Université Paris Nanterre

Université Paris Nanterre

Programmation web et Introduction à PHP

L2 MIASHS

M. NAFI

m.nafi@parisnanterre.fr

Année: 2022/2023

Objectifs du cours

- Création de **sites web statiques, interactifs** et **dynamiques** à l'aide du:
  - Langage **HTML**
  - Langage **CSS**
  - Langage **JavaScript**
  - Langage **PHP**

Plan du cours

1. Introduction
2. Langage Html
3. Langage CSS
4. Langage Javascript
5. Langage PHP

Plan du cours

1. Introduction
2. **Langage HTML**
3. Langage CSS
4. Langage Javascript
5. Langage PHP

Langage HTML

Historique

- HTML a évolué à partir de SGML (**S**tandard **G**eneralized **M**arkup **L**anguage). Un méta langage de la norme ISO

- À la fin des années **80**, le britanique Tim Berners-Lee, employé au CERN (suisse), developpa un langage de balisage qui donna naissance à HTML.

Définition

- HTML (**H**yper**T**ext **M**arkup **L**anguage) (langage de **balisage,** liens **hypertextes** ou hyperliens)

- HTML est un langage de **description** qui fait appel aux balises pour structurer le document

- Il permet de définir et de structurer les éléments d'un document textuel à l'aide de titres, paragraphes, listes, tableaux, etc
- Permet d'ajouter du **contenu** aux pages web

- Le code HTML est interprété par des agents utilisateurs au niveau des navigateurs

- Une page web est un fichier texte ayant l'extension **.html**

- Le code source des pages web est **gratuit** et **accessible**

Structure de base d'un document HTML

```
<!DOCTYPE html>          // Type et la version du document
<html>                   // l'élément racine
    <head>
        <meta charset="utf-8">          //En-tête
        <title> Titre de la page </title>
        …
    </head>
    <body>
        <!– le contenu de la page-->
        <p>ceci est un paragraphe</p>     // Corps
        <a href='www.exemple.com' > accéder au site</a>
        …
    </body>
</html>
```

DOCTYPE

- Un document HTML débute toujours par une ligne nommée DOCTYPE, avant même la première balise **<html>**
- Cette déclaration renseigne le navigateur sur le type du document et la version du HTML utilisée
- **Ex.** <!doctype html> document HTML5
- La version 5 est simplifiée par rapport aux précédentes

  Ex. HTML 4.01

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML4.01//EN"http://www.w3.org/TR/html4/strict.dtd">

Elément, balise et attribut

❑ Elément <*NomElement*> Contenu </*NomElement*>

- Un élément peut ne pas avoir de balise fermante (Ex. img, link, base)
- 02 types d'éléments:
  - **Bloc:** retour automatique à la ligne (<p>, , …)
  - **Ligne:** pas de retour à la ligne (, , …)
- Un élément en bloc peut contenir d'autres éléments en bloc ou en ligne
- Cependant, un élément en ligne ne peut contenir que des éléments en ligne

Élément

- Les éléments peuvent se **succéder**

Ex. <h1> …</h1> …

- ou **s'imbriquer**

Ex. ………

- doit posséder un sous élément <li> (élément enfant) **Ex**.

```
<ul>
    <li> </li>
    <li></li>
</ul>
```

❑ Balise

- Forme concrète d'un élément et permet de structurer le document
- Une balise débute par le chevron ou le signe < et finit par le signe >
- **Deux** types de balises:
  - Balise **ouvrante**: <*NomElement*>
  - Balise **fermante**: </ *NomElement* > le nom de l'élément est précédé du caractère **slash (/)**

Ex. <html> … </html>

❑ Attribut

- **Propriété** d'un élément (nom= "valeur")
- Modifie les propriétés des balises
- Un élément peut comporter **zéro** ou **plusieurs** attributs
- Plusieurs attributs sont séparés par des espaces
- Les attributs font partie de la balise ouvrante
- Certains attributs sont obligatoires comme **src** dans la balise <img src="url">, d'autres sont facultatifs comme alt.

Ex. Type="Text"
href="www.parisnanterre.fr"

Elément racine

- **<html>** est l'élément **racine** de tout document HTML
- Cet élément possède deux enfants directs:

<head> et <body>

- L'attribut global applicable à cet élément est lang qui spécifie la *langue* (si le champs est vide ie langue est inconnue)
- **Ex.** <html lang="fr">

En-tête (head) (1/2)

- Fournit de multiples informations sur le document (Titre, métadonnées (auteur, mots clés, description, etc), liens, style, script, base)
- Ces informations ne sont pas affichées dans le navigateur excepté le **titre**
- À ne pas confondre avec les éléments *heading (titres)* <h1> à <h6>

En-tête (2/2)

- Dans un en-tête, on trouve:
  - Titre du document: <title>..</title>
  - Métadonnées: <meta … />
  - Liens: <link>
  - Style: <style> </style>
  - Script: <script></script>
  - Base: <base>

Titre

- Le titre du document est affiché par le navigateur dans la **barre de titre** ou **l'onglet actif**
- Le titre est utilisé comme intitulé des liens par les moteurs de recherche
- Il est introduit par les balises **<title>** Titre du document **</title>**
- Le titre est le seul élément **obligatoire** dans l'élément <head> d'un document

Métadonnées

- La balise **meta** est utilisée pour fournir davantage d'informations sur le document telles que le nom de l'auteur, la description du contenu, les mots clés, etc
- Cette balise ne possède pas de balise **fermante**
- Ces informations ne seront pas affichées dans la page
- On distingue plusieurs applications:
  - Attribut name et content
  - Attribut http-equiv et content
  - Attribut charset

meta name

- Mots clés: <meta name='**keywords**' content='html, css ,Js'>
- Description: <meta name='**description**' content='Le cours de HTML, CSS et Js'>
- Auteur: <meta name="**author**" content="M. NAFI">
- Dimension de la page: <meta name="**viewport**" content="width=300, initial-scale=1">

meta http-equiv

- **Rafraichissement** de la page

<meta http-equiv="refresh" content= "30" />
(actualisation de la page toute les 30 secondes)

- **Redirection** vers une URL après un délai exprimé en secondes:

<meta http-equiv="refresh" content= "30; url= www.google.fr" />

meta charset

- **Jeu de caractère** spécifie l'encodage de caractères utilisé
- Les valeurs les plus utilisées sont: UTF-8, ISO-8859-1, ASCII

Ex. <meta **charset**= "UTF-8">

Style

- La balise <style> est utilisée pour ajouter à la page du code **CSS interne**
- Elle possède les attributs **type** et **media**
- **Ex.** <style type='text/css'>

```
p {
    color : red;
}
</style>
```

<style **media**="screen"> … </style>

<style **media**="print"> … </style>

Script (Javascript)

- Le code JavaScript est introduit avec l'élément <script>…</script>
- Code interne
- **Ex.** <script>

```
document.getElementById("ID").innerHTML = "Bonjour!";
</script>
```

- Code externe

  Ex.

```
<script type="text/javascript" src="myscript.js"> </script>
```

Liens (1/2)

- Insérer un lien vers d'autre pages ou ressources externes comme un fichier (css, js,…) à l'aide de la balise **<link>**
- Cet élément ne possède pas de balise fermante
- Il possède les attributs :
  - **rel: relation**
  - **type: type du fichier**
  - **href:** Adresse absolue ou relative de la cible

Liens (2/2)

- Adresse relative => 03 cas peuvent se présenter:
  - Fichier se trouve dans **le <u>même</u>** dossier que la page (le répertoire courant)

Ex. <link rel="stylesheet" type="text/css " href="**style.css** " >

  - Fichier se trouve dans un **<u>sous</u>** <u>dossier</u>(niveau **inférieur**)

Ex.<link rel= "stylesheet " type="text/css"
href="**dossier/**style.css " > 1 seul niveau

  - Fichier se trouve dans un dossier **parent** (niveau **<u>supérieur)</u>**

Ex.<link rel="stylesheet" type="text/css" href="**../**style.css">
1 seul niveau

Base

- L'élément <base> définit l'URL de base pour tous les liens relatifs dans une page web
- Il ne peut y avoir qu'un seul élément base dans un document
- Si plusieurs balises <base> sont utilisées, seule la première sera prise en compte par les navigateurs web.
- Ex.

```
<base href="https://www.google.com/">
```

Corps du document html

- Le corps d'un document html est défini par la balise <body>…</body>
- Il peut être structuré comme suit:
  - En-tête: <header >..</header >
  - Menu de navigation: < nav>..</ nav>
  - Contenu principal: < main>…</main >
  - Section: <section >..</ section>
  - Article: < article>..</article >
  - Pied de page: < footer>..</ footer>
- Chacun de ces éléments peut avoir des titres, paragraphes, images, liens, tableaux, formulaires, etc.

Titres

- Définir des titres et sous titres (*headings*)
- Il existe 6 niveaux: h1 à h6
- <h1> le plus **important**: les moteurs de recherche leur affectent un poids plus **élevé**
- <h6> le moins important
- Par défaut, les titres sont affichés en **gras**
- **Ex.**

  <h1> Titre de niveau 1 <h1>

  <h3> Titre de niveau 3 <h3>

Paragraphe

- **Paragraphe** … permet d'ajouter du texte à la page
  - Par défaut, une ligne **avant** et **après** le paragraphe seront affichées par les navigateurs
  - Un paragraphe peut contenir du **texte**, des **images** et les autres éléments en **ligne** (, <b>, etc).

- **Séparateurs** :
  -
 : retour à la ligne
  - <hr> : ligne horizontale

Formatage du texte

- **Gras** ou texte important: … , …
- **Italic**: …,
- Mise en **évidence**: <mark>…</mark>
- Mise en **indice** : _…
- Mise en **exposant** : […]
- **Citation**: <cite>…</cite>
- etc

Liens (ancres) (1/3)

- Un lien hypertexte vers un endroit précis d'une page ou vers une autre page se fait à l'aide de l'élément anchor <a>…
- L'attribut **href** ((**h**ypertext **ref**erence) est utilisé pour définir l'adresse ou l'URL de la page cible

Ex.

 lien vers un endroit précis d'une page (section)

 lien vers une autre page

Liens (2/3)

Deux types d'URLs: absolue et relative

- **Absolue**: adresse complète:

  protocole, **nom du domaine**, **chemin** si nécessaire

- Utilisée pour faire référence à une ressource externe (serveur ou site différent) ie lien externe
- **Ex.** Le site de l'université https => aller sur le web

Liens (3/3)

- **Relative**: chemin vers un fichier ie lien **interne**
  - Sans le http, le navigateur regarde sur le serveur **local**
  - Utilisée pour faire référence à une ressource interne (*même serveur* ou site)
    - Lien à l'intérieur d'un dossier
      - Ex. Page d'acceuil
    - Lien vers un **sous** dossier (1 niveau)
      - Ex. Aide

Liens (3/3)

- Lien vers un dossier **parent** (1 niveau)
- Ex. Page d'acceuil
- Lien à partir de la racine, on ajoute un "/"

- Nb: on peut mettre n'importe quel élément html entre <a> et pour le transformer en un lien
- Ex. Une image comme lien:
- <img src="logo.gif" alt=" logo">

Images

- L'insertion d'images se fait à l'aide de la balise <img>
- Cet élément ne possède pas de balise fermante
- La balise <img> possède les attributs suivants:
  - **src** : la **source** de l'image ou le chemin vers l'image (obligatoire)
  - **height**: **hauteur** de l'image (px, em, %, etc)
  - **width**: spécifie la **largeur** de l'image (px, em, %, etc)
  - **alt**: texte **alternatif** en cas d'échec de l'affichage ou chargement de l'image
- **Ex.** <img src="Image.jpg" height="100px" width="100px" alt="Image">

Liste

- 03 types de listes :
- Non ordonnée. l'ordre des items n'est pas important.
- Ordonnée. l'ordre des items est important.
- Définition ou description. Elle consiste en paires (nom, valeur)

Liste non ordonnée

- Liste non ordonnée est définie avec la balise
- Cette balise possède des items **<li>**
- Les items ** ne** sont **pas** ordonnés (précédés par des puces par défaut) => l'ordre n'est pas important
- Seul des <li> sont permis entre <ul> et
- **Ex.**

```
<ul>
        <li> HTML</li>
        <li> CSS </li>
        <li> JavaScript </li>
</ul>
```

Liste ordonnée

- Liste ordonnée est introduite avec la balise
- Liste où l'ordre est **important** (Ex. recette de cuisine, liste des étudiants admis selon l'ordre de mérite, etc)
- À la place des puces, le navigateur insère des nombres (par défaut) ou des lettres
- Les items **<li>** sont ordonnés avec des chiffres ou lettres alphabétiques
- **Ex.** <ol type= "a" >
   HTML => Liste avec des lettres
   CSS

Remarque

- Si l'on souhaite que la liste commence à partir d'un nombre qui est diffèrent de "1", on utilise l'attribut **start** comme suit:
- Ex. <ol start= "5 "> //commence à partir de 5

```
        <li> HTML </li>
        <li> CSS </li>
        <li> JavaScript </li>
</ol>
```

Liste de description (1/2)

- Liste de *description* est définie avec la balise <dl> </dl>
- Un élément <dl> contient un certain nombre de d' éléments <dt> et leur <dd> respectifs
- Le **nom** ou le **terme** est introduit avec <dt></dt>
- La **valeur** ou la définition est introduite avec <dd> </dd>

Liste de description (2/2)

- **Ex.** <dl>
  <dt> HTML </dt>
  <dd> Langage de balisage </dd>
  <dt> PHP </dt>
  <dd> Langage de script </dd>
  </dl>
- NB: Pour un même terme <dt>, on peut avoir plusieurs définitions <dd>

Tableau (1/3)

- Pour créer un tableau, on utilise la balise <table>
- Cet élément possède une balise ouvrante <table> et fermante </table>
- En HTML, un tableau consiste en un ensemble de **lignes**
- Chaque ligne est introduite à l'aide de la balise **<tr>**
- <th>: permet d'insérer l'entête du tableau
- <td>: permet d'insérer les données du tableau

Tableau (2/3)

- Ex.

```
<table>
    <tr>
        <th> Nom </th>
        <th> Age </th>
    </tr>
    <tr>
        <td> Jean </td>
        <td> 30 </td>
    </tr>
</table>
```

Tableau à 2 lignes et 2 colonnes

Nom	Age
Jean	30

Tableau (3/3)

- Attributs:
  - **colspan**: fusion de cellules d'une même ligne en définissant la valeur de l'attribut colspan d'un élément <td> ou <th> avec un entier
- Cet entier indique le nombre de cellules à fusionner en partant de la gauche
- Ex. <td colspan="N">Contenu de la cellule</td> **N** cellules
  - **rowspan**: fusion de cellules situées dans les lignes ***adjacentes***
- Ex. <td rowspan="N"> Contenu de la cellule</td>

Formulaire (1/9)

- Un formulaire est introduit avec l'élément <form>…</form> et peut contenir des champs de saisie, des boutons, textes, etc
- Il permet de **collecter** les informations des utilisateurs
- Les informations collectées sont traitées par un **script** ou une application au niveau du serveur
- Cet élément peut contenir d'autre éléments en bloc mais **ne** peut pas contenir d'autres élément <form>

Formulaire (2/9)

- L'élément <form> possède les attributs: action, method
  - **action**=" URL " l'adresse du script ou de l'application qui traitera les données saisies
  - **action**="": Si le code PHP se trouve sur la page elle-même
- Ex. <form action="/script.php" >…</form>

Formulaire (3/9)

- method: spécifie comment l'information devrait être envoyée au serveur. Il existe deux méthodes: GET et POST
- Cet attribut est **optionnel**. S'il n'est pas spécifié, la méthode GET est choisie par défaut

Formulaire (4/9)

- GET vs POST
  - GET: données saisies sont insérées dans l'URL après « ? »
    - Nombre de caractères à envoyer est limité
    - Données sont visibles à tous
  - POST: données insérées dans une requête HTTP
    - Nombre de caractères est illimité
    - Sécurisé: seul le serveur est capable de lire les données

Formulaire (5/9)

- Types de **contrôles**: la plupart des éléments sont introduits avec l'attribut « type ». **L'apparence** de l'élément change en fonction de la valeur de « type »
- Plusieurs types de contrôles en HTML5:
  - Champs de texte
    - Monoligne: <input type=" text " name= " nom " value= "Jean" placeholder='Text' maxlength= "25" >
      - **name**: nom de l'élément (obligatoire)
      - **value**: valeur par défaut lors du chargement de la page
      - **placeholder**: son contenu n'est pas envoyé
      - **maxlength** et **minlength**: nombre de caractères

Formulaire (6/9)

- Multilignes: l'élément <textarea> est utilisé lorsque l'utilisateur souhaite saisir plus d'une ligne
- Ex. <textarea name= "adresse " placeholder=" pas plus de 50 caractères " rows= "50 " cols= "5">
- *placeholder*: Son contenu ne sera pas émis au serveur
- *content*: Son contenu sera transmis au serveur

Formulaire (7/9)

- **Champs de texte particuliers:**

- Mot de passe: <input type= " password " …>

  Texte **invisible** mais ne veut pas dire chiffré
- Recherche: <input type= "search " …>
- Email: <input type= " email " …>
- Téléphone : <input type= " tel " …>
- Adresse mail: <input type= "url " …>

Formulaire (8/9)

- Bouton: <input type= " button " …>
  <input type= " radio " …>
  <input type= " checkbox " …>
- Menu: <datalist id="lang">
  <option value="HTML">
  <option value="CSS">
  </datalist>
  <select name="nom">
  <option>HTML</option>
  <option> CSS</option>
  </select>

Formulaire (9/9)

- Fichier: <input type="file" …>
- Date et heure: <input type="date" …>
  <input type="time" …>
- Couleur: <input type="color" …>
- etc

Division (1/2)

- L'élément est utilisé pour créer une division ou section dans un document html
- Cet élément **n'a pas** de signification propre en HTML mais utilisé conjointement avec le CSS
- C'est un élément par **bloc** ie débute et finit par un retour à la ligne
- **Ex.** Voici une liste d'articles:

```
<ol>
    <li> Ordinateur </li>
    <li> Imprimante </li>
</ol>
</div>
```

Division (2/2)

- L'élément span est similaire à div sauf que celui-ci est un élément en **ligne** ie il ne débute pas et ne finit pas par un retour à la ligne
- **Ex.** Le premier est un élément par bloc, le second est un élément en ligne .
- **Résultat**

  Le premier est un

  élément par bloc

  , le second est un élément en ligne.

Section (1/2)

- Elle sert à diviser le corps du document HTML
- **04** éléments sont utilisés pour définir les différentes sections:
  - <header>
  - <nav>
  - <main>
  - <footer>
- Ces éléments font partie de l'élément **body**

Section (2/2)

- <header>…</header>: section en **haut** de la page. Elle peut contenir le logo, etc
- Ne pas confondre avec l'élément <head> dont le contenu ne sera pas affiché dans la page (sauf le titre)
- <nav>…</nav> : liens de navigation (menu)
- <main>…</main>: section principale de la page
- <footer>…</footer>: section en **bas** de page (liens additionnels, copyright, contacts)

Commentaires

- Un commentaire est introduit par les symboles <!– et -->

- Permet de rendre le document HTML facile à lire et à comprendre

- Les commentaires ne seront pas affichés par les navigateurs

Validation du HTML

- Le site pour vérifier le code html en ligne est https://validator.w3.org/
- 03 techniques:
  - Validation par URI
  - Validation par fichier contenant le code HTML
  - Validation par la saisie directe du code HTML