

PHP

Tableaux

$\$tab = [\overset{\text{clé}}{\text{'nom'}} \Rightarrow \overset{\text{valeur}}{\text{'bond'}}, \text{'prenom'} \Rightarrow \text{'James'}];$

$\$tab2 = array(\text{'nom'} \Rightarrow \text{'albensom'}, \text{'prenom'} \Rightarrow \text{'jeff'});$

ajout de case

$\$tab[\text{'a'}] = \text{'b'};$

ajout d'élément (en fin de tableau)

$\$tab[] = \text{'c'}; // \text{clé établie selon indexation} //$

boucles

foreach

$foreach (\$tab \text{ as } \$i) \{$
 $\quad \uparrow \quad \uparrow$
 $\quad \text{tableau} \quad \text{valeur}$

$foreach (\$tab \text{ as } \$cle \Rightarrow \$val) \{$
 $\quad \uparrow \quad \uparrow \quad \uparrow$
 $\quad \text{tableau} \quad \text{clé} \quad \text{valeur}$

for

$for (\$i = 0; \$i < count(\$tab); \$i++) \{$
 $\quad \uparrow$
 $\quad \text{taille du tableau}$

`Var_dump()` : affiche les informations de la variable en paramètre (type, contenu, ...)

`require "dossier/fichier.php"` : inclut et exécute le fichier

`require_once "dossier/fichier.php"` : comme `require` mais si code déjà inclus, il ne le sera pas une 2nd fois

`require` et `require_once` renvoient erreur si fichier inexistant

`include` et `include_once` ont même effet mais Warning si fichier inexistant

Fonctions et attributs statiques

déclaration

```
public static function nomfonction() {
```

```
public static $instance;
```

Appel

```
NomClasse::nomfonction()
```

```
Static::$instance
```

Fonctions et attributs non statiques

Appel dans la classe

```
$this->nomfonction()
```

```
$this->$instance
```


query (String requete): Envoie la réponse de la requête sous une forme non lisible

fetch(): Renvoie la réponse de la requête dans un format lisible

attribut

? : Spécifie que l'attribut peut être null

mixed: Spécifie que l'attribut peut être de n'importe quel type

Modèle, Vue, Contrôleur

Le **modèle** est chargé de la gestion des données dont les interactions avec la BD

Les **vues** sont des fichiers contenant presque seulement du HTML, elles ne font pas de calcul

Le **contrôleur** est l'intermédiaire entre vue et modèle, il prend les décisions adéquates

actions du contrôleur: **Create**, **Read**, **Update**, **Delete** (CRUD)

\$_GET['action'] récupère l'action donnée par l'utilisateur dans l'URL avec **\$action**

__DIR__ retourne le dossier du fichier si utilisé dans une inclusion

namespaces

Les **namespaces** regroupent toutes les classes déclarées dans le fichier dans l'espace de nom

Cookies & Sessions

Les cookies sont des informations stockées sur l'ordinateur du client à l'initiative du serveur

on dépose un cookie avec **setcookie**("nomCookie", "valeurCookie", ^{expire dans 1h}time()+3600);

\$_COOKIE récupère la valeur du cookie placé en paramètre

serialize() transforme une variable en chaîne de caractère, inverse de **unserialize()**

effacer un cookie

unset(\$_COOKIE["cook"]); supprime la valeur du cookie

setcookie("cook", "", time()-1) supprime le cookie

htmlspecialchars() évite que du code HTML puisse être interprété sur une variable PHP

rawurlencode() permet d'échapper les variables PHP qui interviennent dans les URLs

Sessions

Les **Sessions** utilisent les cookies pour stocker des informations sur le serveur. Elles identifient les clients pour que le serveur stocke des informations sur chacun d'eux

initialiser

session_start() démarre une nouvelle session ou reprend une session existante

définir

\$_SESSION ['login'] = 'bob';

Suppression

Variable de session

unset(\$_SESSION ['login']);

Session

session_unset(); // vide \$_SESSION //

session_destroy(); // supprime fichier associé

Cookie::supprimer(session_name());

`headen("Location:". $url);` effectue une redirection sur l'URL
`exit();` (optionnel), termine l'exécution du script courant

Mot de passe

Rainbow Table est une structure de données permettant de retrouver des mots de passe avec un bon rapport Stockage/temps

hachage

`hash()` est une fonction qui permet de crypter un mot de passe

Sel

Le sel est une chaîne aléatoire ajoutée au début de chaque mot de passe avant hachage

Poivre

Le poivre est une chaîne aléatoire ajoutée dans le hachage du mot de passe. Il n'est pas stocké dans la BD pour des raisons de sécurité