



Faculty of Sciences and Technology
Telecommunications & electronics Department

Technologies du web

*** Styles de présentation et design ***

K. ABAINIA

Last update: 10/02/2021

CSS ?

CSS = Cascading Style Sheets
Feuilles de style en cascade



~~N'est pas~~ un langage de programmation (pas de variable)

Langage de **présentation** du **contenu** web (**ensemble de règles**)

Introduit en 1996 par

W3C

But = **Séparer** la présentation du contenu

Usage du CSS

- ❖ **Un style** (au lieu plusieurs) pour **plusieurs** éléments HTML
- ❖ **Une feuille** de styles pour **plusieurs** page HTML
- ❖ Créer des **animations/effets spéciaux** sans Javascript
- ❖ Différentes **présentations/layout** pour différents **médiums**
- ❖ Positionner, dimensionner et tourner un élément HTML
- ❖ **Préformatage** d'une page web pour l'impression

Pourquoi CSS ? Avantages !

Gain du temps -> **réduction** de la redondance

Facilité de la **maintenance**

Chargement **rapide** de la page (**taille réduite**)

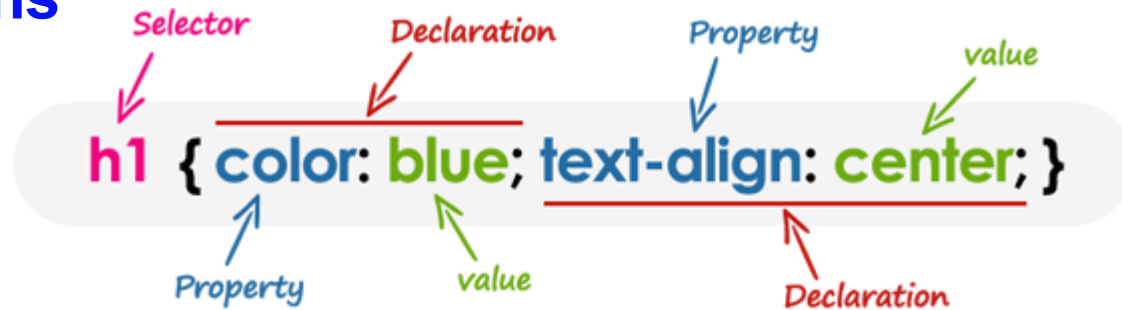
Grand choix de styles par rapport au HTML

Compatibilité sur différents médium (**même page**)



Syntaxe des styles (1)

Règle définie par un **sélecteur** et un ensemble de **déclarations**



Sélecteur = **un** ou **plusieurs** éléments HTML

Sélecteur = **nom** de l'élément, **classe** de l'élément ou **ID**

Déclarations entre **deux** accolades

Syntaxe des styles (2)

Noms des éléments (ou classe ou ID) sont **séparés** par **virgule**

```
h1, h2 {color: red;}
```

Préférable d'écrire chaque **déclaration** dans une ligne (;)

Commentaire (multi-ligne) est écrit entre **/* ... */**

Sélecteur est **sensible** à la case (majuscule/minuscule)

Nom de propriété/valeur **n'est pas sensible** à la case

Inclusion du CSS dans HTML (1)

CSS

plus
pratique

inline

un attribut
style à
l'intérieur de
la balise

Déconseillée

embarqué

entre une
balise
<style> dans
<head>

Déconseillée

externe

dans un
fichier **.css**

Inclusion du CSS dans HTML (2)

inline: on ne peut pas appliquer le style à plusieurs éléments

```
<div style="color:red; font-size:16px;">Notre premier exemple de CSS</div>
```

embarqué: on ne peut pas appliquer le style à plusieurs pages HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<style>
h1{ background-color: #FEFEFE; }
</style>
</head>
<body>
  <h1>Un test</h1>
</body>
</html>
```

+ fichier HTML

volumineux



Inclusion du CSS dans HTML (3)

externe: feuille de styles appliquée à **plusieurs** pages web

- + **même règle** peut être appliquée à **plusieurs** éléments
- + optimiser la bande passante (fichier sauvegardé)

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

```
<style>
@import
url("chemin/style.css");
p {
    color: blue;
    font-size: 16px;
}
</style>
```

```
@import
url("chemin/layout.css");
@import
url("chemin/color.css");
body {
    color: blue;
    font-size: 14px;
}
```

Sélecteurs (1)

Sélecteur universel: dénoté par ***** et appliqué à tous les éléments de la page web, mais il est ignoré pour un élément si une autre règle spécifique à ce dernier existe

```
* {  
  font-family: Arial;  
  font-size: 14px;  
}
```

Sélecteur par type d'élément: dénoté par un nom d'élément ou plusieurs **séparés** par des **virgules**

```
div, span {  
  font-family: Arial;  
  font-size: 14px;  
}
```

Sélecteurs (2)

Sélecteur par ID: dénoté par le nom d'ID précédé par (#) et appliqué à cet élément seulement (ID doit être unique)

```
#input_error {
  color: red;
  font-style: italic;
}
```

Sélecteur par class: dénoté par le nom de classe précédé par (.) et appliqué à tous les éléments appartenant à cette classe

```
.input-field {
  color: gray;
  font-size: 12px;
}
```

```
<textarea name="comment" class="input-field"
/>
<input type="text" name="email" class="input-
field" />
```

Sélecteurs (3)

Sélecteur par élément et class: dénoté par le nom d'élément **succédé** par (.) et le **nom de la classe**, et est appliqué à tous les éléments portant le même nom et classe

```
textarea.input-field {
  color: gray;
  font-size: 12px;
}
```

```
<textarea name="comment" class="input-field" />
<input type="text" name="email" class="input-field" />
```

Sélecteur par élément descendant: dénoté par le nom d'élément **précédé** par un **espace** et le nom de l'**élément ancêtre**

```
h1 p {
  color: red;
}
```

```
<h1>
  Texte non coloré
<div> <p> texte coloré </p> </div>
</h1>
```

Sélecteurs (4)

Sélecteur par élément enfant: comme le cas de l'élément descendant, mais dénoté par le nom de l'élément enfant précédé par le symbole **>** et le nom du parent direct

```
h1 > div > p {
  color: red;
}
```

```
<h1>
  Texte non coloré
<div> <p> texte coloré </p> </div>
</h1>
```

Sélecteur par élément adjacent: dénoté par l'élément concerné précédé par le symbole **+** et le nom de l'élément adjacent ayant le même parent

```
p + div {
  color: red;
}
```

```
<h1>
  <p> Texte non coloré </p>
  <div> texte coloré </div>
</h1>
```

Sélecteurs (5)

Sélecteur par élément voisin: comme le cas de l'élément adjacent, mais dénoté par le nom de l'élément **précédé** par le **symbole ~** et le nom de l'élément **voisin direct** ou **indirect** et ayant le **même parent**

```
p ~ div {  
  color: red;  
}
```

```
<h1>  
<p> Texte non coloré </p>  
<span> Texte non coloré 2 </span>  
<div> texte coloré </div>  
</h1>
```

Sélecteurs avancés (1)

Sélecteur par attribut: le nom d'attribut est mis entre **[]** et le style s'applique à tous les éléments satisfaisant la règle

tous les
éléments
élément
spécifié
élément et valeur
spécifiés
mot séparé par
espace
mot en début succédé
par -
se débute par la
chaîne
se termine par la
chaîne
contient le
mot

```
[href] { color: red; }
```

```
a[href] { color: red; }
```

```
img[src="button.jpeg"] { border: 1px solid  
black; }
```

```
div[class~="warning"] { border: 1px solid red; }
```

```
div[lang|="en"] { border: 1px solid  
red; }
```

```
a[href^="https://"] { color:  
red; }
```

```
a[href$=".pdf"] { color:  
red; }
```

```
div[lang*="en"] { border: 1px solid  
red; }
```

Sélecteurs avancés (2)

```
div[class~="warning"] {  
  color: red;  
}
```

```
<div class="alert warning"> texte coloré </div>
```

```
<div class="warning"> texte coloré </div>
```

```
<div class="warning alert"> texte coloré </div>
```

```
div[lang|="en"] {  
  color: red;  
}
```

```
<div lang="en"> texte coloré </div>
```

```
<div lang="en-US"> texte coloré </div>
```

```
<div lang="US-en"> texte coloré </div>
```

```
div[class*="en"] {  
  color: red; }  
  
a[href^="https://"] {  
  color: green; }
```

```
<div class="genx"> texte coloré </div>
```

```
<a href="https://site.com"> site sécurisé </a>
```

```
<a href="http://site.com"> site non sécurisé </a>
```


Sélecteurs avancés (3)

Sélecteur par pseudo-class: le nom de l'élément **succédé** par **:** et le nom du **pseudo-class**

lien

```
a:link { color: blue; }
```

lien
visité
curseur
dessus
click
dessus
raccourci du clavier

```
a:visited { color: red; }
```

```
a:hover { color: red; }
```

```
a:active { color: red; }
```

```
a:focus { color: red; }
```

appuyé
premier

```
div p:first-child { color: red; }
```

fil
dernier

```
div p:last-child { color: red; }
```

fil
un plusieurs
fil

```
div p:nth-child(arg) { color: red; }
```

spécialement
pour <a>

odd
even
2n
1n
2n+1

Sélecteurs avancés (4)

```
div p:first-child {
  color: red; }
p.under {
  text-decoration: underline;
}
```

```
<div>
<p class="under"> texte souligné et coloré </p>
<p class="under"> texte souligné </p>
<p> texte normal </p>
</div>
```

```
div p:nth-child(even) {
  color: red; }
div p:nth-child(odd) {
  color: blue; }
```

```
<div>
<p> texte normal </p>
<p> texte normal </p>
<p> texte normal </p>
<p> texte normal </p>
</div>
```

Sélecteurs avancés (5)

par attribut

```
élément:lang(valeur) { color: red; }
```

lang

input

```
élément:checked { color:  
red; }
```

sélectionné

input

```
élément:disabled { color:  
red; }
```

désactivé

contenu vide

```
élément:empty { background:  
red; }
```

valeur d'input

invalide

```
élément:invalid { color:  
red; }
```

valeur d'input dans

l'intervalle

```
élément:in-range { color:  
green; }
```

valeur d'input dehors

del'intervalle

```
élément:out-of-range { color:  
red; }
```

tout élément différent de

l'élément sélectionné

```
:not(élément) { color: red;  
}
```

Il y a d'autres sélecteurs par pseudo-class

Sélecteurs avancés (6)

```
input:out-of-range {  
  background: red; }
```

```
<div>  
<label> Mois: </label>  
<input type="number" min="1"  
max="12"/>  
</div>
```

20

```
p:empty{  
  background: red;  
  width: 100px;  
  height: 20px; }
```

```
<div>  
<p> texte </p>  
<p> </p>  
<p> </p>  
<p> texte </p>  
</div>
```

texte



texte

Sélecteurs avancés (7)

Sélecteur par pseudo-élément: le nom d'attribut **succédé** par **::** et le nom du **pseudo-élément** (dans l'absence de l'ID et la classe)

première ligne de
paragraphe
première lettre du
texte

contenu avant

l'élément

contenu après

l'élément

élément input contenant

placeholder

(valeur attendue)

sélection d'une partie du
texte

```
élément::first-line { color:
blue; }
```

```
élément::first-letter { color:
red; }
```

```
élément::before { content: "un
texte"; }
```

```
élément::after { content: url("http://lien-vers-
image"); }
```

```
élément::placeholder { color:
red; }
```

```
::selection { color: red; background:
yellow; }
```

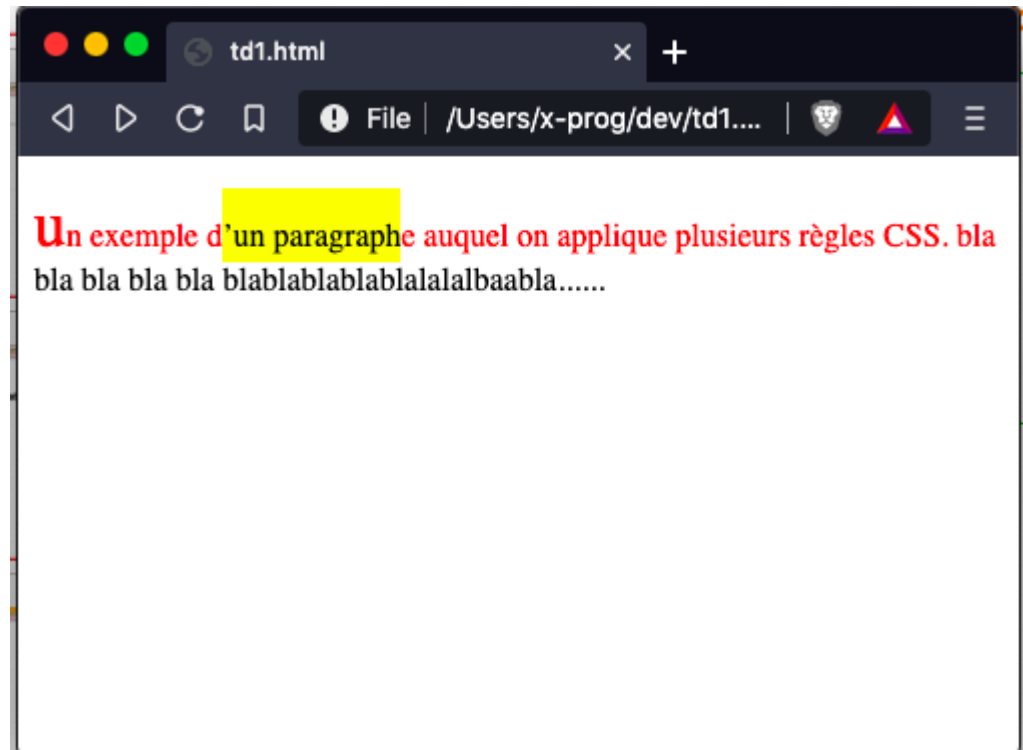
Sélecteurs avancés (8)

```
p:first-line {
color: red;
}
p:first-letter {
font-size: xx-large;
}
::selection {
background: yellow;
}
```

<p>

un exemple d'un paragraphe auquel on applique plusieurs règles CSS. bla bla bla bla bla blablablablalalalbaabla.....

</p>



Valeurs des couleurs en CSS

❖ **nom de la couleur**: white, red, black, purple, etc.

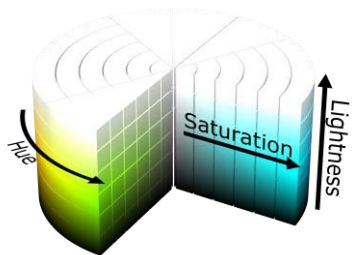
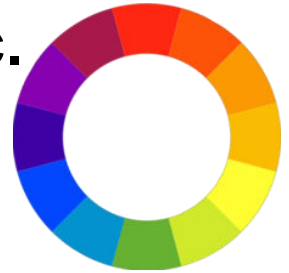
❖ **valeur hexadécimale**: #FFF ou #FFFFFF

❖ **couleur RGB**: fonction `rgb(255, 255, 255)` ou `rgb(100%, 100%, 100%)`

❖ **couleur RGBA**: fonction `rgb(255, 255, 255, [0.0-0.1])` ou `rgb(100%, 100%, 100%, [0.0-0.1])`

❖ **couleur HSL**: fonction `hsl([0-360], 100%, 100%)`

❖ **couleur HSLA**: fonction `hsla([0-360], 100%, 100%, [0.0-0.1])`



Unités en CSS

- ❖ **px**: pixel.
- ❖ **cm**: centimètre
- ❖ **mm**: millimètre
- ❖ **in**: inch (1 in = 2.54cm)
- ❖ **pt**: points (1 pt = 0.353 mm)
- ❖ **pc**: picas (1 pc = 12 pt)
- ❖ **em**: taille de police courante
- ❖ **ex**: hauteur du x courant



Pour les
textes

Propriétés CSS pour les textes (1)

- ❖ **color**: changer la couleur du texte (valeur)
- ❖ **direction**: changer la direction d'écriture (rtl | ltr)
- ❖ **text-decoration**: souligner le texte (underline | overline | line-through)
- ❖ **text-align**: aligner le texte (right | center | left | justify)
- ❖ **text-indent**: indenter un paragraphe (valeur avec unité)
- ❖ **text-transform**: capitaliser le texte (capitalize | uppercase | lowercase)
- ❖ **text-justify**: spécifier la méthode de justification (auto | inter-word | inter-character | none | initial | inherit)

#RRGGBB
#RGB
rgb(0-255,0-255,0-255)
rgb(n%,n%,n%)
keyword

Propriétés CSS pour les textes (2)

❖ **text-shadow**: ajouter un ombre au texte

text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit;

❖ **text-overflow**: rétrécir le texte en cas dépassement de la zone

white-space: nowrap;
overflow: hidden;
text-overflow: clip|ellipsis|"string"|initial|inherit;

❖ **letter-spacing**: ajouter ou réduire l'espace entre les lettres

❖ **word-spacing**: ajouter ou réduire l'espace entre les mots

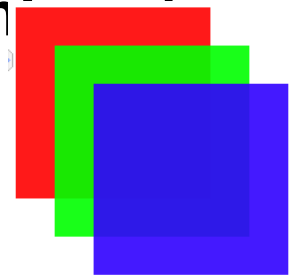
❖ **white-space**: contrôler le formatage du texte (nowrap |

Propriétés CSS des polices

- ❖ **font-family**: choisir la police des caractères (georgia | serif |...)
- ❖ **font-style**: spécifier le style d'affichage (normal | italic | oblique)
- ❖ **font-variant**: afficher le texte en petite majuscule (normal | small-caps)
- ❖ **font-weight**: incrémenter ou décrémenter l'épaisseur (bold | normal | bolder | lighter | 100-900)
- ❖ **font-size**: spécifier la taille de la police (valeur + unité) ou keyword (small | medium | large | smaller | larger | x-small | xx-small | x-large | xx-large)

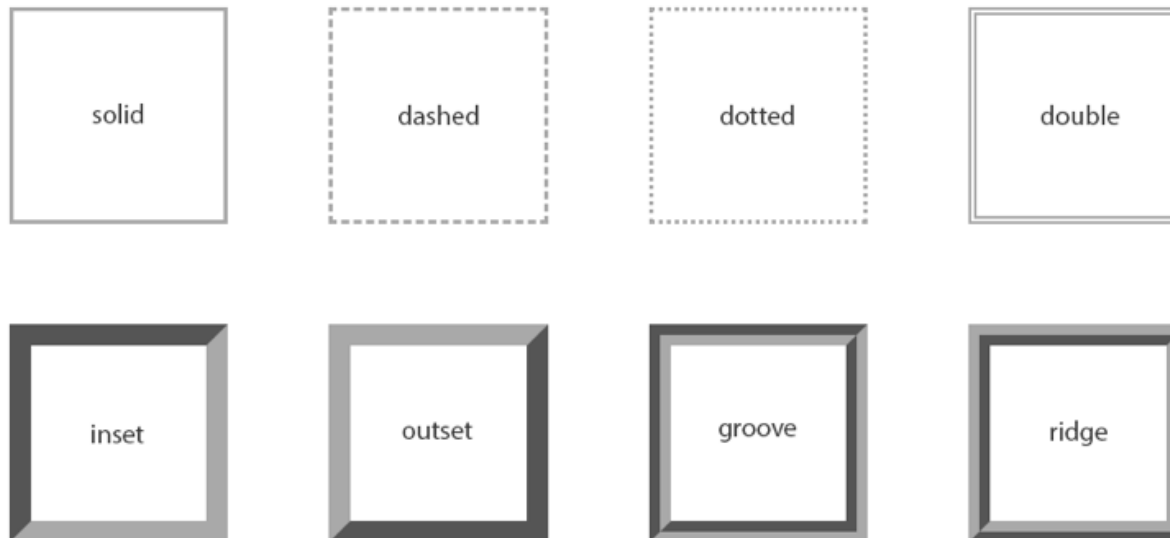
Propriétés CSS d'affichage

- ❖ **display**: choisir le mode d'affichage (block | inline | inline-block | none)
- ❖ **visibility**: contrôler la visibilité d'un élément (visible | hidden | collapse)
- ❖ **opacity**: contrôler la transparence d'un élément [0.0 , 0.1]
- ❖ **z-index**: positionner un élément sur l'axe Z (valeur en entier)










Propriétés CSS de bordure

- ❖ **border**: largeur style couleur (**1px solid black**)
- ❖ **border-radius**: arrondir les angles (valeur en px)
- ❖ **outline**: largeur style couleur (**1px solid black | none**)



Changement du curseur

❖ **cursor**: valeur ou fonction url(lien vers image)

Look	Values	Example
	default	<code>a:hover{cursor:default;}</code>
	pointer	<code>a:hover{cursor:pointer;}</code>
I	text	<code>a:hover{cursor:text;}</code>
	wait	<code>a:hover{cursor:wait;}</code>
	help	<code>a:hover{cursor:help;}</code>
	progress	<code>a:hover{cursor:progress;}</code>
+	crosshair	<code>a:hover{cursor:crosshair;}</code>
	move	<code>a:hover{cursor:move;}</code>
	url()	<code>a:hover{cursor:url("custom.cur"), default;}</code>

tirée du site <https://www.tutorialrepublic.com>

Propriétés CSS d'espacement (1)

- ❖ **margin**: ajouter un espace autour de l'élément (**valeur en px**)
- ❖ **margin-top margin-bottom margin-left margin-right**: ajouter un espace sur une région spécifique
- ❖ **padding**: ajouter un espace entre le contenu d'un élément et sa bordure (**valeur en px**)
- ❖ **padding-top padding-bottom padding-left padding-right**: ajouter un espace sur une région spécifique

Propriétés CSS d'espacement (2)

```
.class1 {  
padding-bottom: 15px;  
border: 1px solid black;  
}
```

```
.class2 {  
padding-bottom: 50px;  
padding-left: 20px;  
border: 1px solid black;  
}
```

```
<p class = "class1">
```

Espace à gauche

```
</p>
```

```
<p class = "class2">
```

Espace à gauche et en bas

```
</p>
```

Espace à gauche

Espace à gauche et en bas

Propriétés CSS d'espacement (3)

```
.class1 {  
margin-left: 15px;  
border:1px solid black;  
}
```

```
.class2 {  
margin-left: -50px;  
border:1px solid black;  
}
```

```
<p class = "class1">
```

Décalage de l'élément vers la droite

```
</p>
```

```
<p class = "class2">
```

Décalage de l'élément vers la gauche

```
</p>
```

Dépassement
de la fenêtre

Décalage de l'élément vers la droite

Décalage de l'élément vers la gauche

Propriétés CSS de positionnement (1)

❖ **position**: spécifier la relativité de l'élément (**relative** | **absolute** | **fixed**)

position
fixe par
rapport à
l'écran

position
relative au
parent

position
relative à
l'écran

❖ **left, right, top, bottom**: spécifier la valeur de la position de l'élément (négative ou positive)

Propriétés CSS de positionnement (2)

```
.class1 {  
  position: relative;  
  left: 80px;  
  top: 2px;  
  background-color: yellow;  
  width: 200px;  
}
```

```
<div class = "class1">  
  Position relative avec 80px .  
</div>
```

Position relative avec 80px .

Propriétés CSS pour des effets spéciaux (1)

❖ **linear-gradient**: créer une couleur en gradient verticale, horizontale ou diagonale (**fonction**)

linear-gradient(couleur 1, couleur 2)

linear-gradient(to right, couleur 1 , couleur 2)

linear-gradient(to bottom right, couleur 1 , couleur 2)

linear-gradient(couleur 1, couleur 2, couleur 3, couleur 4,...)

❖ **radial-gradient**: créer une couleur en gradient radiale (**fonction**)

radial-gradient(red: 100%, yellow: 100%, brown: 100%)

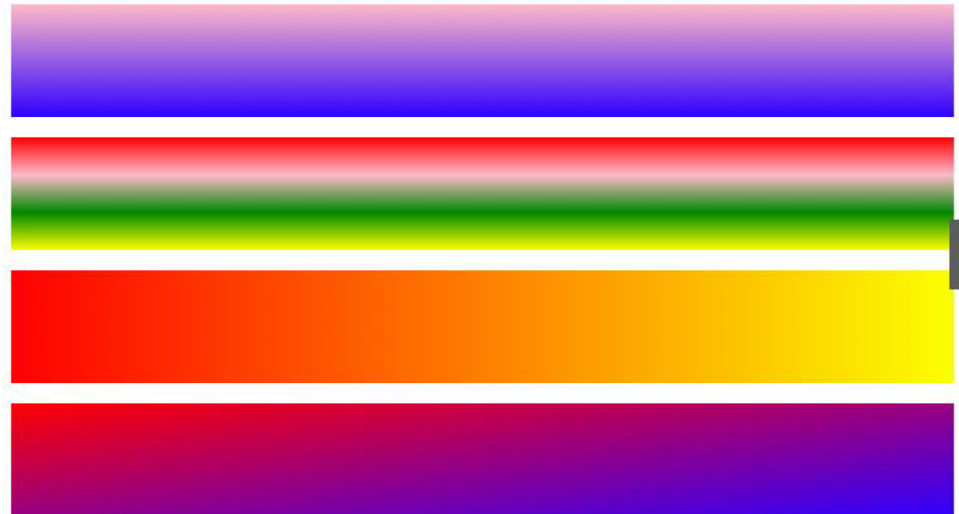
❖ **repeating-radial-gradient**: répéter le gradient radial (**fonction**)

repeating-radial-gradient(red: 100%, yellow: 100%, brown: 100%)

Propriétés CSS pour des effets spéciaux (2)

```
#grad1 {
height: 100px;
background: linear-gradient(pink,
blue);
}
#grad2 {
height: 100px;
background: linear-gradient(red,
pink, green, yellow);
}
#grad3 {
height: 100px;
background: linear-gradient(to
right, red, yellow);
}
#grad4 {
height: 100px;
background: linear-gradient(to
bottom right, red, blue);
}
```

```
<body>
  <div id = "grad1"></div>
  <br/>
  <div id = "grad2"></div>
  <br/>
  <div id = "grad3"></div>
  <br/>
  <div id = "grad4"></div>
</body>
```

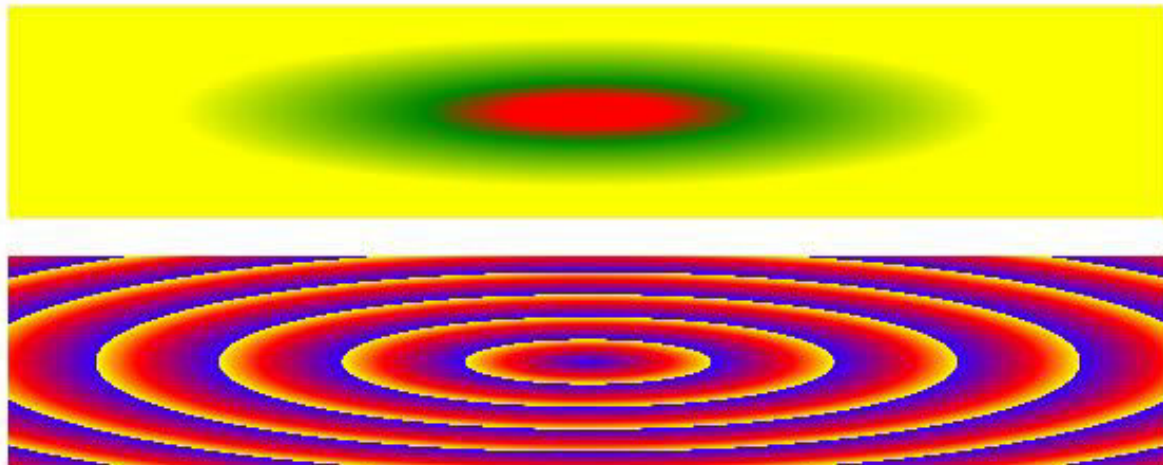


Propriétés CSS pour des effets spéciaux (3)

```
#grad1 {
height: 100px;
background: radial-gradient(red
10%, green 20%, yellow 50%);
}

#grad2 {
height: 100px;
background: repeating-radial-
gradient(blue, red 10%, yellow
15%);
}
```

```
<body>
  <div id = "grad1"></div>
  <br/>
  <div id = "grad2"></div>
</body>
```



Propriétés CSS pour des effets spéciaux (4)

- ❖ **column-count**: diviser le texte en plusieurs colonnes (nombre entier)
- ❖ **column-gap**: spécifier l'espace entre les colonnes
- ❖ **column-rule-color**: spécifier la couleur du séparateur
- ❖ **column-rule-style**: spécifier le style du séparateur
- ❖ **column-rule-width**: spécifier la largeur du séparateur
- ❖ **column-span**: spécifier si une partie du texte est divisée comme le reste ou occupe tout la largeur (**none** | **all**)

Propriétés CSS pour des effets spéciaux (5)

```
.newspaper {
  column-count: 3;
  column-gap: 40px;
  column-rule: 1px solid red;
}

h2 {
  column-span: all;
}
```

```
<div class="newspaper">
<h2>Lorem ipsum dolor sit amet,
consectetuer adipiscing elit</h2>
Lorem ipsum dolor sit amet, consectetuer adipiscing
elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat volutpat. Ut wisi
enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea
commodo consequat. Duis autem vel eum iriure dolor
in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at
vero eros et accumsan et iusto odio dignissim qui
blandit praesent luptatum zzril delenit augue dui
dolore te feugait nulla facilisi.
</div>
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
sed diam nonummy nibh
euismod tincidunt ut laoreet
dolore magna aliquam erat
volutpat. Ut wisi enim ad
minim veniam, quis nostrud

exerci tation ullamcorper
suscipit lobortis nisl ut
aliquip ex ea commodo
consequat. Duis autem vel
eum iriure dolor in hendrerit
in vulputate velit esse
molestie consequat, vel

illum dolore eu feugiat nulla
facilisis at vero eros et
accumsan et iusto odio
dignissim qui blandit
praesent luptatum zzril
delenit augue dui dolore te
feugait nulla facilisi.

Prochain cours

Langages de programmation coté client