

Module 2 - Rappels HTML/Javascript et introduction au PHP

Introduction

Quelques rappels sur les choix effectués lors des cours précédents et restant d'actualité :

- Navigateur Firefox (<https://www.mozilla.org/en-GB/firefox/>) : par défaut, on travaillera avec le navigateur Firefox. Malgré les efforts du W3C, il reste de nombreuses différences techniques entre les navigateurs existant. Ces différences techniques étant non seulement sources de pertes de temps incroyables mais aussi assez bien détaillées sur le web, on choisit de les ignorer dans le cadre des cours orientés web.

Naturellement, dans le cadre d'un travail professionnel ou d'un stage, vous devez vous assurer que vos productions web fonctionnent sur un éventail assez large de navigateurs !

- Utilisation de Notepad++ (<http://notepad-plus-plus.org/>) pour l'édition des documents. Il existe plusieurs IDE et éditeurs riches pour les langages du web. Dans le cadre de ce cours, on fait le choix d'utiliser simplement Notepad++, qui facilite l'édition par l'ajout automatique de couleurs mais n'apporte pas d'aide supplémentaire. Il est donc important d'être rigoureux dans l'écriture de son code, tant au niveau de la forme (clean code, indentation...) que du fond (bien connaître la syntaxe du langage) !
- Choix des matières abordées. L'informatique orientée web est un domaine extrêmement vaste, où de nouvelles extensions, de nouveaux protocoles et de nouveaux outils apparaissent chaque jour. Il est impossible d'en faire le tour intégralement. Aussi, dans ce cours, on mettra surtout l'accent sur
 - les éléments de base nécessaires pour comprendre la structure d'une application web ;
 - les aspects difficiles (particularités des langages), qui sont généralement peu documentés ou mal expliqués sur le web et qu'il est difficile de maîtriser par soi-même.

On a donc choisi de laisser de côté les extensions et outils qui, par leur attrait visuel, sont suffisamment faciles à aborder seuls (hors cours) ou dont l'intérêt est spécifique à un certain type d'application.

Prise en main du serveur pour les laboratoires

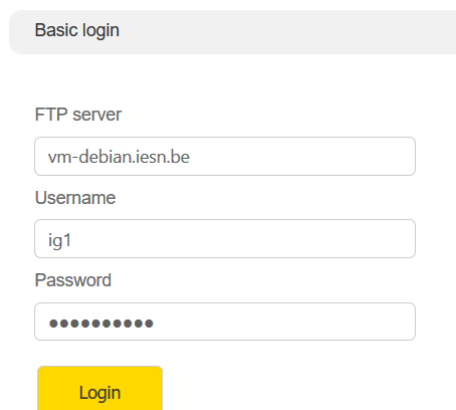
Une application web avec PHP nécessite l'installation des éléments suivants sur le serveur web :

1. Un serveur SQL. Dans le cadre du cours de DW, c'est **MySQL** qui a été choisi car il est libre et gratuit mais bien plus encore : complet, performant, etc.
2. Un serveur http. Un serveur http est un logiciel permettant d'interpréter les requêtes envoyées par l'internaute. **Apache**, le plus connu d'entre eux, est celui qui est utilisé sur le serveur web du cours de DW. Tout comme MySQL, celui-ci est gratuit et très présent sur les serveurs web. La version utilisée est **Apache/2.2.16 (Debian)**.
3. Un analyseur PHP. L'acronyme **PHP** est à la fois utilisé pour dénommer le langage de script et le moteur qui interprète ce langage sur le serveur Web. L'ensemble permet de créer des pages web dynamiques.

Afin de vous permettre de travailler sur un serveur distant (environnement de travail proche du monde professionnel) et non sur votre propre machine (en "localhost"), un serveur web est à votre disposition à l'adresse <http://vm-debian.iesn.be>. Un login "igXX" (XX ∈ [1..60]) et un mot de passe donnent accès à un compte individuel sur ce serveur.

Afin de transférer les fichiers produits sur le serveur, il est nécessaire d'utiliser un client FTP (File Transfert Protocol). Diverses solutions existent telles que l'installation du client *Filezilla* (<https://filezilla-project.org>) sur votre machine ou l'utilisation d'une application web telle que *net2ftp* (<http://www.net2ftp.com>).

Dans le cas de *net2ftp*, il suffit de compléter le formulaire avec les informations correspondant à votre compte, comme indiqué dans la Figure 1.



The image shows a web form titled "Basic login". It contains four input fields: "FTP server" with the value "vm-debian.iesn.be", "Username" with the value "ig1", and "Password" which is masked with dots. Below the fields is a yellow "Login" button.

Figure 1 - net2ftp

Le répertoire auquel vous avez accès en FTP se nomme `\www`.

Cependant, pour des raisons de sécurité, la configuration du serveur est telle qu'un client HTML n'a accès qu'au répertoire `\www\public`. Ce répertoire est appelé *répertoire racine* de la partie web.

Pour visualiser (en tant qu'utilisateur) le résultat de votre travail, il faut entrer l'adresse du serveur, le login et enfin le nom de la page d'accueil du site ou de l'application (<http://vm-debian.iesn.be/ig1/hello.html> par exemple).

Pour information, le serveur étant configuré pour vous permettre de suivre le tutoriel sur le Zend Framework 1, l'adresse <http://vm-debian.iesn.be/ig1> mène à la page d'accueil d'une application existante, respectant l'architecture imposée par Zend.

Exercice 1 : prise en main du serveur

Objectifs

- manipuler les fichiers placés sur le serveur
- rappels sur les URL absolues et relatives
- rappels sur l'utilisation de HTML/CSS
- rappels sur Javascript
- premières manipulations en PHP

■ Étape 1 : création d'une page dynamique

Dans Notepad++, créez un nouveau fichier appelé `test.php` et contenant le code repris ci-dessous (il s'agit de l'exemple montré lors du cours théorique).

```
<html>
  <head>
  </head>
  <body>
    <h1>Bienvenue !</h1>
    <p>Il est actuellement
    <?php
      $heure = date('H:i');
      echo $heure;
    ?>
    et tout va très bien.</p>
  </body>
</html>
```

Sauvegardez le fichier en local.

Sur le serveur, dans le répertoire racine de la partie web, créez un sous-répertoire appelé Labo1 puis copiez-y le fichier `test.php`.

Ouvrez le navigateur Firefox et accédez à la page en question en entrant l'adresse suivante.

```
|| http://vm-debian.iesn.be/igXX/Labo1/test.php
```

■ Étape 2 : Une image de plus

Cherchez sur le web un logo du langage PHP (éléphant ou autre). Sauvegardez l'image en local. Sur le serveur, dans le répertoire racine de la partie web, créez un sous-répertoire appelé Images et placez-y le logo en question.

Modifiez la copie locale du fichier `test.php` (considérez-le comme un fichier HTML) pour ajouter le logo PHP après le paragraphe indiquant l'heure. Utilisez un style CSS inline pour faire en sorte que l'image soit centrée horizontalement. Téléversez le nouveau fichier et vérifiez le résultat sur le navigateur.

■ Étape 3 : Adresses relatives et absolues

Dans le fichier php, avez-vous utilisé une adresse relative ou une adresse absolue pour indiquer l'emplacement de l'image ?

Modifiez le fichier pour utiliser l'autre sorte d'adressage.

■ Étape 4 : Une dose de CSS

Rendez-vous sur le site « Lorem ipsum » (<http://fr.lipsum.com/>) et demandez-y la création de 5 paragraphes de texte (c'est l'option par défaut). Cliquez sur le bouton « Générer ». Copiez-collez les cinq paragraphes dans le fichier `test.php`, sans oublier d'encadrer chacun d'eux par des balises `<p>` et `</p>` !

Créez un nouveau fichier sous Notepad++, une feuille de style appelée `test.css`. Dans celle-ci, ajoutez les règles suivantes.

1. Le fond de la page (balise `body`) sera de couleur « olive ».
2. Le contenu de la page (ajoutez une balise `<div>` encadrant tout le contenu) fera 800 px de largeur, sera centré, aura un rembourrage de 6 px sur tous les côtés et sera affiché en blanc sur fond noir.
3. Le titre « Bienvenue » sera affiché en couleur jaune.
4. Le paragraphe affichant l'heure (et uniquement lui - utilisez un identificateur) sera placé dans un cadre de couleur olive dont les côtés feront 2 px d'épaisseur.
5. L'image ne sera plus centrée et sur une ligne à elle toute seule mais elle sera collée contre le bord droit du cadre noir et le texte s'écoulera autour d'elle (`float`) ; l'image sera séparé du bord (à droite) et du texte (en-dessous) par un rembourrage de 6 px.
6. Les paragraphes importés depuis le site « Lorem ipsum » (créez une classe pour les désigner) seront écrits en italique.
7. La première lettre de chacun des paragraphes importés depuis le site « Lorem ipsum » sera affiché en jaune et au double de la taille normale (200%).

■ Étape 5 : Un soupçon de Javascript

Pour l'instant, la page affiche uniquement l'heure du serveur, produite par PHP.

Éditez le document PHP pour que le paragraphe encadré présente un texte du type « Il est actuellement HH:mm sur le serveur et HH:mm:ss sur le navigateur et tout va bien. »

Pour ce faire, utilisez Javascript en vous aidant des conseils suivants :

- L'expression `new Date ()` donne un objet Javascript représentant le moment (date, heure) actuel ; l'objet en question est créé par la fonction constructrice `Date()`.
- (Rappel) Dans le corps (`body`) d'un document HTML, on peut incorporer à n'importe quel endroit un code Javascript grâce aux balises `<script>` et `</script>`. Si le code contient une instruction `document.write(...)`, l'argument de la fonction `document.write` est ajouté au texte HTML à l'endroit en question.

- Tous les objets créés par une fonction constructrice `fonc()` ont comme prototype l'objet dénoté par `fonc.prototype`. Si on tente d'utiliser une propriété (méthode ou champ) d'un objet et que cet objet n'en a pas, on recherche la propriété en question chez son prototype.
- L'objet `Date.prototype` possède une méthode `toLocaleTimeString()` qui convertit un objet date en une chaîne de caractères représentant l'heure associé à cet objet.

Note. Oui, on aurait pu vous dire tout ça de manière beaucoup plus simple ou vous donner directement le code à copier/coller... mais on suppose qu'après le cours d'IG2, vous êtes capables de comprendre chacune des phrases citées ci-dessus dans le détail !

■ Étape 6 : Un brin de PHP

En PHP, la fonction « `date` » permet de traduire une date (par défaut, on travaille sur le moment actuel) en une chaîne de caractères correspondant à un format indiqué.

Le code utilisé ci-dessus comporte deux lignes : la première stocke la chaîne de caractères correspondant au format *heures:minutes* dans la variable appelée `$heure` (les identificateurs de variables en PHP commencent quasiment tout le temps par un dollar) ; la seconde ligne « écrit » le contenu de la variable `$heure` dans le fichier.

```
|| $heure = date('H:i');
|| echo $heure;
```

Ainsi, après l'action du processeur PHP, tout le contenu des balises `<?php ... ?>` est remplacé par les sorties effectuées par le code, c'est-à-dire par l'heure.

À la fin de la phrase donnant l'heure (donc, après le « et tout va bien »), ajoutez ce qu'il faut pour afficher « et nous sommes le 21/09/2026 ». Pour obtenir cette écriture, utilisez le format « `j/m/Y` » (j pour le jour, m pour le mois avec éventuellement un 0 devant et Y pour l'année en 4 chiffres).

■ Étape 7 : Et la touche finale !

Après le paragraphe encadré, ajoutez un nouveau paragraphe qui affichera soit « C'est le début du mois. », soit « C'est la fin du mois. » selon que le numéro du jour est `< 15` ou pas.

Pour ce faire, complétez le code suivant.

```
|| <p>C'est le
|| <?php
||     ... = date('j');
||     if ($jour < 15)
||         echo 'le début';
||     else
||         ...
||     ?>
|| du mois.</p>
```

Vérifiez le résultat.

Puis, dans un second temps, supprimez les retours à la ligne avant et après le code PHP, de sorte que « le » soit collé à la balise `<?php` et que la balise `?>` soit collée à « du ». Observez le résultat.

Exercice 2 : Méta-HTML et Méta-méta-HTML

Objectifs

- rappels sur l'envoi de données via la méthode GET
- rappels sur Javascript et la programmation événementielle
- Générer du HTML avec Javascript
- Générer du HTML et du Javascript qui génère du HTML avec PHP

■ Étape 1 : Envoi par GET

Quand un internaute demande à visiter une page, le navigateur envoie une requête http via Internet. Cette requête contient, entre autres, l'url de la page désirée. De temps en temps, d'autres informations sont nécessaires. Ces informations peuvent être envoyées soit selon la méthode GET (dans l'url) soit selon la méthode POST (où il s'agit d'envoyer les données d'un formulaire HTML).

Du côté PHP, on peut réceptionner les données envoyées via GET en utilisant la variable portant le doux nom de `$_GET`, qui agit comme un tableau associatif (plus d'informations à ce sujet plus tard).

Pour l'instant, contentez-vous de créer un fichier `commande.php` contenant le code suivant et de le téléverser dans un répertoire sur le serveur.

```
<html>
  <head>
  </head>
  <body>

    <p>Votre commande :
      <?php
        if (isset($_GET) && isset($_GET['article']))
          echo $_GET['article'];
        else
          echo 'aucun article commandé';
      ?>
      !!!
    </p>

  </body>
</html>
```

Sans rentrer dans les détails, ce fichier php vérifie si des informations ont été envoyées via GET et plus particulièrement si une information est associée au nom « article ». Si c'est le cas, on affiche le nom de l'article. Dans le cas opposé, on affiche « aucun article commandé ». Dans les deux cas, on termine la phrase par trois points d'exclamation.

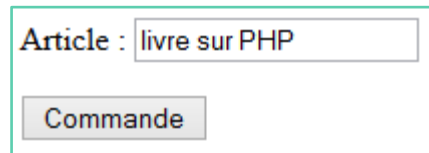
Testez le fonctionnement de ce fichier en ciblant les url suivantes (une par une) dans votre navigateur (ajoutez le domaine et les éventuels répertoires nécessaires).

```
...commande.php
...commande.php?article=montre
...commande.php?article=réveil
...commande.php?article=<i>tour de Pise</i>
```

```
|| ...commande.php?article=pasdexclamation</p>
```

■ Étape 2 : Du Javascript pour faciliter les tests

Créez maintenant un nouveau fichier `choix.html`, qui contiendra une page HTML sans code PHP (vous pouvez donc lui donner l'extension `.html`, même si l'extension `.php` fonctionnera tout aussi bien).



Cette page html comportera simplement un champ textuel (donnez-lui un identificateur) et un bouton intitulé « Commande ». Arrangez-vous pour que, lorsqu'on clique sur le bouton commande, on se rende sur la page `commande.php` avec, comme argument GET, `article` = ce qui est indiqué dans le champ textuel.

Pour rappel, la commande Javascript pour se rendre vers une autre url est

```
|| location.href=nouvelleURL;
```

Testez quelques cas (dont les cinq de l'étape précédente).

Ces tests mettent-ils en évidence quelques dangers potentiels ? Lesquels ?

■ Étape 3 : Générer du HTML avec Javascript

Reprenez le document `choix.html`. Dorénavant, pour simplifier le problème, on supposera que les articles sont repérés par des numéros (1, 2, 3 etc.). Au niveau HTML, la page sera vide. Par contre, elle contiendra un script Javascript qui sera exécuté dès la fin du chargement (c'est-à-dire lors de l'événement `window.onload`).

Ce script commencera par initialiser une variable `numMax` à 10 puis il ajoutera, dans le corps de la page, des boutons numérotés de 1 à `numMax`. Un clic sur l'un de ces boutons devra envoyer l'utilisateur vers la page `commande.php` avec, comme `article`, le numéro correspondant au bouton.

Note. Si tous vos boutons mènent à la commande d'un article numéroté « 11 », vous avez un problème relatif à la notion de closure !

Note. Si vous n'arrivez vraiment pas à trouver une solution qui fonctionne, référez-vous à la solution donnée à la fin de ce document.

■ Étape 4 : Mise en situation

Imaginez que les articles 1, 2, 3, ... puissent être achetés en dépensant des « points » gagnés en visitant un site. En fonction du nombre de points accumulés, on a un choix plus ou moins grand d'articles. Pour simplifier, disons que, si on dispose de `N` points, on peut choisir n'importe quel article entre l'article 1 et l'article `N`.

Transformez le fichier `.html` en un fichier `.php`.

Dans le code Javascript, et plus précisément dans l'affectation `numMax = 10`, remplacez le 10 par du code PHP. Normalement, le code en question irait rechercher dans une base de données le nombre de points dont l'internaute dispose. Ici, on va faire plus simple :

```
<?php
    $nbPoints = 15;
    echo $nbPoints;
?>
```

Mettez à jour le fichier sur le serveur et vérifiez que tout fonctionne !

■ Étape 5 : Générer du Javascript (qui génère de l'HTML) avec du PHP

On peut insérer du PHP à n'importe quel endroit dans une page HTML pour produire du code. Le code produit peut être du texte placé à l'intérieur de balises HTML ou encore du code HTML ou même encore du Javascript.

Le code produit n'est pas forcément un bout de code qui a du sens en lui-même : il peut s'agir d'une partie de texte, d'une partie de code HTML ou d'un bout d'instruction Javascript (attention toutefois à la clarté de la chose... le mieux est que le script PHP reste facilement lisible).

Dans le code Javascript qui génère les boutons, vous avez sans doute utilisé quelque part une boucle du style

```
for (var i = 1 ; i <= maxNum ; i++)
```

Si on voulait plutôt afficher les boutons dans l'ordre décroissant, on utiliserait plutôt

```
for (var i = maxNum ; i >= 1 ; i--)
```

Dans le document php, au sein du code qui complète l'affectation de `numMax` (ce code représente la lecture des informations dans la base de données), ajoutez l'initialisation de variable suivante.


```
$ordreCroissant = true;
```


À l'endroit où se trouve la boucle « for », ajoutez du code PHP pour que, en fonction de la valeur de la variable `$ordreCroissant`, on utilise l'une ou l'autre boucle. Pour écrire la conditionnelle en PHP, référez-vous à l'exercice précédent.

Pour faciliter les tests, vous pouvez initialiser la variable `$ordreCroissant` à partir des données reçues via GET lors de la demande de `choix.php` (basez-vous sur le code donné plus haut permettant de récupérer la valeur de l'argument « article »).

■ Étape 6 : un peu de réflexion ?

[1] On vous propose le choix entre les deux solutions suivantes. Laquelle pensez-vous être la meilleure ?

- a) le code PHP produit toute l'en-tête de la boucle, c'est-à-dire toute la « ligne »
`for(...);` 
- b) comme le début « `for (var i =` » et la fin « `)` » restent les mêmes dans les deux cas, le code PHP ne remplace que la partie qui change.

[2] Que pensez-vous de l'argument suivant : « Dans un projet normal, on devrait privilégier le clean code et la lisibilité... mais, ici, il s'agit de documents web. Plus le fichier est court et mieux c'est, car la transmission via Internet s'effectuera plus rapidement. » ? 

[3] « Dans la version précédente, quand les boutons à afficher étaient déterminés au niveau local [version sans PHP], il aurait fallu vérifier dans commande.php que l'internaute avait bien droit à cet achat. Par contre, dans la dernière version, vu que c'est le serveur qui détermine le nombre de boutons à afficher, cette vérification n'est plus utile. » Êtes-vous d'accord ou non ? Justifiez .

Solution exercice 2, étape 3

```
var numMax = 10;

function actionComande (num) {
    function act() {
        location.href = "commande.php?article=" + num;
    }
    return act;
}

function init () {
    for (var i = 1 ; i <= numMax ; i++) {
        var nvNoeud = document.createElement("button");
        nvNoeud.type = "button";
        nvNoeud.innerHTML = i;
        nvNoeud.onclick = actionComande(i);
        document.body.appendChild(nvNoeud);
    }
}

window.onload = init;
```

Note. On peut supprimer la définition de la fonction actionComande et remplacer la ligne définissant nvNoeud.onclick par la version condensée ci-dessous.

```
nvNoeud.onclick = (function (num) {
    var act = function () {
        location.href = "commande.php?article=" + j;
    };
    return act;
})(i);
```