Conception de documents & interfaces

Partie 5

Sébastien VIALLEMONTEIL
Développeur Web
sebastien.viallemonteil@agrosupdijon.fr

Les formulaires

Les formulaires sont utiles lorsque l'on souhaite transmettre des données à notre serveur web pour par exemple enregistrer un nouvel utilisateur en base de données, inscrire un personne à une newsletter, etc...

Il existe deux méthodes pour envoyer des données via formulaire

- **GET :** les données sont envoyées par le biais de l'url
- **POST :** les données sont envoyées directement au serveur web (non visible dans l'url)

Ex d'url avec des données *http://monsite.com/exemple.php?id=5&page=5*

Balise <form>...</form>

Les attributs :

- **action :** le script qui recevra les informations du formulaire (URL)
- **method :** la méthode HTTP utilisée pour envoyer les données (GET ou POST) (Défaut = GET)
- **enctype :** indique le format de données utilisé lors de l'envoi (non obligatoire)

Il existe d'autres attributs

La compositon générale d'un formulaire

- Des champs de type texte, nombre, mot de passe, etc. Des listes déroulantes, des cases à cocher, des boutons radio
- Un label (nom) associé à chaque champ qui en a besoin
- Un bouton pour envoyer le formulaire (voire un bouton pour remettre les champs du formulaire à zéro)

Les différents types de champ

Il existe beaucoup de types de champ **input**, tous ne sont pas compatibles avec tous les navigateurs. Il s'agit d'une balise orpheline

- **text :** type le plus classique pour taper une simple chaîne de caractères
- **password :** champ de type mot de passe, le texte est caché
- **email :** pour contenir une adresse email
- **number :** pour contenir des nombres entiers ou décimaux
- **tel :** pour contenir des numéros de téléphones (simple champ texte)

Les différents types de champ

- **date :** contient une date (certains navigateurs affichent un petit calendrier)
- **file :** pour envoyer des fichiers
- **button :** pour afficher un simple bouton
- **submit :** envoie les données du formulaires lors du clique
- **reset :** remet à zéro les champs du formulaire

Il existe beaucoup d'autres types

Les attributs des champs

- **name :** très important, c'est le nom qui est utilisé lors de la reception des données du côté serveur
- **value :** permet de donner une valeur par défaut pour les champs de texte, pour les boutons (button, reset, submit), cela définit le texte affiché dans le bouton
- **placeholder :** texte affiché dans les champs texte qui se cache lorsque l'on écrit dans un champ
- **required :** indique qu'il s'agit d'un champ obligatoire

Les attributs des champs

- **size :** indique la taille du champ en largeur, (px par défaut sauf pour champs texte => nb de caractères)
- **max-length :** indique le nombre de caractères que l'on peut taper au maximum
- **disabled :** indique que le champ est désactivé (grisé)
- **autocomplete :** active ou désactive la saisie automatique avec d'anciennes valeurs (on ou off, on par défaut)

Il existe beaucoup d'autres attributs

Les labels

Les labels permettent de contenir du texte (en général un nom) associé à un champ de formulaire.

Il est possible d'appliquer un comportement à ce label afin qu'il place le curseur dans son champ associé lorsque l'on clique dessus.

Pour cela on utilise l'attribut **for**

< HTML />

```
1 <label for="prenom">Prénom</label>
2 <input type="text" name="prenom" id="prenom" placeholder="Prénom"/>
```

Prénom Prénom

Les listes déroulantes

Il s'agit d'une liste pouvant avoir une ou plusieurs valeurs prédéfinies

< HTML />
```html
<select name="pays" id="pays">
    <option value="fr">France</option>
    <option value="de">Allemagne</option>
    <option value="es">Espagne</option>
    <option value="it">Italie</option>
</select>
```

Pays France

Les cases à cocher et boutons radio

Les cases à cocher et les boutons radio sont assez similaires au niveau de leur fonctionnement, ils permettent à l'utilisateur de faire des choix

On peut cocher plusieurs cases à cocher, en revanche pour les boutons radio, l'utilisateur n'en choisit qu'un (par groupe)

< HTML />
```
<label for="choix1">Choix 1</label>
<input type="checkbox" name="choix1" id="choix1">
<label for="choix2">Choix 2</label>
<input type="checkbox" name="choix2" id="choix2">
<label for="choix3">Choix 3</label>
<input type="checkbox" name="choix3" id="choix3">
```

< HTML />
```
<label for="btn_radio1">Choix 1</label>
<input type="radio" name="btn_radio" id="btn_radio1">
<label for="btn_radio2">Choix 2</label>
<input type="radio" name="btn_radio" id="btn_radio2">
```

Les cases à cocher et boutons radio

Les cases à cocher

Choix 1 Choix 2 Choix 3

Les boutons radio

Choix 1 Choix 2

Les zones de texte

Il est possible de créer des zones de texte qui sont plus grandes qu'un simple champ texte, cela permet par exemple de taper des descriptions

On utilise la balise **textarea**, on peut définir la taille en CSS ou directement en HTML avec **rows** et **cols**

< HTML />

```
1 <label for="desc">Description</label>
2 <textarea name="description" id="desc" rows="10" cols="30"></textarea>
```

Description

Il est possible de désactiver le redimensionnement en CSS avec **resize: none**

Exemple de formulaire

< HTML />

```html
<form action="traitement.php" method="POST">
    <p>
        <label for="nom">Nom</label>
        <input type="text" name="nom" id="nom">
    </p>
    <p>
        <label for="prenom">Prénom</label>
        <input type="text" name="prenom" id="prenom">
    </p>
    <p>
        <label for="email">Email</label>
        <input type="email" name="email" id="email">
    </p>
```

Exemple de formulaire

```html
    <p>
        <label for="password">Mot de passe</label>
        <input type="password" name="password" id="password">
    </p>
    <p>
        <label for="newsletter">Newsletter</label>
        <input type="checkbox" name="newsletter" id="newsletter">
    </p>
    <p>
        <input type="submit" value="Envoyer">
    </p>
</form>
```

Exemple de formulaire

Nom

Prénom

Email

Mot de passe

Newsletter

Envoyer

CSS, la suite

Pseudo-classes pour les champs

3 pseudo-classes utiles pour rendre les champs un peu plus jolis

- **valid :** permet d'affecter du CSS à un champ valide
- **invalid :** permet d'affecter du CSS à un champ invalide
- **required :** permet d'affecter du CSS à un champ obligatoire non rempli

À vous de jouer maintenant

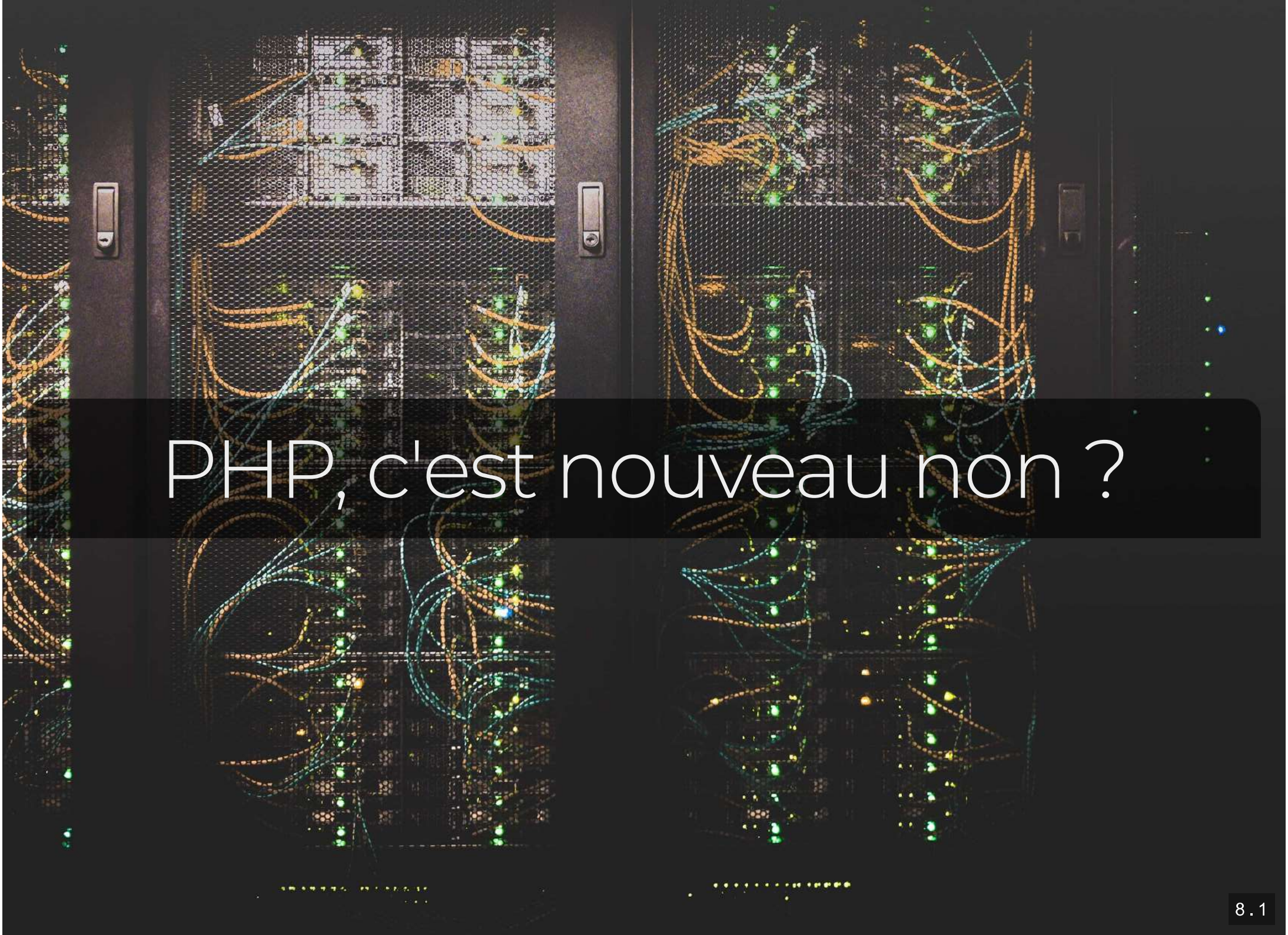
- Créer un formulaire de type **POST** menant à une page **traitement.php**
- Ajouter un champ texte pour le nom, le rendre obligatoire
- Ajouter un champ texte pour le prénom, le rendre obligatoire
- Ajouter un champ nombre pour l'âge
- Ajouter une liste déroulante pour le sexe de la personne, (Homme ou Femme)
- Ajouter un bouton pour envoyer le formulaire
- Chaque champ a son label associé

Correction

```html
<form action="traitement.php" method="post">
    <p>
        <label for="nom">Nom</label>
        <input type="text" id="nom" name="nom" required/>
    </p>
    <p>
        <label for="prenom">Prénom</label>
        <input type="text" id="prenom" name="prenom" required/>
    </p>
    <p>
        <label for="age">Âge</label>
        <input type="number" id="age" name="age"/>
    </p>
    <p>
        <label for="sexe">Sexe</label>
        <select id="sexe" name="sexe">
            <option value="h">Homme</option>
            <option value="f">Femme</option>
        </select>
    </p>
</form>
```

< HTML />

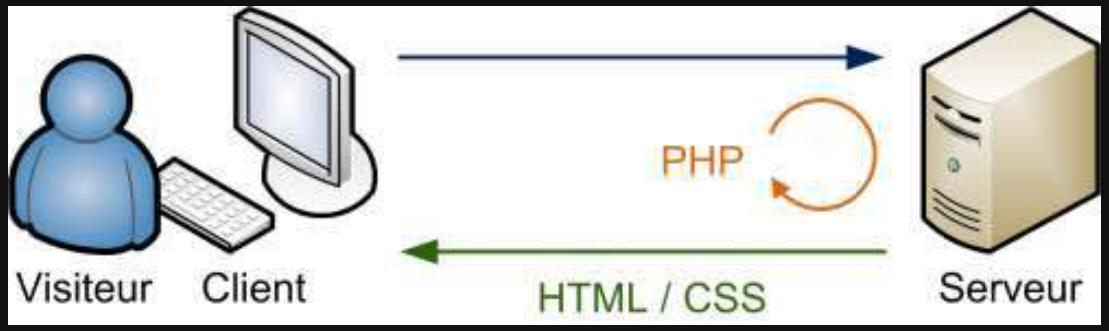
7.2


PHP, c'est nouveau non ?

PHP, c'est quoi ?

PHP (Hypertext Preprocessor) est un langage de programmation que seuls les serveurs peuvent interpréter.

Ainsi on ne peut pas simplement créer des pages PHP et les ouvrir dans le navigateur, on passe par un serveur Web qui va traduire le tout en HTML.



Génération d'HTML par le serveur Web grâce à PHP

Le PHP va permettre de générer du code HTML dynamiquement, qui pourra changer en fonction de ce que l'utilisateur demande

PHP, les variables

PHP, comme beaucoup de langages de programmation, utilise des variables.

Les variables sont typées implicitement, autrement dit, on peut stocker tout type d'information(s) dans une variable

```php
<?php /* Ouverture du code php */
$chaine = 'Salut !';
$entier = 15;
$decimal = 1.5;
$boolean = true;
$tableau = ['zero', 'deux', 'quatre'];
?> /* Fermeture du code PHP */
```

PHP, les variables

Il est possible de voir le type et la valeur d'une variable grâce à une fonction de PHP

```php
<?php /* Ouverture du code php */
$chaine = 'Salut !';
var_dump($chaine);
?> /* Fermeture du code PHP */
```

```
/home/birssan/projects/tmp/c5/index.php:548:string 'Salut !' (length=7)
```

On se sert de cette fonction pour déboguer

Les chaînes

Pour afficher un site dynamique, on utilise du contenu qui peut être dans des variables, on peut tout de même les concaténer avec du texte brut

On utilise le mot clé **echo** pour afficher du texte ou le contenu d'une variable

```php
<?php /* Ouverture du code php */
$user = 'Seb';
echo '<em>Salut ' . $user . ' !</em>';
?> /* Fermeture du code PHP */
```

Salut Seb !

Les conditions

Comme dans tout langage, les conditions permettent d'effectuer des actions en fonction des cas

```php
<?php /* Ouverture du code php */
$now = date('G'); /* heure actuelle */
if($now > 6 && $now < 12) {
  echo 'Good Morning';
}
else if($now >= 12 && $now < 18) {
  echo 'Good Afternoon';
}
else {
  echo 'Good Night';
}
?> /* Fermeture du code PHP */
```

Quelques fonctions utiles

- **strlen :** permet de connaitre le nombre de caractères dans une chaîne
- **strtolower :** transforme toute une chaîne de caractères en minuscule
- **strtoupper :** transforme toute une chaîne de caractères en majuscule
- **substr :** extrait une partie d'une chaîne de caractères
- **isset :** retourne **false** si la variable passée en paramètre n'existe pas
- **empty :** détermine si une variable est vide ou égale à **false**

Traiter un formulaire

Pour récupérer les données envoyées par un formulaire en PHP, on utilise des variables superglobales

- **$_GET :** tableau contenant les données envoyées par GET (URL)
- **$_POST :** tableau contenant les données envoyées par POST

Ces deux variables superglobales sont des tableaux dits associatifs contenant des paires **clé/valeur**

Traiter un formulaire

```php
<?php /* Ouverture du code php */
/* Si le nom n'a pas été envoyé dans le formulaire
on affiche un message d'erreur */
if(empty($_POST['nom'])) {
    echo 'Le nom doit être renseigné !';
}
else {
    echo 'Bonjour ' . $_POST['nom'];
}
?> /* Fermeture du code PHP */
```

La fonction include

Il est possible d'intégrer des fichiers dans des fichiers avec la fonction **include**

Grâce à cela, on peut par exemple intégrer des morceaux de code HTML qui sont communs à plusieurs pages

```php
<?php
include('header.php');
?>
<div id="content">
...
</div>
<?php
include('footer.php');
?>
```


Questions ?