonne, puis on renome le fichier avec la fonction time() qui donne une date à la seconde près ce qui personalise vraiment le nom de fichier.

**Dans un input de type fichier : on met l’attribut required qui oblige l’utilisateur à remplir le champ.**

# LES BASES DE DONNES :

Une BDD c’est plein de tables différentes. mySQL permet les actions sur les BDD. PHP donne un ordre à mySQL qui applique cette action sur la BDD.   
Pour plus de sécurité on a une base de données par service (ex : une pour les mails, une pour l’authentification etc… on n’a pas une seule BDD avec tout dessus).

**PHPMYADMIN :**  On peut modifier du code par du PHP ou nous même dans la BDD grâce à PHPMYADMIN.

1. On crée un table avec le nombre de colonne nécéssaire
2. On écrit leur nom, le type, valeur max, valeur par défaut (tel que défini ou NULL = pas d’erreur ou rien). Interclassement = type et index : UNIQUE OU PRIMAIRE sinon on utilise pas les index
3. AI = Auto Incrémentation = associé avec un index unique pour l’id.
4. Quand on rentre un nouveau user ou autre pas besoin d’enregistre d’ID puisqu’il est AI.

EXPORTER 🡪 Pour envoyer nos tables sur un serveurs en ligne.

Lorsque l’on souhaite utiliser des requêtes PHP et une BDD il faut s’y connecter (Une seule fois suffit)

* mysql\_ 🡪 Trop vieilles on utilise plus car trop d’erreurs.
* mysqli\_ 🡪 Tjrs du sql mais amélioré on utilise de tps en tps.
* PDO 🡪 utile pour tout type de BDD (pas que mySQL) et surtout très sécurisé.

Dans php.ini : activer extension php\_pdo\_mysql.dll et display\_errors = On