

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 de:

- Langage **html**
- Langage **CSS**
- Langage **JavaScript**
- Langage **PHP**

Prérequis

- Avoir des connaissances sur le langage HTML

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 CSS

Introduction

- CSS(Cascading StyleSheet): **Feuilles de style en cascades** est un standard de W3C pour définir la **présentation** des pages web
- Il s'occupe du **style** ou de **l'apparence** des éléments d'un document HTML
- Le langage CSS possède sa propre syntaxe
- La dernière version est **CSS3**

Syntaxe (1/2)

- Une **règle** CSS décrit comment un élément ou un ensemble d'éléments devraient être affichés par le navigateur
- Elle a la forme suivante:

```
Selecteur {  
    propriété 1: valeur 1;    /* déclaration 1  
    propriété 2: valeur 2;    /* déclaration 2  
    ...  
    propriété N: valeur N;    /* déclaration N  
}
```

Syntaxe (2/2)

□ Une règle CSS est constituée de:

- **Sélecteur** qui identifie l'élément ou les éléments qui vont être affectés par le style
- Elle indique au navigateur que le code CSS qui suit s'applique à cet élément ou ensemble d'éléments
- **Déclaration** qui fournit des instructions d'affichage
 - Composée d'une **propriété** et d'une **valeur** séparée par deux points ':'.
• Les déclarations sont séparées entre elles par des ';'.
- **Ex.**

```
h1 {color: blue;}  
body {background-color: orange;}
```


Types de sélecteur

□ Il existe plusieurs types de sélecteurs:

- Sélecteur universel
- Sélecteur d'élément
- Sélecteur d'identifiant
- Sélecteur de classe
- Sélecteur d'attribut
- Sélecteur de pseudo-classe
- Sélecteur de pseudo-élément
- etc

Sélecteur universel

- Le sélecteur universel noté ‘*’, est utilisé pour appliquer un style à tous les éléments du document HTML

- Ex. * {
 color:blue;
}

=> Tous le **texte** de la page web sera affiché en **bleu**.

Sélecteur d'élément

- Le code CSS s'applique à **tous** les éléments de **même type**
- Il est constitué du **nom** de l'élément (**p**, **div**, **table**, etc)
- Il permet de cibler un élément HTML spécifique

Ex.

```
h1 {  
    color:green;  
}
```

=> Tous les titres de niveaux 1 seront affichés en **vert**

Remarque

- On peut appliquer le même style à plusieurs éléments différents en les énumérant et séparant par une **virgule** (,) => Ce regroupement est très utile
- Ex.

h1 {color:blue;} **et** **p** {color:blue;}



h1, p {color:blue;}

Sélecteur d'élément enfant

- Les éléments **parent** et **enfant** sont séparés par un espace
- Ex. `<div p { /* code css */ }`
 - => 'p' est l'élément enfant de 'div' ('div' est son parent)
 - Le code css s'appliquera seulement aux paragraphes qui se trouvent à l'intérieur de l'élément 'div'
- Ex. `<div>`
 - `<p> Paragraphe 1</p>`
 - `</div>`
 - `<p>Paragraphe 2</p>`
 - => Le style CSS ne s'appliquera qu'au 'Paragraphe1'

Sélecteur d'élément enfant

- **first-child**: premier enfant
- **nth-child(n)**: **n** ème enfant
- **last-child**: dernier enfant
- **Ex.** `<div>`
 - `<p>premier enfant</p>`
 - `<p>second enfant</p>`
 - `<p>troisième enfant</p>`
 - `</div>`
- cibler le second => `p:nth-child(2){color:red;}`

Question

Lorsqu'on souhaite cibler un **seul** élément ou un élément **spécifique**.

Quel est le type de sélecteur le plus approprié?

Réponse

On peut utiliser le sélecteur **d'identifiant**

Sélecteur d'ID

- Ajouter l'attribut '**id**' à un élément du document HTML
- Un identifiant doit être **unique** dans une page web
- Le nom du sélecteur commence toujours par le caractère '**#**' suivi par la valeur de l'**id**

Ex. `<h1 id='titre1'>` `/* HTML`

`#titre1 { color:red; }` `/* code CSS`

- Ce style ne s'appliquera qu'à un **seul** élément (l'élément dont la valeur de l'id est '**titre1**').

Question

Lorsqu'on souhaite cibler simultanément **plusieurs** éléments de différents types.

Quel est le type de sélecteur le mieux adapté?

Réponse

On peut utiliser le sélecteur de **classe**

Sélecteur de classe

- Pas besoin d'être unique dans une page
- Un élément peut avoir plusieurs classes séparées par un **espace**
- Ex. `<p class='classe1 classe2'>`
- Ajouter l'attribut **class** aux éléments en question
- Ex. `<h1 class='titre'>` `<h2 class='titre'>`
- Le nom du sélecteur commence toujours par le caractère `'.'` suivi par le nom de la classe
- Ex. `.titre {color:red;}`

Sélecteur d'attribut

- Cibler des éléments selon le nom ou la valeur d'un attribut spécifique
- Ex. `a[href]{color:blue;}`

`a[href='<u>www.parisnanerre.fr</u>']{color:blue;}`

=> Sélectionner tous les liens vers l'adresse

www.parisnanerre.fr

Sélecteur de pseudo-classes

- Pseudo-classe : **état** d'un élément
- On utilise la notation: '**élément:état**'
- **Ex.** l'élément **a** possède 4 états:
 - **link** (lien non visité)
 - **visited** (lien visité)
 - **hover** (lien survolé avec la souris)
 - **active** (lien cliqué)

Ex. **a:hover** { color:white; }
a:visited { color:blue; }

Sélecteur de pseudo élément

- On utilise les ‘`::`’

- ❑ Première lettre:

`<p> Paragraphe</p>`

`p::first-letter { /* code css */ }`

- ❑ Première ligne d'un texte

`p::first-line { /* code css */ }`

- ❑ Insertion d'un contenu

Avant : `before`

Après: `after`

Ex. `p::after {`
 `content: ".";`
 `} /* ajoute un point à la fin de chaque paragraphe`

Application du code CSS

03 méthodes existent:

- Lien vers un fichier CSS externe
- En utilisant l'élément `<style> </style>`
- CSS en ligne (l'attribut `style`)

❑ Lien vers un fichier externe

Le code CSS est écrit dans un fichier texte **séparé** et sauvegarder sous l'extension **.css**

- Relier le fichier CSS au fichier HTML à l'aide de la balise `<link>` dans l'élément `<head>`

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

- Les attributs **rel** et **type** indiquent au navigateur qu'il s'agit d'un fichier CSS
- L'attribut **href** spécifie le chemin vers le fichier CSS

❑ En utilisant l'élément `<style> </style>`

Le code CSS est incorporé dans l'élément `<head>` du document HTML

Ex.

```
<head>  
    <style>  
        /* code CSS  
    </style>  
</head>
```


❑ CSS en ligne (l'attribut style)

Le code CSS est inséré dans la balise ouvrante d'un élément donné en utilisant l'attribut **style**

Ex. `<p style= " color:'red'; " > Paragraphe</p>`

Remarque

- ❑ La première méthode est la plus recommandée car elle permet de séparer le code CSS du code HTML
- Très utile lorsqu'on souhaite appliquer le même code CSS à **plusieurs pages web**
- ❑ La seconde est communément utilisée quand on veut appliquer le même code CSS à **une seule page web**
- ❑ La dernière méthode est utilisée quand on souhaite appliquer des règles CSS à **un seul élément**

Sensibilité à la casse

- Nom des éléments n'est pas sensible

Ex. **P**{} et **p**{} sont identiques => pas d'erreur

- Le nom des **classes** et **identifiants** sont sensibles

Ex. **Classe1** et **classe1** sont différentes => erreur

Ordre de précedence

- ❑ Plus le sélecteur est spécifique, plus est la précedence

Element#id > id > classe > élément

- Ex. `<p id="p1" class="c1">PARAGRAPHE</p>`

Style1:

`p {color:"green";}`

`#p1 {color:"orange";}`

`.c1 {color:"red";}`

Résultat: la **seconde** règle sera appliquée ie le paragraphe sera donc affiché en couleur 'orange'

Style 2: `p {color:"green";}`
`#p1 {color:"orange";}`
`p#p1 {color:"red";}`
`.c1 {color:"white "};`

Résultat: la 3ème règle sera appliquée ie le paragraphe sera donc affiché en couleur ‘rouge’

❑ Un élément **hérite** le style de son parent si **aucun** style n'est spécifié

• Ex. `<p>` Texte en

`` Gras ``

`</p>`

Code CSS:

```
p { color: "blue "; }
```

=> La couleur bleue sera appliquée à la fois à 'texte en' et 'gras' si la règle `strong {color:'autre couleur';}` n'est pas définie

❑ Si des règles ont la **même** précedence, la **dernière** sera appliquée

- Ex.

```
p { font-size: 10px; }  
p { font-size: 12px; }
```

=> Le paragraphe sera affiché avec une police de 12 pixels. ie la dernière règle sera appliquée.

Notion de cascade

- Elle permet de définir l'ordre d'application des propriétés lors d'un conflit
- L'ordre est défini par priorité croissante:
 - Style par **défaut** du navigateur
 - Source **externe** fichier.css
 - Source **interne** avec un style défini dans la page
 - Source en **ligne** avec un style défini dans l'élément HTML (**haute priorité**)

Commentaire

- Le texte présent entre `/*` et `*/` sera ignoré
- Il ne sera pas affiché par le navigateur

Propriétés CSS

□ Texte et police

- **Couleur**: la propriété '**color**' spécifie la couleur du **texte**
- La couleur peut être spécifiée de **trois** manières:
 - Nom de la couleur (ex. **color**: 'blue')
 - RGB (ex. **color**: 'rgb(0,0, 255)')
 - Hexadécimale (ex. **color**: '#0000ff')

○ Alignement du texte:

- La propriété ‘**text-align**’ permet de:

- Centrer le texte `text-align:center;`
- Le positionner à gauche `text-align:left;`
- Le placer à droite `text-align:right;`
- Le justifier `text-align:justify;`

○ Décoration du texte

- La propriété ‘**text-decoration**’ permet de:
 - Souligner le texte : **text-decoration:underline;**
 - Rayer ou surligner le texte: **text-decoration:overline;**
 - Laisser le texte normal **text-decoration:none;**

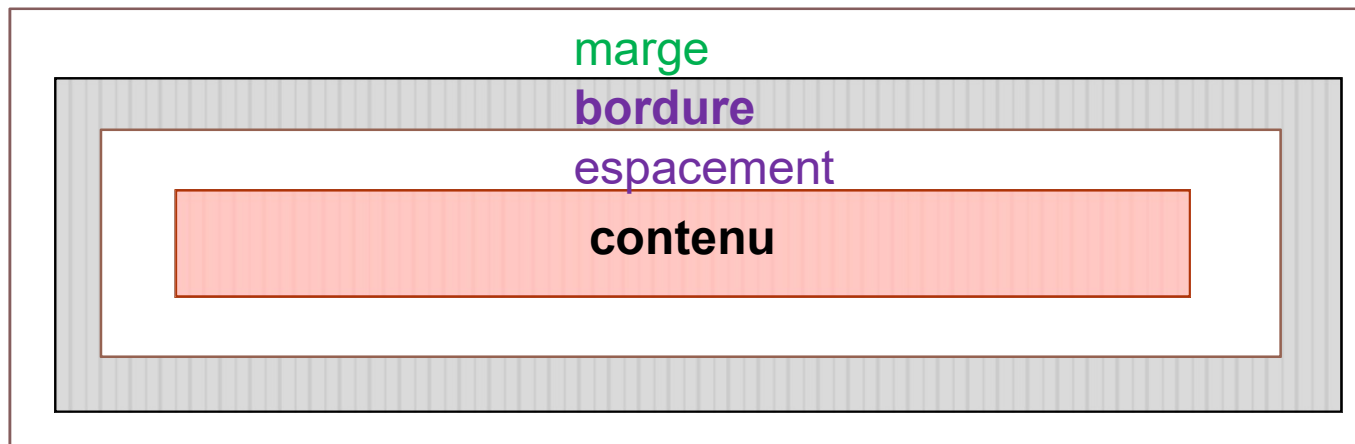
○ Espacement

- **Lettre**: La propriété '**letter-spacing**' permet de:
 - Diminuer l'espace entre les lettres d'un mot (valeur **négative**) **letter-spacing:-2px;**
 - Augmenter l'espace entre les lettres d'un mot (valeur **positive**) **letter-spacing:2px;**
- **Mot**: La propriété '**word-spacing**' permet de:
 - Diminuer l'espace entre les mots (valeur **négative**) **word-spacing:-2px;**
 - Augmenter l'espace entre les mots (valeur **positive**) **word-spacing:2px;**

- **Ligne**: La propriété '**line-height**' permet de spécifier l'espace entre les lignes d'un texte. Elle est exprimée en **pixel**, **pourcentage**, **em**, **cm** et **pt** (point)
- **Ex.** **line-height:20px;**
line-height:20%;

Modèle en boîte

- En CSS, un élément est traité comme une **boîte**
- A chaque élément HTML correspond une zone rectangulaire
- Chaque boîte ou élément possède des **marges** (margin), **bordures** (border), **espacements** (padding) et un **contenu** (content)



- À l'intérieur de la bordure, on trouve les espacements (padding) et le contenu (content)
- À l'extérieur de la bordure, on trouve les marges (margin)

Propriétés

❑ Contenu

- **Width** et **height**: dimension de la zone de **contenu** (largeur et hauteur). Les valeurs sont exprimées en **pixels** (px) ou en **pourcentage** (%)
- Elles peuvent aussi prendre la valeur **auto**
- **Overflow**: parfois la zone de contenu est trop petite pour contenir la totalité du texte. Par conséquent, le texte va déborder. Pour y remédier, on utilise la propriété **overflow**
- Cette dernière peut prendre les valeurs: **visible**, **hidden**, **scroll**, **auto**

❏ Padding et margin

- **Padding**: espace vide entre le contenu et la bordure
- **Margin**: espace entre la bordure et les éléments adjacents
- Ils sont **transparents** (on ne peut pas changer leur couleurs)
- Il existe **04** syntaxes différentes:

- **Ex.** les 03 premières notations sont équivalentes
 - `Margin-top:10px;`
`Margin-right:20px;`
`Margin-botom:10px;`
`Margin-left:20px;`
 - `Margin: 10px 20px 10px 20px;`
 - `Margin: 10px 20px;`
 - `Margin:10px;` Toutes les marges sont identiques

❑ Bordure

- La bordure possède les caractéristiques suivantes:
largeur, style et couleur.

➤ Largeur

`border-width: thin | medium | thick | nombre positif`

➤ Style

`border-style: none | hidden | dotted | dashed | solid |
double | groove | ridge | inset | outset`

- Cette règle s'applique au 4 bordures:

- `border-top-style:none`
- `border-right-style: solid`
- `border-bottom-style:none`
- `border-left-style:solid`

➤ Couleur

`border-color:rgb(255,0,0) | red | #FF0000 | transparent`

- **Définition globale:** `border:largeur | style | couleur`

Ex. `border:3px solid red;`

- **Bordure arrondie:** `border-radius:20px;`

□ Marges

- Définir une marge autour de chaque élément pour aérer le contenu de la page
- **Margin:valeur**: Cette valeur s'applique à **toute** les marges (haute, droite, basse et gauche) dans le sens des aiguilles d'une montre
- Il est possible de définir les marges individuellement
- **Ex.**
 - **margin-top**: 3px; /* marge **haute**
 - **margin-right**: 3px; /* marge **droite**
 - **margin-bottom**: 2px; /* marge **basse**
 - **margin-left**: 2px; /* marge **gauche**

Validation

- Service de validation du code CSS en ligne:
jigsaw.w3.org/css-validator
- La vérification du code peut se faire, soit par:
 - Adresse URI
 - Chargement du fichier CSS
 - Saisie directe du code CSS