

HTML CSS Javascript

CSS

Objectif

- Comprendre le CSS
- Mettre en œuvre les feuilles de style

Introduction

- **Cascading Style Sheets (CSS)** est un langage permettant de décrire la présentation de pages HTML. La cascade représente l'héritage de propriétés par les éléments sous-jacents.
- Quel usage ?
 - **Mette en forme** les pages web
 - Couleurs, bordures, arrière-plan, alignement, typographie, effets...
 - **Mettre en page** les pages web
 - Positions, marges, espacements, flexibilité
- Participe à la séparation de la forme et de contenu

La déclaration d'un style

- Un style est déclaré pour le binôme **property : value**
- Exemple
 - *background-color: blue;*
 - *font-family: arial;*
 - *border-size: 12px;*
 - *border-style: solid;*

Règles CSS

- On utilise un sélecteur pour appliquer des règles d’affichage `Sélecteur { bloc de déclarations }`
 - Sélecteur : permet de sélectionner les éléments HTML que l’on veut styliser
 - Bloc de déclarations : propriétés des éléments à modifier avec la valeur à appliquer
 - Exemple :

```
h1 { color: blue; }  
/* La propriété color de tous les h1 prendra la valeur bleue */
```
- Bloc de déclaration :
 - Liste de propriétés et valeurs sous la forme


```
propriété1 : valeur1;  
propriété2 : valeur2;  
...
```
- Partiellement sensible à la casse (valeurs de class et d’id)

Où écrire le code CSS ?

Trois manières de déclarer du CSS :


- Les **styles locaux**

- Attribut style des balises

```
<!-- Un style local -->  
<p style="background-color: red;">Un texte rouge</p>
```

- Les **styles internes**

- Balises <style>...</style> de la section <head>

```
<head>  
  <style type="text/css">  
    /* Un page entièrement rouge */  
    body {background-color: red;}  
  </style>  
  <title>My music</title>  
</head>
```

Où écrire le code CSS ?

Trois manières de déclarer du CSS :

- Les **styles locaux**

- Attribut style des balises

```
<!-- Un style local -->  
<p style="background-color: red;">Un texte rouge</p>
```

- Les **styles internes**

- Balises <style>...</style> de la section <head>

```
<head>  
  <style type="text/css">  
    /* Un page entièrement rouge */  
    body {background-color: red;}  
  </style>  
  <title>My music</title>  
</head>
```

- La **feuille de style externe** (à privilégier)

- Fichier texte indépendant, avec l'extension **.css**

```
<head>  
  <!-- associer une feuille de style externe -->  
  <link rel="stylesheet" type="text/css" href="mon_style.css" />  
  <title>My music</title>  
</head>
```

Où écrire le code CSS ?

La syntaxe à respecter :

Contenu HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Une page avec du CSS</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Du joli CSS</h1>
  <h2>Un sous-titre qu'il est beau</h2>
  <p>Un paragraphe magnifique</p>
</body>
</html>
```

Contenu CSS

```
p {
  color: blue;
}

h1, h3, h5 {
  background-color: #0000FF;
  color: white;
}
```

Du joli CSS

Un sous-titre qu'il est beau

Un paragraphe magnifique

Les sélecteurs

- Comment appliquer les styles à mes balises ?

- **Sélecteur universel**

- `* { color : #ff0000 }`

- **Sélecteur d'instances de balise**

- `body {
 background-image:url("image/fond.png");
 background-position:center;
 background-repeat:no-repeat;
} /* Image de fond centrée et non répétée sur les pages du site`
 - `p { text-align:justify; } // Tous les paragraphes sont justifiés`

- **Sélection multiple**

- `h1, h2, h3, h4 { font-family:verdana; } /* Tous les titres utiliseront la police Verdana */`

sélecteur
`p.important {`
propriété `font-weight:` *valeur* `bold;`
règle `border: 2px solid red;`
`}`

Les sélecteurs

- Comment appliquer les styles à mes balises ?
 - Sur plusieurs éléments par l'utilisation de **classes CSS**
 - En HTML, possibilité d'ajouter un attribut « class » aux balises, `<p class="ma_classe">Un paragraphe</p>`
 - Toutes les balises qui portent la classe "ma_classe" seront concernées par la règle définie
 - `.ma_classe { color : red; }`
 - Sur une balise en particulier en utilisant l'**identifiant**
 - En HTML, `<p id="un_id">Un deuxième paragraphe</p>`
 - `#un_id { background-color: blue; }`

Les sélecteurs

- Il est possible de regrouper plusieurs sélecteurs pour qu'ils suivent la même règle
 - Les différents sélecteurs sont séparés par des virgules
`h1, #mon_id, .ma_classe { color : yellow; }`

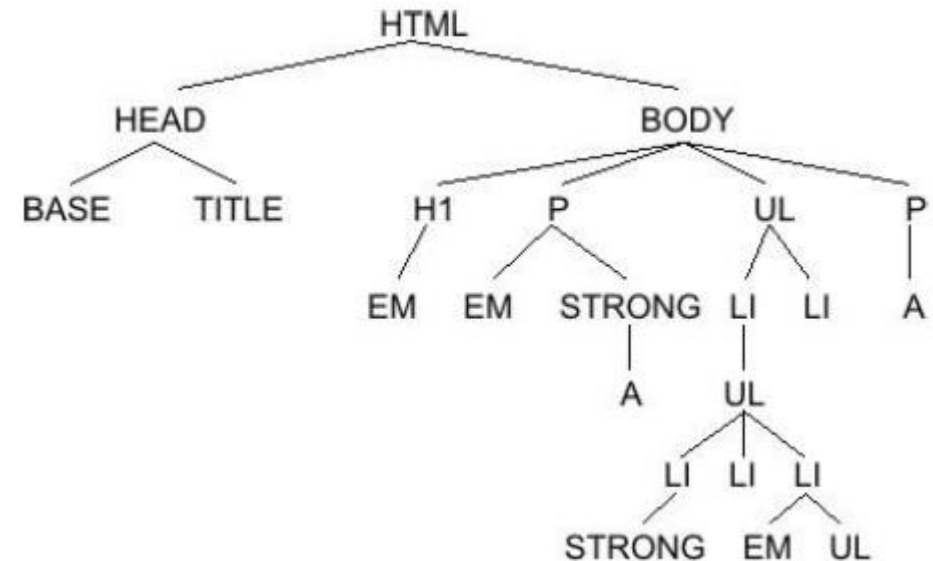
Différentes notations

- Certaines propriétés permettent de définir plusieurs attributs
- Exemple

```
border: 10px;  
est équivalent à  
border-top: 10 px;  
border-right: 10 px;  
border-bottom: 10px;  
border-left: 10px;  
est équivalent à  
border: 10px 10px 10px 10px; /* Notation courte */  
      top  right bottom left
```

Document Object Model (DOM)

- Les éléments HTML décrivent une structure arborescente appelée Document Object Model, ou DOM
- Cette arborescence permet de distinguer les éléments entre eux
 - rôle de parent
 - rôle d'enfant
 - rôle de frère (sibling)



Héritage des propriétés

- Les styles s'appliquent de la racine du DOM (html) vers ses feuilles

```
<!DOCTYPE html>
<html lang="fr">
  <head> <style>body {color: blue;}</style> </head>
  <body>
    <h1>Titre en bleu par héritage</h1>
  </body>
</html>
```

- Le titre h1 hérite des valeurs de propriétés définies par son parent : body
Le titre s'affiche donc en bleu
- Certaines propriétés ne s'héritent pas (règles de positionnement par exemple)

Héritage des propriétés

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style/style.css" />
    <title>Exemple héritage CSS</title>
    <style>
      .grand { font-size: 32px; }
      .gras { font-weight: bold; }
      .rouge { color: red; }
    </style>
  </head>
  <body>
    <span class="grand">Je suis grand,
      <span class="gras">gras
        <span class="rouge">et rouge</span>
      </span>
    </span>
  </body>
</html>
```

Je suis grand, **gras et rouge**

Positionnement des éléments

- Par défaut : les éléments se positionnent dans le flux de la page (position : static)

Positionnement des éléments

- Il existe des alternatives pour positionner nos éléments à l'écran :

Positionnement des éléments

- Il existe des alternatives pour positionner nos éléments à l'écran :


```
.position_absolute {  
  position: absolute;  
  right : 15px;  
  top : 0px;  
}
```

Du joli CSS

Un sous-titre qu'il est beau

Un paragraphe magnifique

Un texte positionné de manière "relative"
apparaît à sa position normale dans le flux, mais avec des
informations supplémentaires sur les marges



Un texte positionné de manière "absolute"
apparaît à une position donnée par rapport à son parent

Un texte positionné de manière "fixed"
apparaît à une position donnée sur l'écran, indépendant
du flux d'affichage sur la page

Positionnement des éléments

- Il existe des alternatives pour positionner nos éléments à l'écran :

```
.position_relative {  
  position : relative;  
  left : 15px;  
  top : 12px;  
}
```

Du joli CSS

Un sous-titre qu'il est beau

Un paragraphe magnifique



Un texte positionné de manière "relative"
apparaît à sa position normale dans le flux, mais avec des
informations supplémentaires sur les marges

Un texte positionné de manière "absolute"
apparaît à une position donnée par rapport à son parent

Un texte positionné de manière "fixed"
apparaît à une position donnée sur l'écran, indépendant
du flux d'affichage sur la page

Positionnement des éléments

- Il existe des alternatives pour positionner nos éléments à l'écran :

```
.position_fixed {  
  position : fixed;  
  right : 10px;  
  bottom : 10px;  
}
```

Du joli CSS

Un sous-titre qu'il est beau

Un paragraphe magnifique

Un texte positionné de manière "relative"
apparaît à sa position normale dans le flux, mais avec des
informations supplémentaires sur les marges

Un texte positionné de manière "absolute"
apparaît à une position donnée par rapport à son parent



Un texte positionné de manière "fixed"
apparaît à une position donnée sur l'écran, indépendant
du flux d'affichage sur la page

Positionnement des éléments

- Il existe des alternatives pour positionner nos éléments à l'écran :

```
.position_fixed {  
  position : fixed;  
  right : 10px;  
  bottom : 10px;  
  z-index: 2;  
}
```

```
.squatter {  
  position : fixed;  
  bottom : 15px;  
  right : 250px;  
  z-index: 1;  
}
```

Du joli CSS

Un sous-titre qu'il est beau

Un paragraphe magnifique

Un texte positionné de manière "absolute"
apparaît à une position donnée par rapport à son parent

Un texte positionné de manière "relative"
apparaît à sa position normale dans le flux, mais avec des
informations supplémentaires sur les marges

Deux éléments qui se sur-
êtent réglés avec la propriété

Un texte positionné de manière "fixed"
apparaît à une position donnée sur l'écran, indépendamment
du flux d'affichage sur la page

Positionnement des éléments

- Il existe des alternatives pour positionner nos éléments à l'écran :

```
.position_fixed {  
  position : fixed;  
  right : 10px;  
  bottom : 10px;  
  z-index: 1;  
}
```

```
.squatter {  
  position : fixed;  
  bottom : 15px;  
  right : 250px;  
  z-index: 2;  
}
```

Du joli CSS

Un sous-titre qu'il est beau

Un paragraphe magnifique

Un texte positionné de manière "absolute"
apparaît à une position donnée par rapport à son parent

Un texte positionné de manière "relative"
apparaît à sa position normale dans le flux, mais avec des
informations supplémentaires sur les marges

Deux éléments qui se superposent peuvent
être réglés avec la propriété "z-index"

Un élément positionné de manière "fixed"
à une position donnée sur l'écran, indépendamment
de son positionnement sur la page

Positionnement des éléments

- Par défaut, les éléments sont positionnées dans leur ordre d'apparition dans le code source HTML
- Le positionnement de chaque élément dépend de son mode d'affichage (propriété display)
 - Un élément de type block se place à la verticale
 - Un élément de type inline se place à l'horizontale
- Un élément peut être masqué avec la valeur de none de l'attribut display

Positionnement des éléments

- La propriété display
 - display: block
 - Affiche les éléments les uns sous les autres
 - L'élément peut être dimensionné (possède une width et une height)
 - display: inline
 - Affiche les éléments les uns à côté des autres
 - L'élément s'adapte à la taille de son contenu
 - display: inline-block
 - Affiche les éléments les uns à côté des autres
 - L'élément peut être dimensionné

Positionnement des éléments

`display: inline;`



~~Dimension
et marge~~

`display:
inline-block;`



Dimension
et marge

`display: block;`



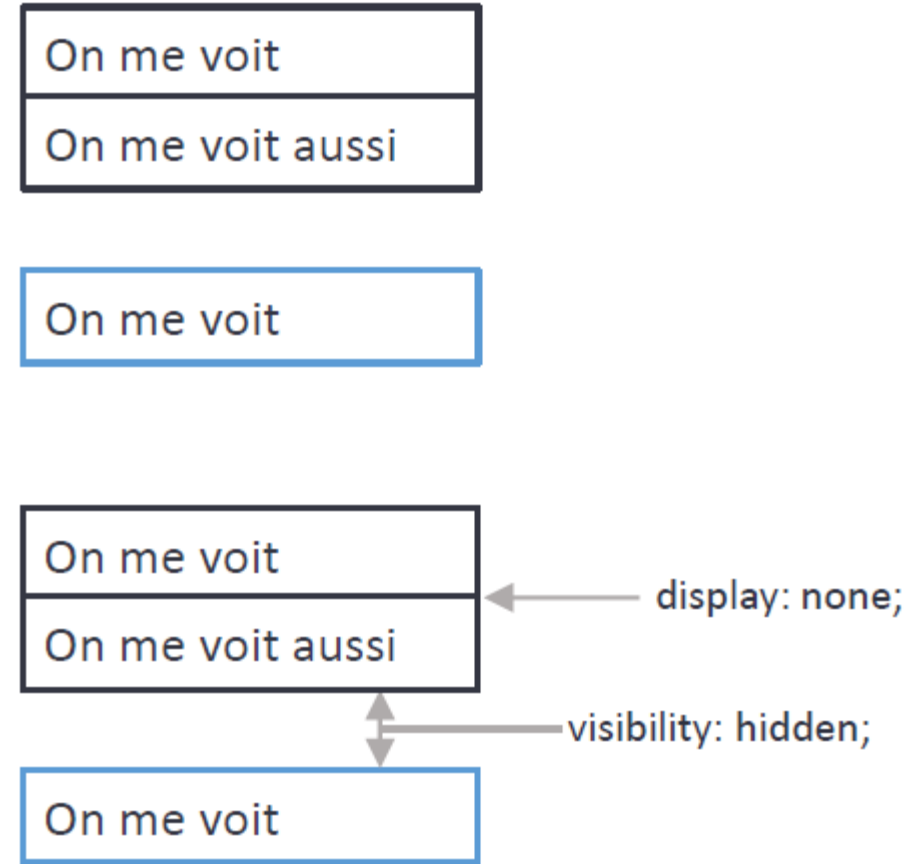
Dimension
et marge

Positionnement des éléments

- L'élément `display` nous permet ainsi de contrôler l'affichage des éléments sur la page.
 - `display: none;`
 - spécifie que l'élément ne sera pas affiché, le retirant simplement de la page sans lui accorder d'espace
- La propriété `visibility` détermine si un élément est caché ou visible
 - En mode `hidden`, l'élément est caché mais sa place sur la page est quand même réservée
 - `visibility : hidden | visible;`

Positionnement des éléments

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Exemple display & visibility</title>
    <style>
      div { width: 150px; height: 40px; border: 1px solid black; }
      .none { display: none; }
      .hidden { visibility: hidden; }
    </style>
  </head>
  <body>
    <div>On me voit</div>
    <div class="none">On ne me voit pas</div>
    <div>On me voit aussi</div>
    <div class="hidden">
      On ne me voit pas mais je réserve ma place
    </div>
    <div>On me voit</div>
  </body>
</html>
```



Le texte

- Couleur : color
- Décoration
 - text-decoration: underline | overline | line-through | none;
- Transformation
 - text-transform: capitalize | uppercase | lowercase;
- Indentation
 - text-indent: 15px;
- Espace entre les lettres
 - letter-spacing: 3px;
- Espace entre les mots
 - word-spacing: 10px;
- Interligne
 - line-height: 15px;

Le texte

- Alignement horizontal
 - `text-align: left | center | right | justify;`
- Alignement vertical
 - `vertical-align: baseline | sub | super | top | middle | bottom;`
- Direction du texte
 - `direction: ltr | rtl`
- Longueur et hauteur
 - `width: 40px;`
 - `height: 100px;`

Les listes

- Sorte de marqueurs
 - `list-style-type: disc | circle | square | decimal | decimal-leading-zero | upper-roman | lower-roman | upper-alpha | lower-alpha`
- Marqueur graphique
 - `list-style-image: url("chemin/vers/une/image.jpg");`
- Retrait
 - `list-style-position: outside | inside;`

Les arrière-plans

- Une couleur d'arrière plan
 - `background-color: ...;`
- Insertion d'une image d'arrière-plan
 - `background-image: url("chemin/vers/une/image.png");`
- Répétition de l'arrière plan
 - `background-repeat: repeat | repeat-x | repeat-y | no-repeat`
- Positionnement de l'image
 - `background-position: valeur;`

Liens hypertextes et pseudo-éléments

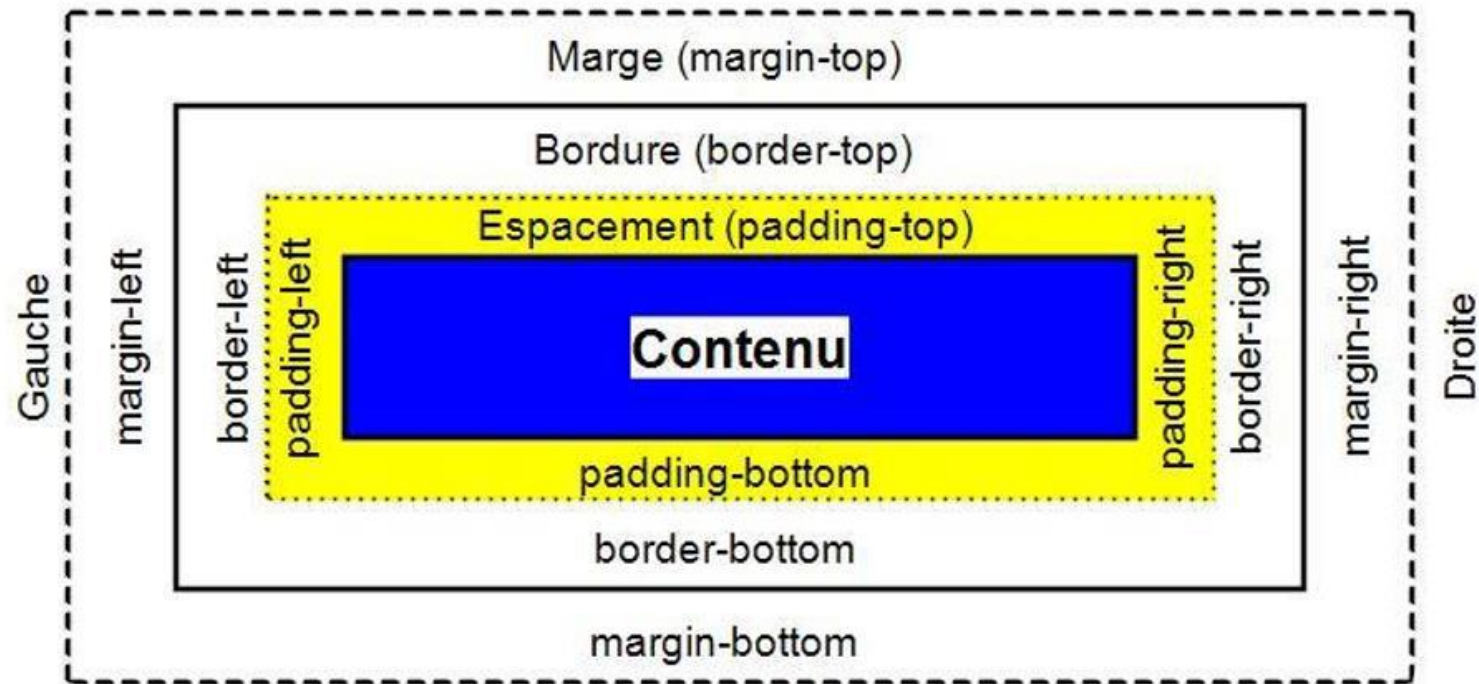
- Lien non visité : **a:link**
- Lien visité : **a:visited**
- Lien activé par le clic de souris : **a:active**
- Lien activé par une touche de clavier : **a:focus**
- Lien survolé : **a:hover**
- Style différent pour le premier élément enfant de l'élément : **xxx:first-child**
- Style différent pour le dernier élément enfant de l'élément : **xxx:last-child**
- Style différent pour un élément enfant sur deux : **xxx:nth-child(odd)** et **xxx:nth-child(even)**
- Première lettre d'un texte : **xxx:first-letter**

Liens hypertextes et pseudo-éléments

- Sélecteur descendant
 - `.articles a { background-color: blue; }`
attribue un couleur bleue à toutes les balises a contenues dans des balises de classe « articles »
- Sélecteur enfant
 - `p > a`
attribue un style aux balises a directement enfant de p
- Sélecteur adjacent
 - `h1 + h2`
attribut un style aux h2 suivant directement un h1

Propriétés des boîtes

- Un élément boîte est défini par le W3C comme étant une zone rectangulaire constituée ainsi :



Les bordures

- Il est possible de personnaliser les bordures d'un élément avec plusieurs propriétés.
 - border-color
 - border-width
 - border-style
- Toutes les propriétés des bordures sont déclinées pour n'affecter qu'un seul côté
 - border-color : border-top-color, border-left-color, border-bottom-color, border-right-color
 - ...

HTML CSS Javascript

CSS