

TECHNOLOGIE et PROGRAMMATION WEB

2020-2021



marwahentati@yahoo.fr

CSS Cascading Style Sheets



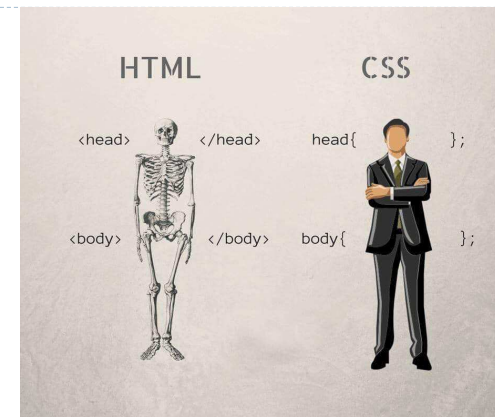
2

Introduction

- ▶ Les feuilles de style **CSS** sont un complément direct à HTML. Il s'agit d'un langage pour définir les propriétés de format de commandes HTML distinctes.
- ▶ CSS permet de spécifier dans une page web les polices de caractères à utiliser, leurs attributs d'affichage (gras, italique, souligné, taille, couleur.), la couleur ou l'image de fond, les formats de puces.
- ▶ CSS est une solution pour séparer le contenu de la forme.
 - ▶ Séparer le travail de la rédaction de celui de la présentation.
- ▶ CSS assure la cohérence graphique d'un site.
 - ▶ Plusieurs pages peuvent partager la même feuille de style.

3

Introduction



4

Historique

- ▶ Les différentes versions officielles de ce standard sont les suivantes:
 - ▶ CSS1 (level 1), recommandation depuis Décembre 1996.
 - ▶ CSS2 (level 2), recommandation depuis Mai 1998.
 - ▶ CSS3 (level 3), recommandation depuis 29 Septembre 2011
- ▶ CSS1 et CSS2 sont supportés par la plupart des navigateurs web (Microsoft Internet Explorer, Mozilla,...).
- ▶ CSS2 apporte de nombreuses fonctionnalités, notamment dans les réglages d'affichage graphique, ainsi que la gestion du son.
- ▶ CSS3 propose un support international encore mieux amélioré avec une gestion de l'affichage des langues arabes et chinoises ainsi que l'intégration des dessins vectoriels.

▶ 5

Syntaxe

- ▶ Une règle de style des CSS est composée de:
 - ▶ Un **sélecteur** qui désigne à quoi les définitions qui suivent s'appliquent.
 - ▶ Au moins une déclaration qui présente les propriétés CSS modifiant la format du document HTML.

h1 {color: #0000FF; font-family: Arial; font-size: 12px;}
 sélecteur déclaration (composée d'une propriété et de sa Valeur)

- ▶ **h1** est le sélecteur qui définit le style qui sera appliqué à la balise html <h1>, la déclaration attribue la couleur bleue et la police Arial de taille 12 au titre.

▶ 6

Syntaxe

- ▶ La **déclaration** du style proprement dite est placée entre des accolades { }
- ▶ La propriété et sa valeur sont séparées l'une de l'autre par « : »
- ▶ Les expressions complètes contenant propriété et valeur sont séparées par « ; »
 balise { propriété de style: valeur; propriété de style: valeur; }
- ▶ Il est possible de grouper plusieurs sélecteurs ayant une règle de style identique, séparés par une virgule :
 h1, h2, h3 { color: red }
- ▶ Les lignes de commentaires dans les fichiers de style CSS commencent par "/*"

▶ 7

Appliquer un style

- ▶ Les formats CSS peuvent être définis soit dans la balise d'ouverture d'un élément HTML ou bien dans un passage de style global, dans un fichier CSS externe.
 - ▶ Appel à l'attribut style
 - ▶ Feuille de style interne
 - ▶ Feuille de style externe
- ▶ Feuille de style en cascade
 - ▶ Puisqu'on peut placer plusieurs feuilles de style dans un même document, les instructions de style successives peuvent s'opposer.
 - ▶ L'hierarchie prédéfinie est la suivante:

Style externe << Style interne << Style en attribut dans la balise

▶ 8

Appliquer un style

1. Appel à l'attribut style

- Le moyen le plus simple d'appliquer un style est l'utilisation de l'attribut style dans une balise.
- Ajouter du CSS à une balise donnée


```
<p style="text-indent: 15px">
```

 Un style en ligne...


```
</p>
```
- Cette solution présente l'inconvénient d'encore mêler des informations sur la présentation et la structure d'un document.
- Cela rend difficile les mises à jour du code.

► 9

Appliquer un style

2. Feuille de style interne

- Celle-ci sera située dans l'entête, entre les balises <head> </head>
 - On utilisera alors la balise <style> qui prévient le navigateur que des CSS sont utilisées.
- ```
<html><head>
 <style type="text/css">
 body { font-family: sans-serif; background-color: white }
 h1, h2 { color: blue }
 p { text-align: justify }
 </style>
</head>
<body> ... </body>
</html>
```

► 10

## Appliquer un style

### 3. Feuille de style externe

- ```
<html> <head>
<link rel="stylesheet" href="feuilledestyle.css" type="text/css" />
</head> </html>
```
- La feuille de style aura comme extension **.css** et ne devra pas contenir de code HTML ou autre.

<LINK>	La balise avertit le navigateur qu'il faudra réaliser un lien.
REL="stylesheet"	L'attribut précise que le navigateur trouvera une feuille de style externe.
TYPE="text/css"	L'attribut précise que l'information est du texte et de type cascading style sheets (css).
HREF=" ... "	L'attribut de lien donne le chemin d'accès et le nom du fichier à lier.

► 11

Quelques propriétés des CSS

- Les polices
 - font-family: nom de la police;
 - font-size: la taille;
 - color: la couleur;
 - font-weight: normal|bold|light;
- L'alignement du texte
 - text-align: left|right|center|justify;
- L'arrière-plan
 - background-color: couleur;
 - background-image: url|none;
- etc

► 12

Format des styles: Format simple

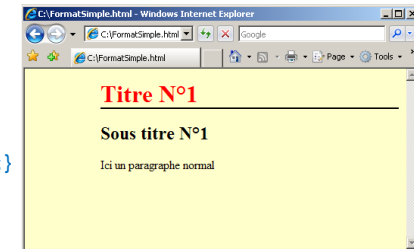
Format simple:

- ▶ Cette format permet de définir des propriétés CSS pour des types d'éléments HTML, par exemple pour structurer le texte ou bien pour les tableaux.
- ▶ A l'utilisation de ces éléments dans le corps du fichier HTML, tous les formats définies pour le type d'éléments seront automatiquement appliqués.

▶ 13

Exemple: Format simple

```
<html> <head>
<style type="text/css">
body {
background-color:#FFFFCC;
margin-left:100px;}
h1 {
color:#FF0000;
border-bottom:solid thin black;}
</style> </head> <body>
<h1>Titre N°1 </h1>
<h2>Sous titre N°1 </h2>
<p>Ici un paragraphe normal</p>
</body> </html>
```



▶ 14

Format des styles: Format pour les balises HTML imbriqués

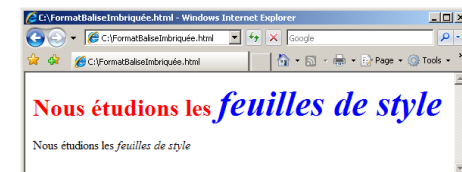
Format pour les balises HTML imbriqués

- ▶ Cette format permet de définir qu'une balise HTML n'aura certaines caractéristiques de formatage que s'il se trouve imbriqué dans une autre balise HTML déterminée.

▶ 15

Exemple: Formats pour les balises imbriqués

```
<html> <head> <style type="text/css">
h1 { color:red;}
h1 i { color:blue;font-size:35pt;}
</style> </head> <body>
<h1>Nous étudions les <i>feuilles de style</i></h1>
<p>Nous étudions les <i>feuilles de style</i></p>
</body> </html>
```



▶ 16

Format des styles: Format liée à l'attribut

Format liée à l'attribut

- ▶ Cette format permet de mentionner dans des sélecteurs que les définitions de format ne doivent s'appliquer qu'à des éléments avec certains attributs ou même qu'à des éléments dont les attributs se voient affecter des valeurs bien précises.

▶ 17

Exemple: Format liés à l'attribut

```
<html><head><style type="text/css">
p { font-weight:bold;font-family:Tahoma,sans-serif;font-size:14pt;}
p[align] { color:red;}
p[align=center] { color:blue;font-size:18pt;}
*[lang=en] { background-color:#FF34F3;color:#FFFFFF;}
</style>
</head>

<body>
<p>Texte Simple</p>
<p align="right">Texte à droite</p>
<p align="center">Texte centré</p>
<p>CSS:<span lang="en">Cascading Style Sheets</span></p>
</body></html>
```



▶ 18

Format des styles: Formats pour des classes

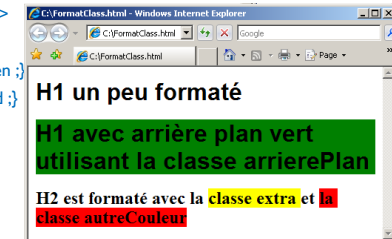
Formats pour des classes

- ▶ Vous pouvez définir des formats pour des classes. Vous pouvez utiliser de telles classes en HTML avec l'attribut universel «class»

▶ 19

Exemple: Format pour des classes

```
<html> <head> <style type="text/css">
h1 { font-family:Ariel;font-size:24pt;}
h1.arrierePlan { background-color:green;}
*.autreCouleur { background-color:red;}
.extra { background-color: yellow;}
</style>
</head>
<body>
<h1>H1 un peu formaté</h1>
<h1 class="arrierePlan"> H1 avec arrière plan vert utilisant la classe arrierePlan </h1>
<h2>H2 est formaté utilisant la <span class="extra"> classe extra </span> et
<b class="Autrecouleur">la classe autreCouleur</b> </h2>
</body>
</html>
```



▶ 20

Format des styles

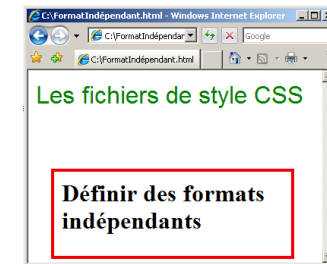
Formats indépendants

- Étant donné que l'affectation de valeur à un tel attribut doit être un nom sans ambiguïté pour tout le document, il s'agit ici simplement d'une définition de format globale pour ce seul élément qui porte ce nom **id**.

21

Exemple: Formats indépendants

```
<html><head><style type="text/css">
#passagerouge {
  position:absolute;
  top:130px;left:30px;
  width:320px;padding:10px;
  border:4px solid #EE0000;
}
h1#titre {
  font-family: Arial,sans-serif;
  font-size:24pt;
  font-weight:normal;
  color:green;
}
</style>
</head><body>
<h1 id="titre">Les fichiers de style CSS</h1>
<div id="passagerouge"><h1>Définir des formats indépendants</h1></div>
</body></html>
```



22

Propriété « Position »:

Position: static;

```
.static {
  position: static;
}
```

static est la valeur par défaut de tous les éléments. Un élément avec **position: static** n'est positionné d'aucune manière spéciale. Un élément **static** est dit *non positionné* et un élément avec une propriété position ayant une valeur autre que **static** est dit *positionné*.

Position: relative;

```
.relative1 {
  position: relative;
}
.relative2 {
  position: relative;
  top: 20px;
  left: 20px;
  background-color: white;
  width: 200px;
}
```

relative se comporte de la même façon que **static** sauf si vous ajoutez quelques propriétés en plus.

Ajouter les propriétés **top**, **right**, **bottom** et **left** à un élément positionné en **relative** va le placer ailleurs que sa position normale. Le reste du contenu ne sera pas ajusté pour prendre la place dans l'espace laissé par l'élément.

23

Propriété « Position »:

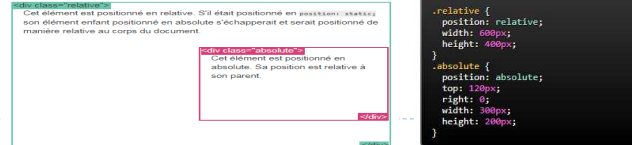
Position: fixed

Un élément positionné en **fixed** est positionné par rapport à la fenêtre du navigateur, ce qui signifie qu'il reste toujours à la même place même si la page défile.



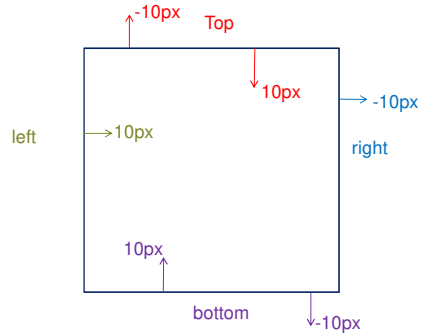
Position: absolute;

absolute est la valeur la plus délicate. **absolute** se comporte comme **fixed** sauf que son positionnement est relatif à l'élément parent positionné le plus proche au lieu d'être relatif à la fenêtre du navigateur.



24

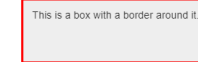
Propriété Top, bottom, right, left



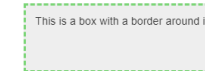
► 25

Propriété: Border

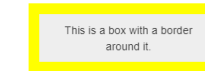
- Border: solid red;



- Border: dashed #7AD576;

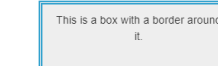


- Border: 1em solid yellow;



Epaisseur border: px, pt, cm, em ou bien en utilisant l'un des valeurs prédéfinie: thin, medium, ou thick.

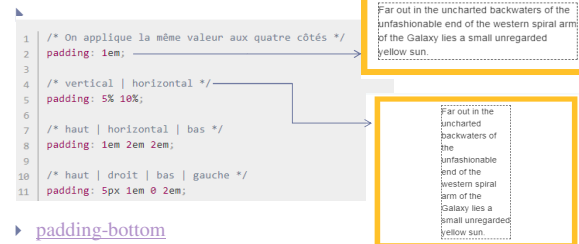
- border: thick double #32a1ce;



► 26

Propriété: padding

- La propriété **padding** permet de définir les différents écarts de remplissage sur les quatre côtés d'un élément. Elle synthétise **padding-top**, **padding-right**, **padding-bottom**, **padding-left**.

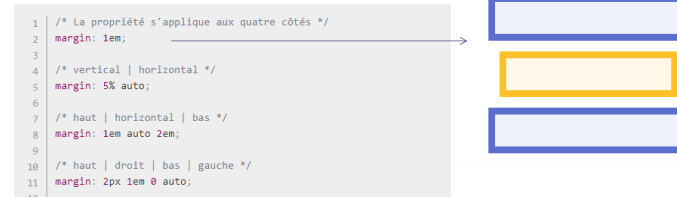


- [padding-bottom](#)
- [padding-left](#)
- [padding-right](#)
- [padding-top](#)

► 27

Propriété: margin

- La propriété **margin** définit la taille des marges sur les quatre côtés de l'élément. C'est une propriété raccourcie qui permet de manipuler les autres propriétés de marges : **margin-top**, **margin-right**, **margin-bottom** et **margin-left**.



- auto est remplacé par une valeur adaptée (il peut être utilisé pour centrer les blocs, par exemple)

► 28

Propriété font:

Font-family :	Time new roman Verdana Arial	définit la police de caractère utilisée
Font-size:	20% 5pt 10px medium, large	définit la taille de la police de caractère.
Font-style:	italic, oblique, normal	Elle détermine l'orientation de la police. 3 valeurs aux choix : italic, oblique, normal (par défaut).
Font-weight :	bold, bolder, lighter	Elle définit l'épaisseur de la police. 4 possibilités : bold, bolder, lighter et normal (par défaut)

► 29

Propriété: Background

background-color:	red, #8A27F2
background-image:	url("Myimage.png");
background-repeat:	no-repeat repeat-x round

Round : (Par défaut, les images répétées sont rognées à la taille de l'élément mais elles peuvent être redimensionnées pour occuper l'espace avec un nombre entier de répétitions

► 30

- Indiquez pour chaque élément de la liste la couleur dans laquelle il s'affiche.

- Justifiez vos réponses.

```
<html>
<head>
<title> exercice1 </title>
<style type="text/css">
body{color: red;}
ul.default{color:black;}
style.style {color:green;}
*.style {color:yellow;}
#js{color: orange;}
ul#dyn {color:blue;}
.new {color:green;}
</style>
</head>
<body>
<ul>
<li class="default"> HTML</li>
<li class="style"> CSS</li>
<li id="js"> JS</li>
</ul>
<ul id="dyn">
<li> Perl</li>
<li class="new"> Python</li>
</ul>
</body>
</html>
```

► 31

Sélecteurs d'attributs (CSS3)

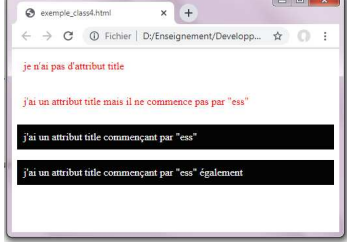
3 nouveaux sélecteurs d'attribut sont introduits par la norme CSS 3 :

- **[att^="val"]** Représente un attribut dont la valeur commence exactement par le **préfixe** "val".
- **[att\$="ident"]** Représente un attribut dont la valeur finit exactement par le **suffixe** "ident".
- **[att*="val"]** Représente un attribut dont la valeur contient au moins une fois la sous-chaîne "val".

► 32

Exemple 1: Sélecteurs d'attributs

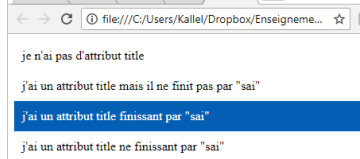
```
<html> <head>
<style type="text/css">
p.example{padding:10px;
        color:red;}
p.example[title^="ess"]{ color:#FFFFFF;
        background-color:#030303;}
</style> </head>
<body>
<p class="example"> je n'ai pas d'attribut title</p>
<p class="example" title="comment"> j'ai un attribut title mais il ne commence pas
par "ess"</p>
<p class="example" title="essai"> j'ai un attribut title commençant par "ess"</p>
<p class="example" title="esson"> j'ai un attribut title commençant par "ess"
également</p> </body> </html>
```



33

Exemple 2: Sélecteurs d'attributs

```
<html> <head>
<style type="text/css">
p.example2{padding:10px;
        color:#0000;}
p.example2[title$="sai"]{ color:#fffff;
        background-color:#045FB4;}
</style> </head>
<body>
<p class="example2"> je n'ai pas d'attribut title</p>
<p class="example2" title="comment"> j'ai un attribut title mais il ne finit pas par
"sai"</p>
<p class="example2" title="essai"> j'ai un attribut title finissant par "sai"</p>
<p class="example2" title="esson"> j'ai un attribut title ne finissant par "sai" </p>
</body> </html>
```



34

Exemple 3: Sélecteurs d'attributs

```
<html> <head>
<style type="text/css">
p.example3{ margin:0;padding:10px;
        color:#00FF00;}
p.example3[title*="val"]{ color:#FFFFFF;
        background:#990000;}
</style> </head>
<body>
<p class="example3"> je n'ai pas d'attribut title</p>
<p class="example3" title="comment"> j'ai un attribut title mais il ne contient pas "val"</p>
<p class="example3" title="val"> j'ai un attribut title contenant au moins "val"</p>
<p class="example3" title="evaluer"> j'ai un attribut title contenant au moins "val"
également</p>
<p class="example3" title="eval"> j'ai un attribut title contenant au moins "val"
également</p> </body> </html>
```



35

Format des styles: format pour des pseudo-éléments (CSS3)

- Un **pseudo-élément** est un mot-clé ajouté à un sélecteur qui permet de mettre en forme certaines parties de l'élément ciblé par la règle.
- Syntaxe:

`sélecteur::pseudo-élément { propriété: valeur; }`

36

Les pseudo-éléments

- ▶ **::first-letter** : ne sélectionner que la première lettre d'un texte contenu dans un élément ;
- ▶ **::first-line** : ne sélectionner que la première ligne d'un texte contenu dans un élément ;
- ▶ **::selection** : sélectionner la partie d'un élément sélectionné (par un double clic) par l'utilisateur ;
- ▶ **::before** : insérer du contenu au début d'un élément HTML ;
- ▶ **::after** : insérer du contenu à la fin d'un élément HTML.

▶ 37

Exemple 1: Les pseudo-éléments

styles.css

```
p::first-letter{
  font-size: 39px;
  color: #A92;
}
```

Pseudo éléments CSS

Un premier paragraphe

Ceci est un deuxième paragraphe

Et voilà un troisième paragraphe

Un paragraphe hors div

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo éléments</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Pseudo éléments CSS</h1>
    <div>
      <p>Un premier paragraphe</p>
      <p>Ceci est un deuxième paragraphe</p>
      <p>Et voilà un troisième paragraphe</p>
    </div>
    <p>Un paragraphe hors div</p>
  </body>
</html>
```

▶ 38

Exemple 2: Les pseudo-éléments

styles.css

Pseudo éléments CSS

Un premier paragraphe qui contient beaucoup de texte

afin que celui-ci occupe plusieurs lignes au sein de notre page web

On peut aussi ajouter des break...

Un paragraphe hors div

```
p::first-line{
  font-size: 39px;
  color: #A92;
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo éléments</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Pseudo éléments CSS</h1>
    <div>
      <p>Un premier paragraphe qui contient beaucoup de texte afin
      que celui-ci occupe plusieurs lignes au sein de notre page web</p>
      <p>On peut<br>aussi<br>ajouter des break...</p>
    </div>
    <p>Un paragraphe hors div</p>
  </body>
</html>
```

▶ 39

Exemple 3: Les pseudo-éléments

styles.css

- ▶ Dans l'exemple, évidemment, j'ai sélectionné les mots "zone sélectionnée" avec ma souris au moment de prendre la capture d'écran.

```
/*Alternative pour Mozilla (qui supporte mal ::selection)*/
p::-moz-selection{
  background-color: orange;
}
/*Pour les autres navigateurs*/
p::selection{
  background-color: orange;
}
```

Pseudo éléments CSS

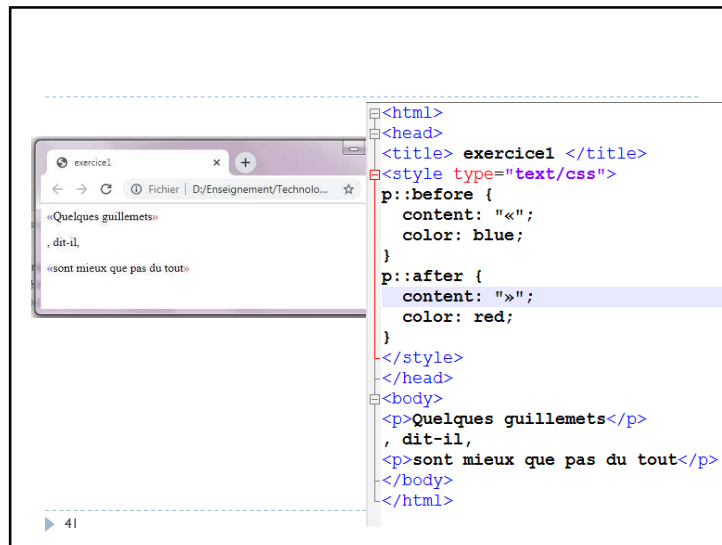
Un premier paragraphe avec une zone sélectionnée

Un deuxième paragraphe

Un paragraphe hors div

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo éléments</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Pseudo éléments CSS</h1>
    <div>
      <p>Un premier paragraphe avec une zone sélectionnée</p>
      <p>Un deuxième paragraphe</p>
    </div>
    <p>Un paragraphe hors div</p>
  </body>
</html>
```

▶ 40



Format des styles: Format pour des pseudo-classes (CSS3)

- ▶ Une **pseudo-classe** est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être ciblé par la déclaration.
- ▶ Syntaxe:

Sélecteur:pseudo-classe { propriété: valeur; }

Les pseudo-classes

- ▶ **:link** : styliser un lien non visité;
- ▶ **:visited** : styliser un lien une fois celui-ci visité.
- ▶ **:hover** : changer l'aspect d'un élément lorsque vos visiteurs poseront leur curseur dessus ;
- ▶ **:active** : modifier l'aspect d'un lien lors du clic.
- ▶ **:first-child** : sélectionner le premier élément enfant HTML d'un certain type par rapport à un élément parent.
- ▶ **:last-child** : sélectionner le dernier élément enfant HTML d'un certain type par rapport à un élément parent.
- ▶ **:nth-child()** : cibler les éléments en se basant sur leur position dans la liste des enfants de leur parent. (Vous pouvez utiliser un numéro, une expression numérique ou les mots "odd" et "even" correspondant à impair et pair)
- ▶ **:checked** : cibler les éléments qui sont cochés
- ▶ **:empty** : cibler les éléments qui n'ont pas d'enfant ou qui sont vides
- ▶ **etc**

Exemple 1 : Les pseudo-classes

styles.css

- ▶ Le curseur de notre souris se trouve sur notre titre h1; celui-ci change donc de couleur et de police. De plus, nous avons déjà visité notre deuxième lien mais nous n'avons pas encore cliqué sur le premier.



Exemple 2 : Les pseudo-classes

styles.css

```
p:first-child{
  color: orange;
}
p:last-child{
  color: green;
}
```

file:///Users/Pierre/Desktop/Cours%20

Pseudo classes CSS

Un premier paragraphe
Un deuxième paragraphe
Un troisième paragraphe
Un paragraphe hors div

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo classes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Pseudo classes CSS</h1>
    <div class="test">
      <p>Un premier paragraphe</p>
      <p>Un deuxième paragraphe</p>
      <p>Un troisième paragraphe</p>
    </div>
    <p>Un paragraphe hors div</p>
  </body>
</html>
```

45

Exemple 3 : Les pseudo-classes

styles.css

```
/*Ce sélecteur va cibler tous les éléments p qui sont des enfants
*pairs (2è enfant, 4è enfant, etc.) d'un élément HTML dans notre page*/
p:nth-child(even){
  color: purple;
}

/*Ce sélecteur va cibler tous les éléments h2 qui sont le deuxième
*enfant d'un élément parent*/
h2:nth-child(2){
  color: green;
}
```

file:///Users/Pierre/Desktop/cours.html

Un titre de niveau 1

Un titre de niveau 2

Un premier paragraphe
Un deuxième paragraphe

Un autre titre de niveau 2

Un troisième paragraphe
Un quatrième paragraphe

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pseudo classes CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <h2>Un titre de niveau 2</h2>
    <p>Un premier paragraphe</p>
    <p>Un deuxième paragraphe</p>
    <div>
      <h2>Un autre titre de niveau 2</h2>
      <p>Un troisième paragraphe</p>
      <p>Un quatrième paragraphe</p>
    </div>
  </body>
</html>
```

46

Les seize noms de couleurs utilisables en CSS

white	
silver	
gray	
black	
red	
maroon	
lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

47

Modes de sorties

- ▶ L'écran et l'imprimante par exemple sont des modes de sortie très différents pour des données structurées agréablement.
- ▶ Avec les fichiers de styles CSS, on peut incorporer différents fichiers de feuilles de style pour les différents **modes de sortie**, qui peuvent contenir des définitions de formats différentes.
- ▶ Le logiciel doit décider lors de la présentation des pages quel fichier CSS incorporé est déterminant.
 - ▶ Si le navigateur Web affiche les pages à l'écran, il doit utiliser le fichier CSS qui fixe expressément l'"**écran**" comme mode de sortie,
 - ▶ Si l'utilisateur imprime une page, le navigateur doit employer à la place le fichier CSS pour le mode "**imprimante**".

48

Modes de sorties

```
<html>
<head> <title>Titre du fichier</title>
<link rel="stylesheet" media="screen" href="website.css">
<link rel="stylesheet" media="print" href="impression.css">
<link rel="stylesheet" media="aural" href="bafles.css">
</head>
<body>
...
</body>
</html>
```

► 49

Modes de sorties

Mention	Signification
media="all"	CSS s'applique à tous les types de médias.
media="aural"	CSS s'applique à des systèmes de restitution vocale assistée par ordinateur.
media="handheld"	CSS s'applique à l'affichage sur des ordinateurs de poche petits.
media="print"	CSS s'applique à l'impression sur papier. Les navigateurs Web doivent utiliser ces définitions de format quand l'utilisateur désire imprimer la page Web.
media="projection"	CSS s'applique à la projection de données avec des rétroprojecteurs ou appareils similaires.
media="screen"	CSS s'applique à l'affichage à l'écran.
media="tty"	CSS s'applique à des médias de sortie non graphiques avec une largeur de signes invariable comme par exemple des télex.
media="tv"	CSS s'applique à des médias de sortie semblables à la télévision qui se distinguent par une résolution grossière

► 50

- Indiquez pour chaque élément de la liste la couleur dans laquelle il s'affiche.
- Justifiez vos réponses.

```
<html>
<head>
<style type="text/css">
body {color: blue; }
h1 {color: green; }
.h1 {color: blue; }
ol[start] li {color: purple;}
#h2{color:yellow;}
li:first-child {color: orange;}
li[title^="sel"] {color: red; }
</style> </head>
<body>
<h1>texte [1]</h1>
<ul>
<li>texte [2]</li>
<li>texte [3]</li>
</ul>
<h2 id="h2" title="selection">texte [4]</h2>
<ul>
<li>texte [5]</li>
<li class="h1" title="selecteur">texte [6]
<ol start="2">
<li>texte [7]</li>
</ol></li></ul>
</body>
</html>
```

► 51