

Base Développement Web

PHP / HTML

Introductions aux Bases de Données
Nathanaël Martel

PHP : les bases

- **Langage de programmation orienté objet**
 - **Exécuté côté serveur (derrière un serveur web comme apache) ou en ligne de commandes**
 - **Non-compilé (script)**
 - **Non-typé (ou très peu)**
 - **Syntaxe proche du c, c++ et java**
- ce qui en fait un langage très facile à apprendre tout en pouvant être très puissant**

PHP : fonctions de base

- `// commentaire fin de ligne`
- `/* commentaire sur plusieurs lignes */`
- `$foo = 12 ; $bar = 'foo bar' ; // affectation de variable.`
- `echo $foo ; // affiche la valeur de la variable`
- `var_dump($foo) ; // affiche la variable $foo avec son type (très utile pour le débogage)`
- `$t = $foo.$bar ; // concaténation de chaîne`
- `if($foo) {} elseif () else {} // conditions`
- `for($i=0;$i<10;$i++) {}`
- `$foo = array('orange', 'pomme', 'banane') ; // tableau de valeurs`
- `$foo = array(0 => 'orange', 2 => 'banane', 1 => 'pomme', 'arg' => 'kiwi') ; // tableau associatif`
- `$foo[0] ; $foo['arg'] ; // accès à une valeur d'un tableau`
- `$foo[] = $bar; $foo['bar'] = 'string' ; // ajout d'élément dans un tableau`
- `count($foo) ; // nombre d'élément du tableau $foo`
- `foreach ($foo as $key => $value) {}`
- `While() {}`
- `$Object = new Class; $Object = new Class($args); // instantiation d'objet`
- `$Object->getSomething() ; // appel d'une méthode de la classe sur l'objet`
- ... RTFM php.net

PHP : connexion à la base

- **Utiliser un « connecteur » avec la base de données :**
 - Une extension qui gère la connexion
 - Trois entre PHP et Mysql :
 - Mysql (obsolète)
 - PDO

```
$pdo = new PDO('mysql:host=example.com;dbname=database',  
    'user', 'password');
```
 - Mysqli

```
$mysqli = new mysqli("example.com", "user", "password",  
    "database");
```

PHP : générer des requêtes SQL

- Du point de vue de PHP,
la requête est une chaîne de caractère
- Il va donc falloir fabriquer des chaînes de caractère
 - Concaténation
`$query = 'SELECT * FROM articles WHERE id=' . $id ;`
 - Sprintf
`$query = sprintf('SELECT * FROM articles WHER id=%s', $id) ;`

PHP : traiter les données

- **Nous exécutons la requête :**
 - `$results = $pdo->query('SELECT * FROM articles') ;`
- **Et nous récupérons un jeu de données que l'on traite comme un tableau associatif**
 - ```
foreach ($results as $row) {
 echo $row['title'];
}
```
  - ```
<ul>  
    <?php foreach ($results as $row) : ?>  
        <li><?php echo $row['title'] ?></li>  
    <?php endforeach ;?>  
</ul>
```

PHP : traiter les données

- **La méthode query() est une version courte pour :**

```
- $statement = $pdo->prepare('SELECT * FROM  
articles');  
$statement->execute();  
$articles = $statement->fetchAll();  
// $results est un array contenant tous les  
résultats de la requête  
var_dump($articles) ;  
echo count($articles) ;  
foreach ($articles as $article) {  
    echo $article['title'];  
}
```

PHP : traiter les données

- **Nous exécutons la requête :**
 - `$nb = $pdo->exec('INSERT INTO articles (title) VALUES ("mon merveilleux titre")') ;`
 - `$nb = $pdo->exec('UPDATE articles SET title="mon merveilleux titre"') ;`
- **Et nous récupérons le nombre d'enregistrements affectés par la requête**
 - `echo $nb ;`

PHP : traiter les données

Codes pour tester :

<https://gist.github.com/nathanaelmartel/da3859bc2a93691ace4eb6cd59b63c64>

HTML : langage de markup

- **HTML = XML particulier**

- `<html>`
 `<body id="page-12" class="simple-page">`
 `<h1>Le titre !</h1>`
 `</body>`
 `</html>`

- **Avec des balises dédiées**

- **Et des appels à des ressources externes**

- CSS, pour gérer la présentation
 - Javascript, pour mettre en place de la programmation côté client
 - Images
 - Vidéo

HTML : langage de markup

- **HTML = XML particulier**

```
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Le titre de l'onglet</title>
  <link rel="stylesheet" href="/assets/css/bootstrap.min.css" type="text/css" />
  <script src="/assets/js/bootstrap.min.js"></script>
</head>
<body>
  <header>
    <p class="big-title">Le titre de la page</p>
    <nav>
      <ul>
        <li><a href="https://google.com">Google</a></li>
        <li><a href="/une-autre-page">page 2</a></li>
        <li><a href="encore-une">page 3</a></li>
      </ul>
    </nav>
  </header>
```

HTML : langage de markup

```
<section>
  <article>
    <header>
      <h1>Le titre de l'article</h1>
    </header>
    <div class="content">
      
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
      <p>Quis, asperiores. Placeat facere nam vero odio, voluptatum quia dolore eligendi ex omnis ducimus.</p>
      <p>Totam minus est magni quasi, sed reiciendis.</p>
      <p>Nemo, neque nam corrupti repellat dolorum sed hic voluptas ipsum?</p>
    </div>
    <footer>
      <p>Écrit le 14 mars 2017 par Nathanaël Martel</p>
    </footer>
  </article>
</section>
<footer>
  <p>copyright Nathanaël Martel</p>
</footer>
</body>
</html>
```

PHP : générer une page HTML

- Avec PHP on injecte des valeurs dynamiques

- ```
<html>
 <body id="page-<?php echo $id ?>" class="<?php
 echo $template_name ?>">
 <h1><?php echo $title ?></h1>
 </body>
</html>
```

- Génère le HTML suivant

- ```
<html>  
    <body id="page-12" class="simple-page">  
        <h1>Le titre !</h1>  
    </body>  
</html>
```

PHP : générer une page HTML

- **Conditions**

```
<html>
  <body class="<?php echo ($template_name != ' ')?
$template_name:'default'; ?>">
    <?php if ($title == ' '): ?>
      <h1>Titre vide</h1>
    <?php else: ?>
      <h1><?php echo $title ?></h1>
    <?php endif ?>
  </body>
</html>
```

PHP : générer une page HTML

- **Boucle**

```
<?php $foo = array(0 => 'orange', 2 => 'banane', 1 => 'pomme', 'arg' => 'kiwi') ;?>
```

```
<html>
  <body>
    <ul>
      <?php foreach($foo as $key => $value): ?>
        <li>
          <strong><?php echo $key ?></strong>
          <?php echo $value ?>
        </li>
      <?php endforeach ?>
    </ul>
  </body>
</html>
```

PHP : récupérer des données

Requête HTTP de type GET

- Par exemples une URL du type
 - `http://monserveur.com/mapage.php?param=valeur&foo=bar`
- Il est possible d'accéder à ces paramètres avec la variable PHP : `$_GET`
 - Dans l'exemple : `$_GET['param']='valeur'`
 - C'est un tableau
 - Ça porté est global

PHP : récupérer des données

Formulaire HTML

```
<form action="" method="POST">  
  <label for="nom">Nom</label>  
  <input type="text" class="form-control" id="nom" name="nom"  
required />  
  <label for="commentaire">Commentaire</label>  
  <textarea id="commentaire" name="commentaire" class="form-  
control" rows="5" required ></textarea>  
  <input type="submit" class="btn btn-primary" value="Envoyer">  
</form>
```

PHP : récupérer des données

Requête HTTP de type POST

- Ce sont typiquement les valeurs d'un formulaire HTML
- Il est possible d'accéder à ces paramètres avec la variable PHP : `$_POST`
 - Dans l'exemple : `$_POST['nom']`, `$_POST['commentaire']`
 - C'est un tableau
 - Ça porté est global

Conclusion

PHP est un langage de programmation sophistiqué qui permet de générer des pages HTML et d'accéder à la base de données