

CSS

Prof. Ilyas El jaafari

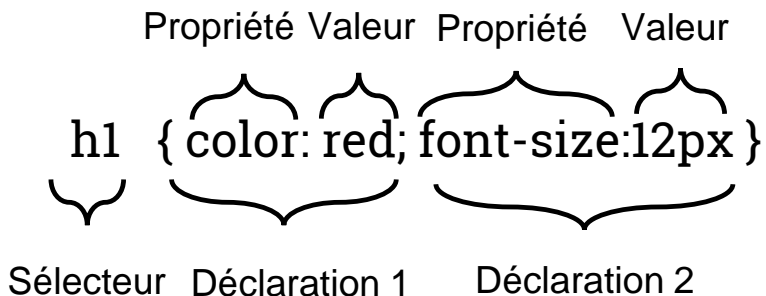
SMI/IGE 2018

Introduction

- CSS est l'acronyme de Cascading Style Sheets.
- CSS décrit comment les éléments HTML doivent être affichés à l'écran, sur papier ou sur un autre support.
- CSS économise beaucoup de travail. Il peut contrôler la mise en page de plusieurs pages Web en même temps
- Les feuilles de style externes sont stockées dans des fichiers CSS (.css)
- CSS permet de définir les styles de vos pages Web, notamment la conception, la disposition et les variations d'affichage pour différents appareils et tailles d'écran.
- Le World Wide Web Consortium (W3C) a créé CSS pour supprimer la mise en forme du style de la page HTML.
- CSS économise beaucoup de travail! Les définitions de style sont normalement enregistrées dans des fichiers .css externes. Avec une feuille de style externe, vous pouvez changer l'apparence d'un site Web entier en changeant un seul fichier!

Syntaxe et sélecteurs CSS

Syntaxe CSS



- Le sélecteur pointe sur l'élément HTML que vous souhaitez styler.
- Le bloc de déclaration contient une ou plusieurs déclarations séparées par des points-virgules.
- Chaque déclaration inclut un nom de propriété CSS et une valeur, séparés par deux points.
- Une déclaration CSS se termine toujours par un point-virgule et les blocs de déclaration sont entourés d'accolades.

Exemple :

```
p {  
    color: red;  
    text-align: center;  
}
```

Syntaxe et sélecteurs CSS

Sélecteurs CSS

Les sélecteurs CSS permettent de "rechercher" (ou de sélectionner) des éléments HTML en fonction de leur nom, id, classe, attribut, etc. :

❑ Le sélecteur d'élément

- Le sélecteur d'élément sélectionne les éléments en fonction du nom de l'élément.
- Vous pouvez sélectionner tous les éléments <p> sur une page comme ceci (dans ce cas, tous les éléments <p> seront alignés au centre, avec une couleur de texte rouge):

```
p {  
    text-align: center;  
    color: red;  
    /* ceci est un commentaire */  
}
```

- Les commentaires sont utilisés pour expliquer le code et peuvent vous aider lorsque vous modifiez le code source à une date ultérieure. Un commentaire CSS commence par / * et se termine par * /.

Syntaxe et sélecteurs CSS

Sélecteurs CSS

❑ Le sélecteur id

- Le sélecteur id utilise l'attribut id d'un élément HTML pour sélectionner un élément spécifique.
- id = L'identifiant d'un élément et il doit être unique dans une page, le sélecteur d'identifiant sert donc à sélectionner un élément unique!

Exemple: élément HTML avec un id `<p id="para1">mon text</p>`

- En CSS, pour sélectionner un élément avec un identifiant spécifique, écrivez un caractère dièse (#), suivi de l'identifiant de l'élément.
- La règle de style ci-dessous sera appliquée à l'élément HTML avec id = "para1":

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

Syntaxe et sélecteurs CSS

Sélecteurs CSS

❑ Le sélecteur class

- Le sélecteur class sélectionne les éléments avec un attribut de classe spécifique.
- En CSS, pour sélectionner des éléments avec une classe spécifique, écrivez un caractère point (.), Suivi du nom de la classe.
- Dans l'exemple ci-dessous, tous les éléments HTML avec class = "center" seront en rouge et centrés:

```
.center {  
    text-align: center;  
    color: red;  
}
```

- Vous pouvez également spécifier que seuls des éléments HTML spécifiques doivent être affectés par une classe.
- Dans l'exemple ci-dessous, seuls les éléments <p> avec class = "center" seront alignés au centre:

```
p.center {  
    text-align: center;  
    color: red;  
}
```

- Les éléments HTML peuvent également faire référence à plus d'une classe.

```
<p class="center large">This paragraph refers to two classes.</p>
```

Syntaxe et sélecteurs CSS

Sélecteurs CSS

- Si vous avez des éléments avec les mêmes définitions de style, comme ceci:

```
h1 {  
  text-align: center;  
  color: red;  
}  
h2 {  
  text-align: center;  
  color: red;  
}  
p {  
  text-align: center;  
  color: red;  
}
```

- Il sera préférable de regrouper les sélecteurs pour minimiser le code.
- Pour regrouper les sélecteurs, séparez chaque sélecteur par une virgule.

Exemple: sélecteurs regroupés

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

Comment utiliser le CSS

- Lorsqu'un navigateur lit une feuille de style, il formate le document HTML en fonction des informations contenues dans la feuille de style.
- Il y a trois façons d'insérer une feuille de style:
 1. Style en ligne
 2. Feuille de style interne
 3. Feuille de style externe

❑ Style en ligne

Un style en ligne peut être utilisé pour appliquer un style unique à un seul élément.

Pour utiliser les styles en ligne, ajoutez l'attribut style à l'élément concerné. L'attribut style peut contenir n'importe quelle propriété CSS.

L'exemple ci-dessous montre comment changer la couleur et la marge gauche d'un élément <h1>:

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```


Comment utiliser le CSS

❑ Feuille de style interne

- Une feuille de style interne peut être utilisée si une seule page a un style unique.
- Les styles internes sont définis dans l'élément <style>, dans la section <head> d'une page HTML:

```
<html>
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
<body>...</body>
</html>
```

Comment utiliser le CSS

❑ Feuille de style externe

- Avec une feuille de style externe, vous pouvez changer l'apparence d'un site Web entier en ne modifiant qu'un seul fichier!
- Chaque page doit inclure une référence au fichier de feuille de style externe à l'intérieur de l'élément <link>. L'élément <link> va à l'intérieur de la section <head>:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

- Une feuille de style externe peut être écrite dans n'importe quel éditeur de texte. Le fichier ne doit contenir aucune balise HTML. Le fichier de feuille de style doit être enregistré avec une extension .css.
- Voici à quoi ressemble "mystyle.css":

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```

Couleurs CSS

❑ Noms de couleur

- En HTML, une couleur peut être spécifiée en utilisant un nom de couleur:



❑ Couleur de fond

- Vous pouvez définir la couleur d'arrière-plan des éléments HTML par la propriété background-color:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Hello World

Lorem ipsum...

Couleurs CSS

❑ Couleur du texte

- Vous pouvez définir la couleur du texte par la propriété color:

Hello World

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Lorem ipsum...

Ut wisi enim...

❑ Couleur de la bordure

- Vous pouvez définir la couleur des bordures par la propriété border:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>  
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>  
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Hello World

Hello World

Hello World

Couleurs CSS

❑ Valeurs de couleur

`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgb(60, 179, 113)`

`rgb(238, 130, 238)`

`rgb(255, 165, 0)`

`rgb(106, 90, 205)`

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
```

```
<h1 style="background-color:#ff6347;">...</h1>
```

```
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
```

Arrière-plans CSS

Les propriétés background de CSS permettent de définir les effets d'arrière-plan des éléments.

Propriété	Description
background	Définit toutes les propriétés d'arrière-plan dans une déclaration
background-attachment	Définit si une image de fond est fixe ou défile avec le reste de la page
background-clip	Spécifie la zone de peinture de l'arrière-plan
background-color	Définit la couleur d'arrière-plan d'un élément
background-image	Définit l'image d'arrière-plan d'un élément
background-position	Définit la position de départ d'une image d'arrière-plan
background-repeat	Définit comment une image d'arrière-plan sera répétée
background-size	Spécifie la taille des images d'arrière-plan

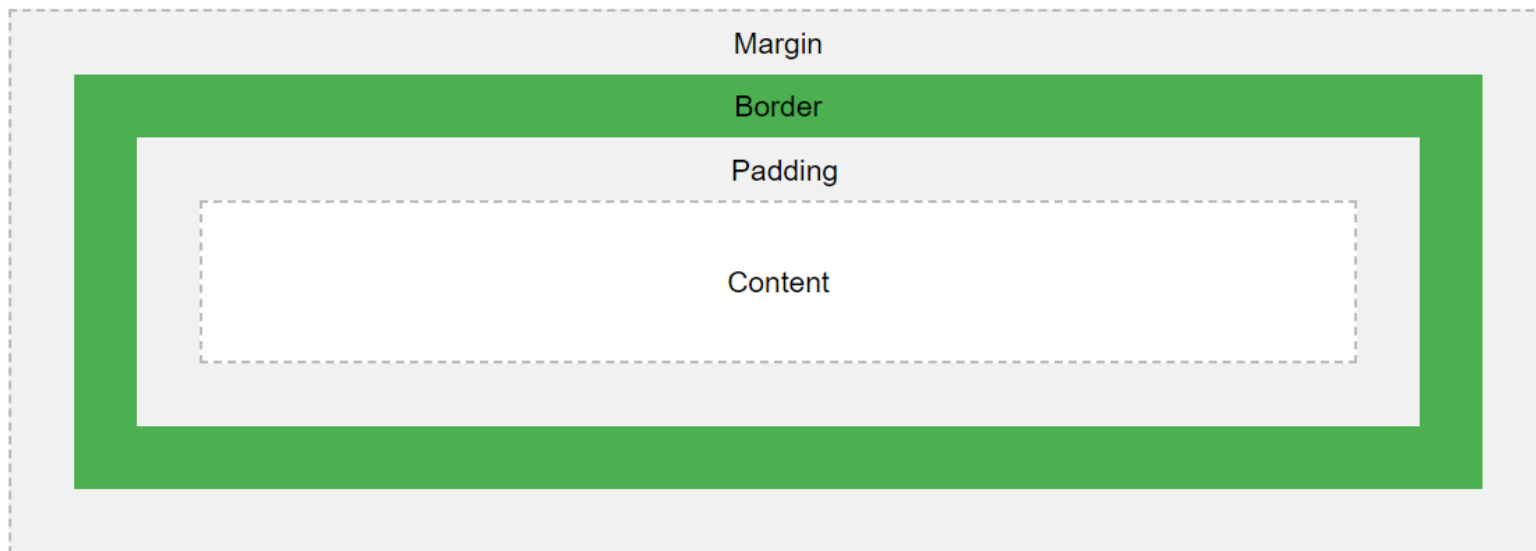
Exemple:

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}  
  
body {  
    background: #ffffff url("img_tree.png") no-repeat  
    right top;  
}
```

Modèle de boîte CSS

Tous les éléments HTML peuvent être considérés comme des boîtes. En CSS, le terme "modèle de boîte" est utilisé quand on parle de design et de mise en page.

Le modèle de boîte CSS est essentiellement une boîte qui entoure chaque élément HTML. Il comprend: les marges, les bordures, le remplissage et le contenu réel. L'image ci-dessous illustre le modèle de boîte:



Bordures (border) CSS

Les propriétés border de CSS vous permettent de spécifier le style, la largeur et la couleur de la bordure d'un élément.

propriété	Description
border	Définit toutes les propriétés de la bordure dans une déclaration
border-bottom	Définit toutes les propriétés de la bordure inférieure dans une déclaration
border-bottom-color	Définit la couleur de la bordure inférieure
border-bottom-style	Définit le style de la bordure inférieure
border-bottom-width	Définit la largeur de la bordure inférieure
border-color	Définit la couleur des quatre bordures
border-radius	Définit les quatre propriétés de bordure - * - pour les coins arrondis
border-style	Définit le style des quatre bordures
border-width	Définit la largeur des quatre bordures

A la place bottom on peut avoir aussi top, left, right pour les bordures supérieur, gauche et droite

Exemples :

```
p {  
  border-left: 6px solid red;  
  background-color: lightgrey;  
}  
  
p {  
  border-top-style: dotted;  
  border-right-style: solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;  
}  
  
p {  
  border: 5px solid red;  
}
```


Marges (margin) CSS

- Les propriétés margin CSS sont utilisées pour créer un espace autour des éléments, en dehors de toute bordure définie.
- Avec CSS, vous avez un contrôle total sur les marges. Il existe des propriétés permettant de définir la marge de chaque côté d'un élément (haut, droite, bas et gauche).

- CSS possède des propriétés permettant de spécifier la marge de chaque côté d'un élément:

margin-top
margin-right
margin-bottom
margin-left

- Toutes les propriétés de la marge peuvent avoir les valeurs suivantes:

auto - le navigateur calcule la marge

longueur - spécifie une marge en px, pt, cm, etc.

% - spécifie une marge en% de la largeur de l'élément conteneur

inherit - spécifie que la marge doit être héritée de l'élément parent

Exemple:

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

Exemple: la marge gauche de l'élément p est 100px hérité de l'élément père (div)

```
<div>  
<p class="ex1"> Ce paragraphe a une marge gauche héritée (de l'élément div) </p>  
</div>
```

```
div {  
    border: 1px solid red;  
    margin-left: 100px;  
}  
p.ex1 {  
    margin-left: inherit;  
}
```

Remplissage (Padding) CSS

- Les propriétés padding de CSS permettent de créer un espace autour du contenu d'un élément, à l'intérieur de tout bordure.
- Avec CSS, vous avez un contrôle total sur le remplissage. Il existe des propriétés permettant de définir le remplissage pour chaque côté d'un élément (haut, droite, bas et gauche).
- CSS possède des propriétés permettant de spécifier le remplissage de chaque côté d'un élément:

padding-top
padding-right
padding-bottom
padding-left

- Toutes les propriétés de remplissage peuvent avoir les valeurs suivantes:
longueur - spécifie un remplissage en px, pt, cm, etc.
% - spécifie un remplissage en% de la largeur de l'élément conteneur
inherit - spécifie que le remplissage doit être hérité de l'élément parent

Exemple 1:

```
div {  
  padding-top: 50px;  
  padding-right: 30px;  
  padding-bottom: 50px;  
  padding-left: 80px;  
}
```

Exemple 2:

```
div {  
  padding: 25px 50px 75px 100px;  
  /*top    right  bottom  left*/  
}
```

hauteur et largeur CSS

- Les propriétés height et width de CSS permettent de définir la hauteur et la largeur d'un élément.
- La hauteur et la largeur peuvent être définies sur auto (valeur par défaut. Indique que le navigateur calcule la hauteur et la largeur) ou spécifiées en valeurs de longueur, telles que px, cm, etc. ou en pourcentage (%) du bloc qui le contient.

Exemple:

```
div {  
    height: 200px;  
    width: 50%;  
    background-color: powderblue;  
}
```

Propriété	Description
height	Définit la hauteur d'un élément
max-height	Définit la hauteur maximale d'un élément
max-width	Définit la largeur maximale d'un élément
min-height	Définit la hauteur minimale d'un élément
min-width	Définit la largeur minimale d'un élément
width	Définit la largeur d'un élément

Texte CSS

```
body {  
  color: blue;  
}
```

```
h1 {  
  text-align: center;  
}
```

```
h2 {  
  text-align: left;  
}
```

```
h3 {  
  text-align: right;  
}
```

```
div {  
  text-align: justify;  
}
```

```
h1 {  
  text-decoration: overline;  
}
```

```
h2 {  
  text-decoration: line-through;  
}
```

```
h3 {  
  text-decoration: underline;  
}
```

```
h1 {  
  text-shadow: 3px 2px red;  
}
```

```
p.uppercase {  
  text-transform: uppercase;  
}
```

```
p.lowercase {  
  text-transform: lowercase;  
}
```

```
p.capitalize {  
  text-transform: capitalize;  
}
```

Police (font) CSS

```
p {  
  font-family: "Times New Roman";  
}  
  
p.normal {  
  font-weight: normal;  
}  
  
p.thick {  
  font-weight: bold;  
}  
  
p.normal {  
  font-style: normal;  
}  
  
p.italic {  
  font-style: italic;  
}  
  
p.oblique {  
  font-style: oblique;  
}  
  
h1 {  
  font-size: 40px;  
}  
  
h1 {  
  font-size: 2.5em; /* 40px/16=2.5em */  
}
```

Liens CSS

- Les liens peuvent être stylés avec n'importe quelle propriété CSS (couleur, famille de polices, arrière-plan, etc.).

Text Link

Text Link

Link Button

Link Button

- Les liens peuvent être stylés différemment en fonction de l'état dans lequel ils se trouvent.

Les quatre états de liens sont:

a: link- un lien normal non visité

a: visited - un lien que l'utilisateur a visité

a: hover - un lien lorsque l'utilisateur passe la souris dessus

a: active - un lien au moment où il est cliqué

```
a:link, a:visited {  
    background-color: #f44336;  
    color: white;  
    padding: 14px 25px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
}
```

```
a:hover, a:active {  
    background-color: red;  
}
```

```
/* unvisited link */  
a:link {  
    color: red;  
}
```

```
/* visited link */  
a:visited {  
    color: green;  
}
```

```
/* mouse over link */  
a:hover {  
    color: hotpink;  
}
```

```
/* selected link */  
a:active {  
    color: blue;  
}
```

Position CSS

- La propriété position spécifie le type de méthode de positionnement utilisé pour un élément (static, relative, fixed, absolute or sticky).

static (par défaut): Les éléments positionnés statiques ne sont pas affectés par les propriétés top, bottom, left et right.

relative: Si vous définissez les propriétés haut, droit, bas et gauche d'un élément relativement positionné, celui-ci sera ajusté pour s'éloigner de sa position normale. Les autres contenus ne seront pas ajustés pour s'insérer dans les espaces laissés par l'élément.

fixed: Un élément avec position: fixed; est positionné par rapport à la fenêtre d'affichage, ce qui signifie qu'il reste toujours au même endroit même si la page est défilée. Les propriétés top, right, bottom et left sont utilisées pour positionner l'élément.

absolute: Un élément avec position: absolute; est positionné par rapport à son père « positionné » le plus proche (au lieu de positionné par rapport à la fenêtre d'affichage, comme fixed). Toutefois; Si un élément absolu positionné n'a pas de père positionnés, il utilise le corps du document et se déplace avec le défilement de page.

sticky: Un élément avec position: sticky; est positionné en fonction de la position de défilement de l'utilisateur.

Position CSS

```
div {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid #73AD21;  
}
```

```
div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

```
div.sticky {  
  position: -webkit-sticky; /* Safari */  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```

Si vous voulez tester ces exemples et voir les différences entre ces propriétés consulter:

https://www.w3schools.com/css/css_positioning.asp

Débordement (overflow) CSS

- La propriété overflow de CSS contrôle ce qui arrive au contenu trop volumineux pour le mettre dans une zone.

visible - Par défaut. Le débordement n'est pas coupé. Il rend en dehors de la boîte de l'élément

hidden - Le débordement est tronqué et le reste du contenu sera invisible.

scroll - Le débordement est tronqué, mais une barre de défilement est ajoutée pour afficher le reste du contenu.

auto - Si le débordement est tronqué, une barre de défilement doit être ajoutée pour afficher le reste du contenu.

```
div {  
  width: 200px;  
  height: 50px;  
  background-color: #eee;  
  overflow: visible;  
}
```

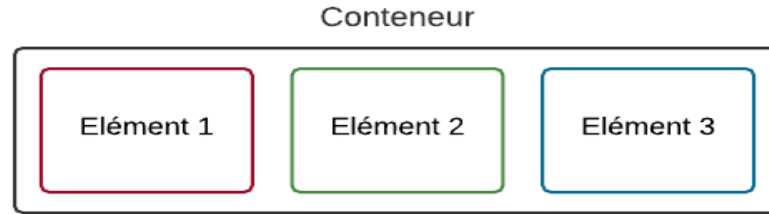
```
div {  
  overflow: scroll;  
}
```

```
div {  
  overflow: hidden;  
}
```

```
div {  
  overflow-x: hidden; /* Hide horizontal scrollbar */  
  overflow-y: scroll; /* Add vertical scrollbar */  
}
```

Flexbox CSS

- Le principe de la mise en page avec Flexbox est simple : vous définissez un conteneur, et à l'intérieur vous placez plusieurs éléments.



- Le conteneur est une balise HTML, et les éléments sont d'autres balises HTML à l'intérieur

HTML

```
<div id="conteneur">
  <div class="element">Elément 1</div>
  <div class="element">Elément 2</div>
  <div class="element">Elément 3</div>
</div>
```

CSS

```
#conteneur
{
  display: flex;
}
```