

Les feuilles de style CSS pour documents HTML

Noureddine EL ABID AMRANI

3^{ème} ISI 2019/2020

Présentation : Qu'est ce que CSS?

- CSS : Cascading Style Sheets
- 1996 : CSS 1.0
- Langage de présentation d'un document HTML
- Résultat de la séparation entre le contenu et la présentation
- Il est encapsulé dans la page HTML (ou dans un fichier lié) pour être interprété par le client.

Objectifs

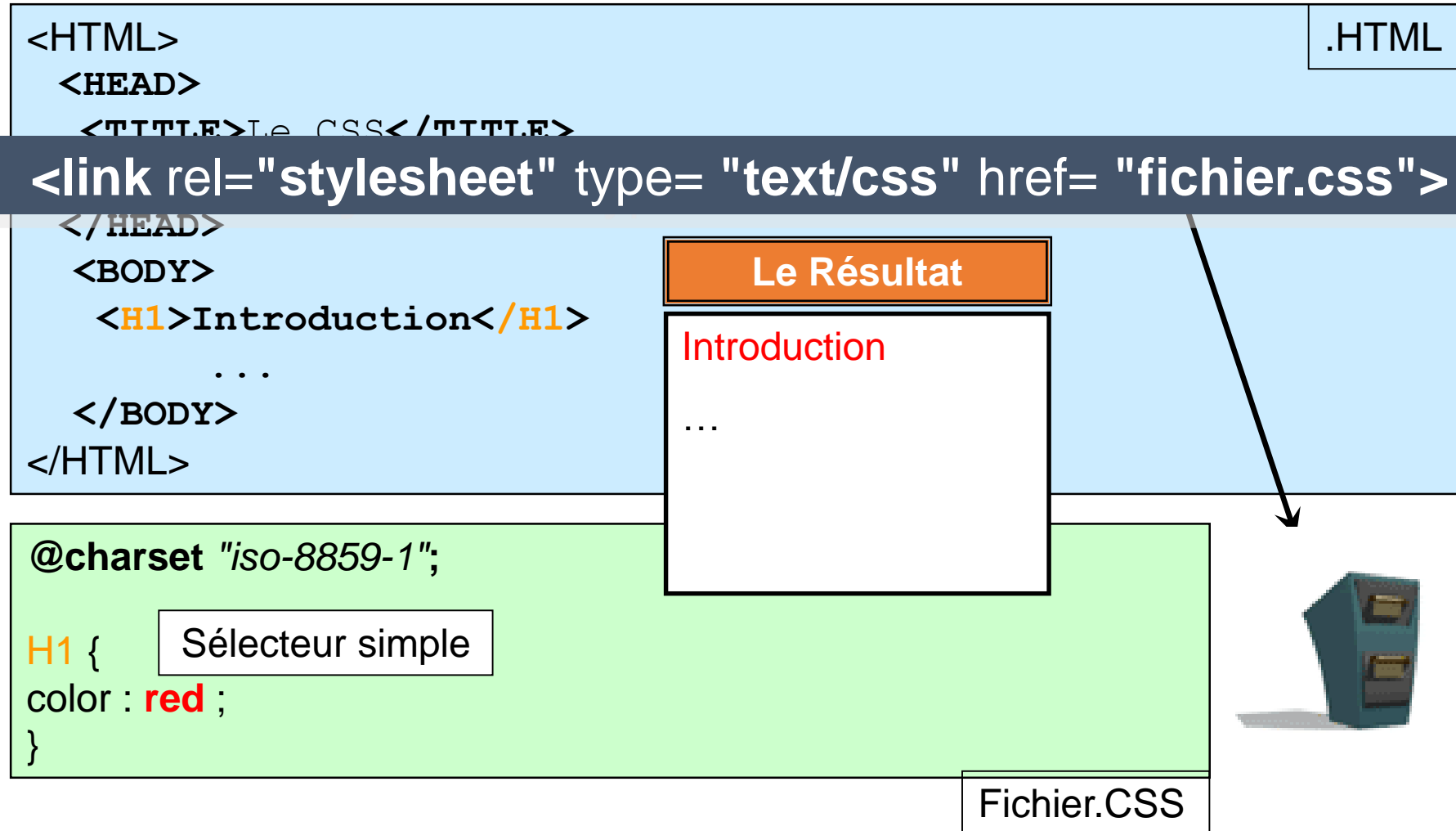
- L'un des objectifs majeurs des feuilles de style CSS est de :

Séparer la structure d'un document HTML de ses styles de présentation

Avantages

- La structure du document et la présentation peuvent être gérées dans des fichiers séparés.
- La conception d'un document se fait dans un premier temps sans se soucier de la présentation, ce qui permet d'être plus efficace.
- Dans le cas d'un site web, la présentation est uniforme : les pages web font référence aux mêmes feuilles de styles.
- Le code HTML est réduit, ne contient que les balises.

Du HTML au CSS : Principe

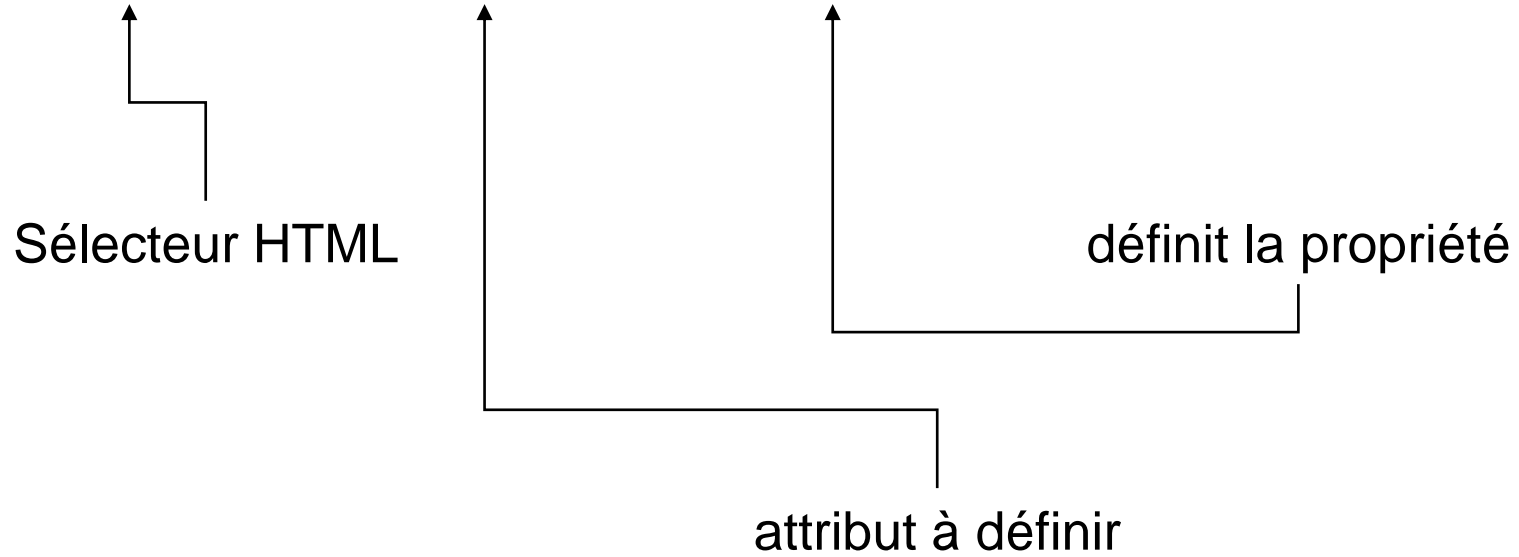


Vocabulaire : les sélecteurs

- Question : à quel type d'éléments HTML on souhaite appliquer un style?
- Réponse : un sélecteur, tout d'abord, va servir à déterminer à quel(s) élément(s) HTML ou à quel type d'éléments on souhaite appliquer un style particulier. Si l'on souhaite appliquer un style particulier à tous nos paragraphes, par exemple, on utilisera le sélecteur « p ».

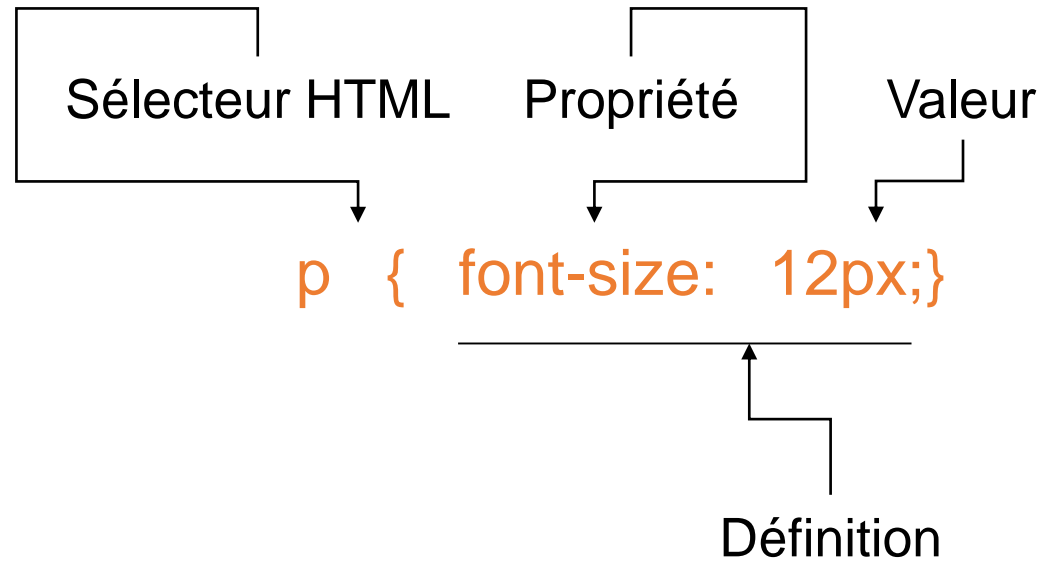
Syntaxes

Selecteur { propriété: valeur; }



Exemple

- Les styles d'une balise



Vocabulaire : les propriétés et valeurs

- Une propriété va nous servir à modifier le style d'un élément en ciblant un critère bien particulier comme la taille d'un texte, sa police ou sa couleur par exemple.
- Une valeur, enfin, va venir compléter une propriété et va en déterminer le comportement. Pour la propriété servant à changer la couleur d'un texte par exemple, la valeur va être la nouvelle couleur à appliquer.

Où écrire le CSS ?

- Nous avons trois possibilités pour écrire notre CSS :
 1. A l'intérieur de l'élément head de notre document HTML ;
 2. Dans la balise ouvrante des éléments de notre fichier HTML ;
 3. Dans un fichier portant l'extension .css séparé.

C.S.S : Où écrire le CSS ?

```
<p style="text-align: right; color: black"> Les  
feuilles de style permettent de gérer la mise en page des divers  
éléments d'une page web </p>
```

- Dans les balises

- Mais aussi...
 - Dans la page
 - Dans un fichier de style

Exemple : l'attribut :style

```
<p style="text-align: right; color: black" > Les feuilles  
de style permettent de gérer la mise en page des divers  
éléments d'une page Web </p>
```

```
<p style="text-align: right; color: black" > Bien sur, une  
page Web peut être composée de nombreux paragraphes, comme  
n'importe quel texte.</p>
```

```
<p style="text-align: right; color: black" > Mais alors, il  
devient fastidieux de devoir saisir à nouveau toutes les  
informations de style ! </p>
```

```
<p style="text-align: right; color: black" > Heureusement,  
les CSS nous permettent d'éviter ce travail inutile... </p>
```

Exemple : la balise <style>

```
<html>
  <head>
    <style type="text/css" title="mes_styles" media="all" >
      p { text-align: right; color: black }
    </style>
  </head>

  <body>
    <p> Les feuilles de style permettent de gérer la mise en
      page des divers éléments d'une page Web </p>
    <p> Bien sur, une page Web peut être composée de nombreux
      paragraphes, comme n'importe quel texte.</p>
    <p> Mais alors, il devient fastidieux de devoir saisir à
      nouveau toutes les informations de style ! </p>
    <p> Heureusement, les CSS nous permettent d'éviter ce
      travail inutile... </p>
  </body>
</html>
```

Explications : la balise <style>

```
<style
    type="text/css"
    title="mes_styles"
    media="all" >

p
{
    text-align : right;
    color : black
}
</style>
```

Voyons **<style>**
plus
en
détail

Explications : la balise <style>

```
<style
```

```
  type="text/css"
```

```
  title="mes_styles"
```

```
  media="all" >
```

```
p
```

```
{
```

```
  text-align : right;
```

```
  color : black
```

```
}
```

```
</style>
```

● Les informations de la balise sont de type texte (optionnel)

Explications : la balise <style>

```
<style
    type="text/css"
    title="mes_styles"
    media="all" >

p
{
    text-align : right;
    color : black
}

</style>
```

Nom donné (choisi) aux informations de style (optionnel)

Explications : la balise <style>

```
<style
  type="text/css"
  title="mes_styles"
  media="all" >

p
{
  text-align : right;
  color : black
}
</style>
```

Indique à quel média
s'applique la feuille de
style. (optionnel)

- all
- screen
- print
- projection
- tv
- braille

Explications : la balise <style>

```
<style  
    type="text/css"  
    title="mes_styles"  
    media="all" >
```

```
{
```

```
text-align : right;
```

```
color : black
```

```
}
```

```
</style>
```

élément pour lequel on
défini le style

délimiteurs de début et
de fin de style

Explications : la balise <style>

```
<style
    type="text/css"
    title="mes_styles"
    media="all" >
```

```
P
```

```
{
```

```
text-align : right ;
```

```
color : black
```

```
}
```

```
</style>
```

● propriété

● valeur pour
cette propriété

Explications : la balise <style>

```
<style
    type="text/css"
    title="mes_styles"
    media="all" >
```

```
P
{
```

```
    text-align : right ;
    color : black
}
```

```
h1
{
```

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

```
</style>
```

Il est possible de définir
le style de plusieurs
éléments

Lier une feuille de style externe

```
<html>

<head>

< link rel="stylesheet"
      href="style.css"
      type="text/css"
      media="screen" >

</head>
```

● c'est une
feuille de style
qui est liée

Lier une feuille de style externe

```
<html>

<head>

  < link rel="stylesheet"

    href="style.css"

    type="text/css"

    media="screen" >

</head>
```

Ou se trouve la feuille de style ?

Ici, dans le fichier `style.css`
du répertoire courant

Lier une feuille de style externe

```
<html>

<head>

  < link rel="stylesheet"

    href="style.css"

    type="text/css"

    media="screen" >

</head>
```

Une feuille de style, c'est
un fichier texte

Lier une feuille de style externe

```
<html>

<head>

  < link rel="stylesheet"

    href="style.css"

    type="text/css"

    media="screen" >
  [ ]
</head>
```

Pour quel média la feuille de style est-elle valable ?

(optionnel, par défaut : tous)

le fichier style.css

- Il comporte les informations de style sur les différents éléments

```
Fichier « style.css »  
  
P  
{  
  text-align : right ;  
  color : black  
}  
  
h1  
{ text-align : center }
```

NB : Pour des raisons de performances de votre code, de clarté et d'économie de temps, je vous recommande vivement d'utiliser la dernière méthode dès que cela est possible.

Quelques propriétés pour commencer ?

Après cette présentation générale,
voyons quelques exemples...

- body
- p
- h1, h2, h3, h4, h5, h6

body

- Couleur du texte & du fond
- Placer une image de fond
- Rajouter des marges

Un peu de couleur...

- `color`

la propriété `color` permet de fixer la couleur du texte.

- Comment manipuler les couleurs ?

- liste de couleur standard
- Un code hexadécimal long `#00FF45`
- Un code hexadécimal court `#0F3`
- Un code rgb, de 0 à 255 `rgb(0,0,100)`
- un code rgb, en pourcentage `rgb(10%,10%,30%)`

Un peu de couleur (suite)

- Liste des couleurs standard :

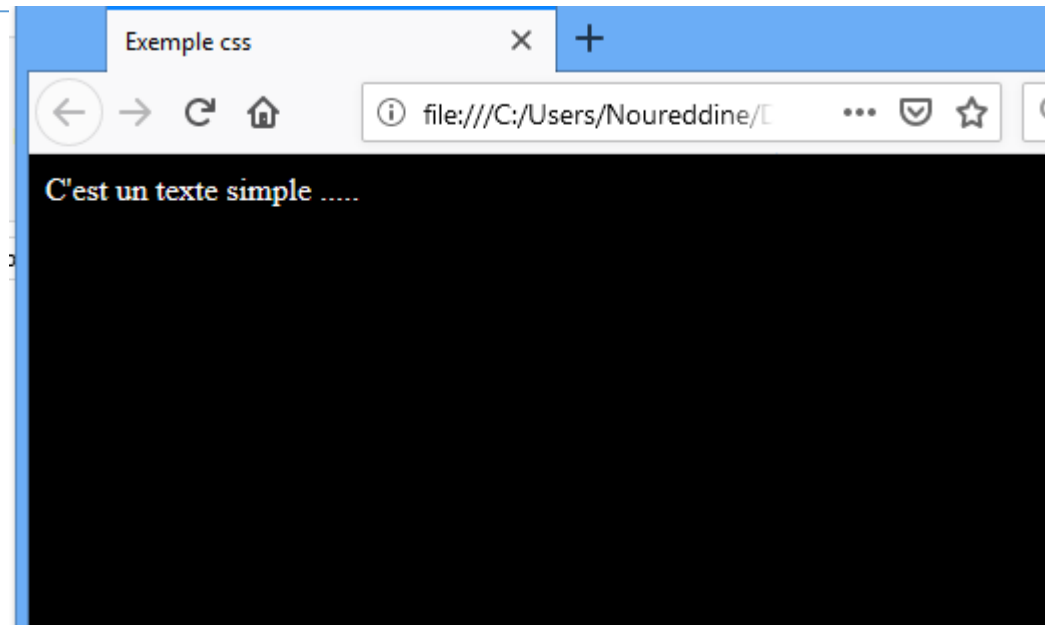
aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive,
purple, red, silver, teal, white, yellow

Un peu de couleur (de fond)

- `background-color`
`body`

la propriété `background-color` permet de fixer la couleur de fond de la page.

```
background-color : rgb(0,0,0) ;  
color : white  
}
```



Une image de fond

- `background-image`

permet de spécifier une image de fond

- `background-repeat`

l'image de fond doit-elle être répétée ?

- `background-position`

Où placer l'image de fond ?

- `background-attachment`

L'image de fond bouge-t-elle en même temps que la page ?

background-image

- none
- url()

background-repeat

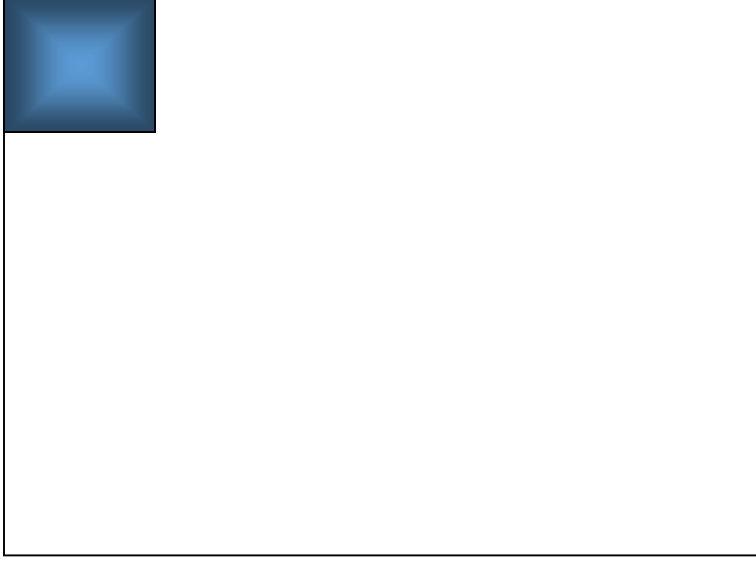
- repeat
- no-repeat
- repeat-x
- repeat-y

background-position

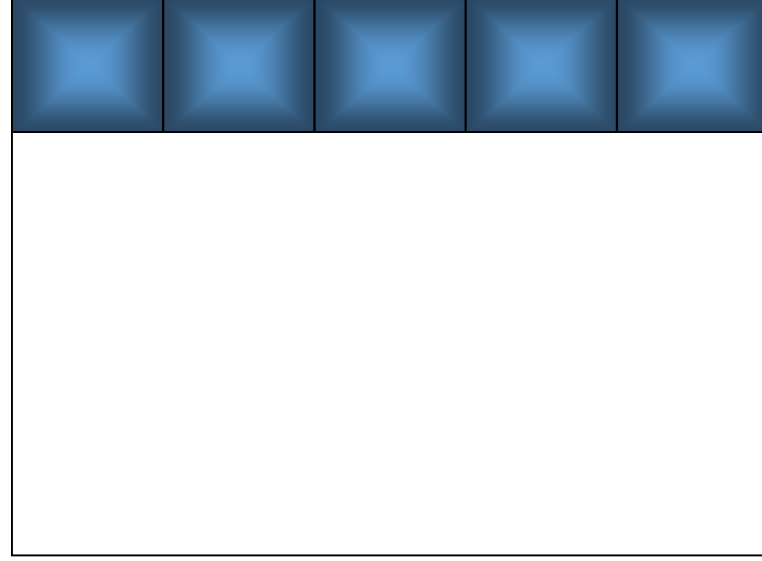
background-attachment

background-repeat

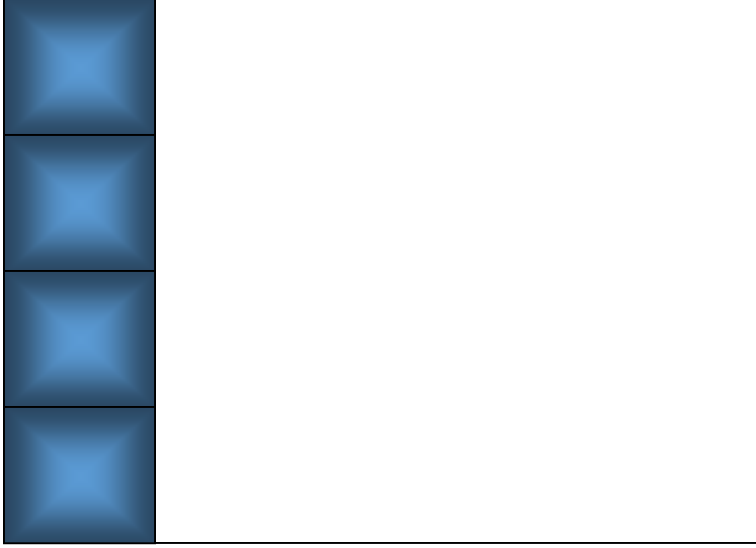
no-repeat



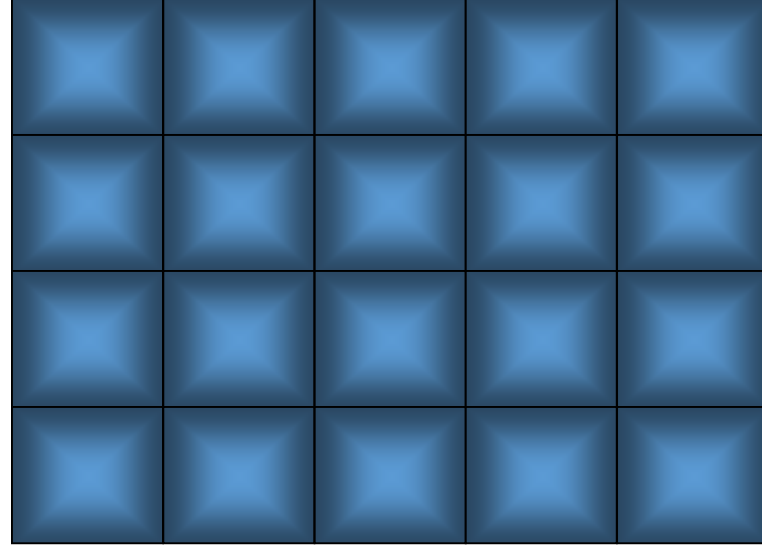
repeat-x



repeat-y

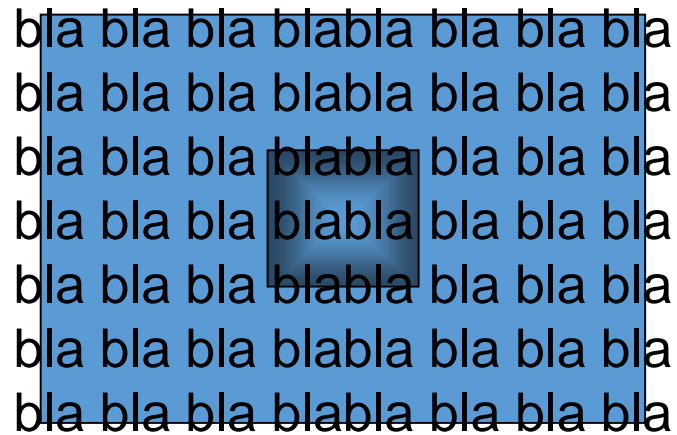


repeat



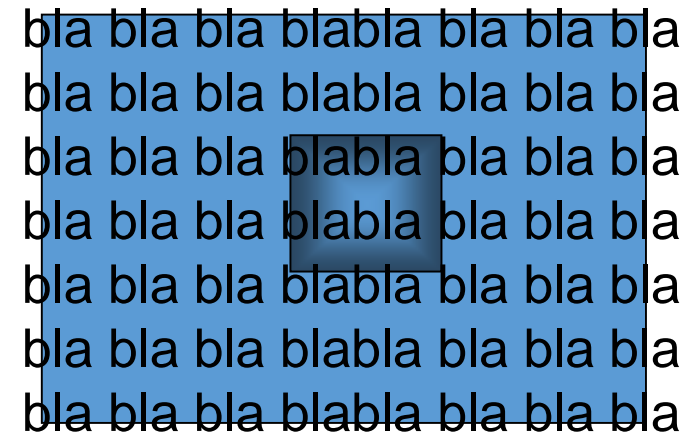
positionnement fixe ou mobile

background-attachment



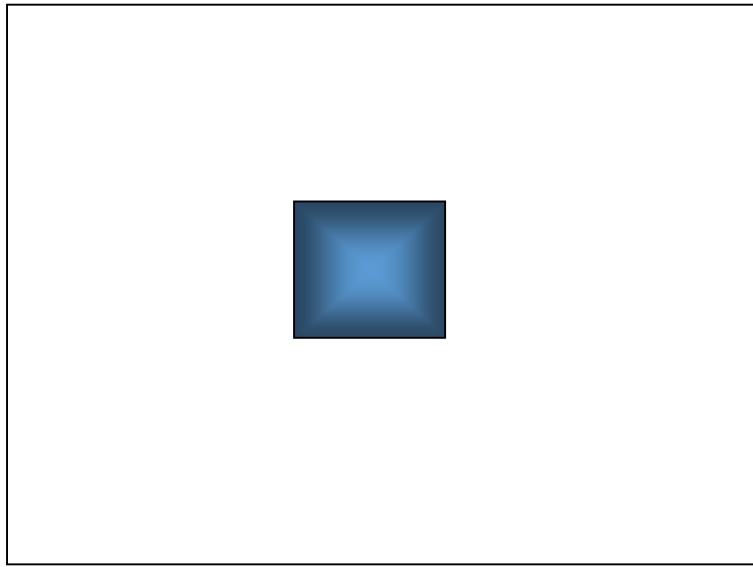
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla

```
background-attachment:  
fixed;
```



bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla
bla bla bla blabla bla bla bla

```
background-attachment:  
scroll;  
(valeur par défaut)
```

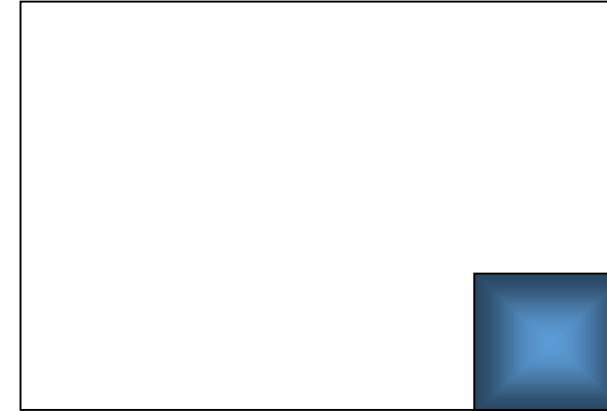
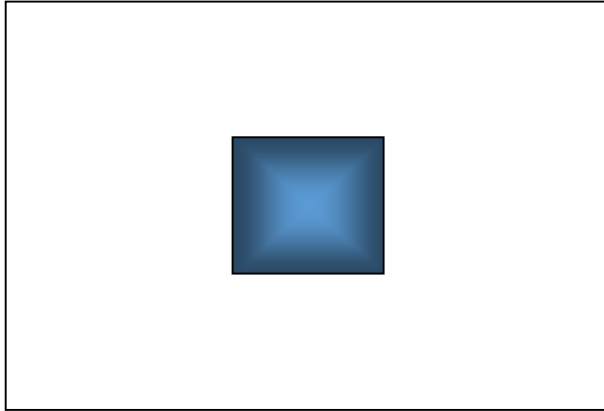


Mais au fait, comment avoir une image au milieu ?

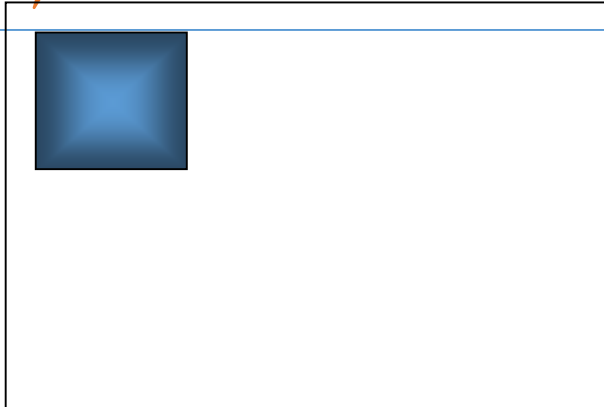
```
background-position : x y
```

Détermine la position de l'image de fond
par rapport au coin supérieur gauche

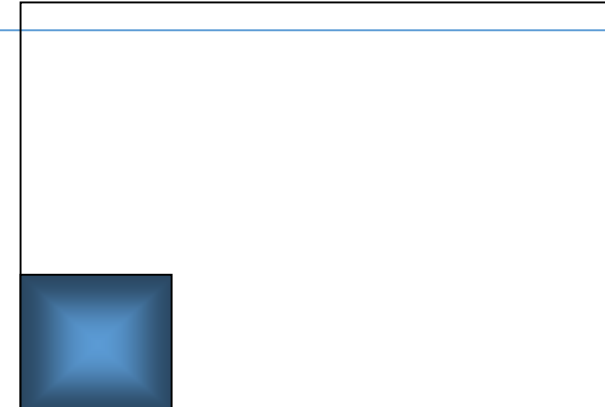
Exemples de positionnement



```
background-position: 50%  
50% ;
```



```
background-position: 100%  
100%;
```



```
background-position: 10px  
10px;
```

```
background-position: 0%  
100%;
```

Mesures en C.S.S

- % (pourcentage)
 - la taille / position de l'élément est calculée de façon relative
- px (pixel)
 - Le pixel est la plus petite unité de l'écran
- em
 - 1em = 100%, 1.2em = 120%, ...

Mesures en C.S.S

! Préciser l'unité de mesure est obligatoire !

sauf pour la valeur 0, identique quelque soit l'unité
utilisée

```
background-position: 50% 50% ; // Valide
```

```
background-position: 10px 50% ; // Valide
```

```
background-position: 0 50% ; // Valide
```

```
background-position: 10 50% ; // Non Valide
```

En C.S.S, en cas d'erreur, la propriété erronée est ignorée

Quelques propriétés sur les fontes

- `font-size`
 - permet de fixer la taille de la police
 - les unités de mesures vues précédemment sont utilisables
 - il existe aussi des tailles relatives prédéfinies :
 - `xx-small`
 - `x-small`
 - `small`
 - `medium,`
 - `large,`
 - `x-large,`
 - `xx-large,`
 - `larger,`
 - `smaller`

Quelques propriétés sur les fontes

- `font-family`
 - permet de spécifier le type de police que l'on souhaite afficher
 - ! Toutes les polices ne sont pas disponibles sur tous les ordinateurs !
 - Il est possible de spécifier plusieurs polices, dans l'ordre où l'on souhaite les utiliser

Quelques propriétés sur les fontes

- `font-family`
 - familles de polices génériques :
 - `serif`,
 - `sans-serif`,
 - `monospace`,
 - `cursive`,
 - `fantasy`.

exemple :

```
font-family : "times new roman", serif, sans-serif
```

Quelques propriétés sur les fontes

- `font-weight`
 - précise le niveau de gras de la police :
 - `normal`,
 - `bold`,
 - `100`, `200`, ...
- `font-style`
 - précise le style de fonte :
 - `normal`
 - `italic`
 - `oblic`

exemple : afficher du texte en italique et en gras

```
font-weight : bold ;  
font-style  : italic ;
```

Quelques propriétés sur les fontes

- `font-variant`
 - utilise une fonte normale ou en petite capitale :
 - `normal`,
 - `small-caps`.
- `font`
 - raccourci permettant de tout spécifier

exemple : utilisation du raccourci `font`

```
font : italic bold 1em cursive ;
```

Quelques propriétés sur les textes

- `text-align`

- permet de gérer l'alignement du texte

- `right,`
 - `left,`
 - `center,`
 - `justify.`

- `text-decoration`

- comment doit apparaître le texte

- `none` (aucun)
 - `underline` (souligné)
 - `overline` (surligné)
 - `line-through` (barré)
 - `blink` (clignotant)

Quelques propriétés sur les textes

- `text-transformation`
 - on peut appliquer des transformations au texte
 - `capitalize` (1ere lettre de chaque mot en majuscule)
 - `uppercase` (transformer en majuscule)
 - `lowercase` (transformer en minuscule)
 - `none` (aucune)
- `text-indent`
 - Indentation de début de paragraphe (taille de l'espace)

exemple :

```
P {  
  text-transformation : lowercase ;  
  text-indent          : 1em;  
}
```

On peut affecter les mêmes styles au groupe de balises :

```
h1 {text-decoration : underline }  
h2 {text-decoration : underline }  
h3 {text-decoration : underline }  
h4 {text-decoration : underline }  
h5 {text-decoration : underline }
```

peut s'écrire ...

```
h1,h2,h3,h4,h5,h6 {text-decoration : underline }
```

Changer le style

Comme dit précédemment ...

```
p {text-decoration : underline }  
strong {text-decoration : underline }
```

peut s'écrire ...

```
p,strong {text-decoration : underline }
```

Mais attention ...

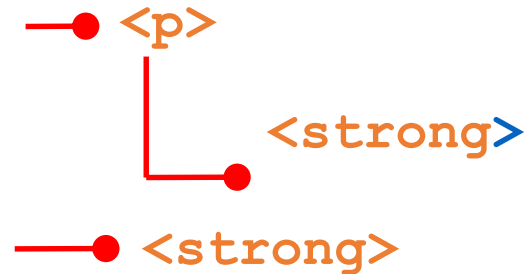
```
p strong {color : red }
```

possède un autre sens :

Le style n'est appliqué à `strong` que s'il est imbriqué dans une balise `p`

Exemple ...

```
<p>La couleur <strong>rouge</strong>  
est appliquée ici.</p>  
<strong>Mais pas la</strong>
```



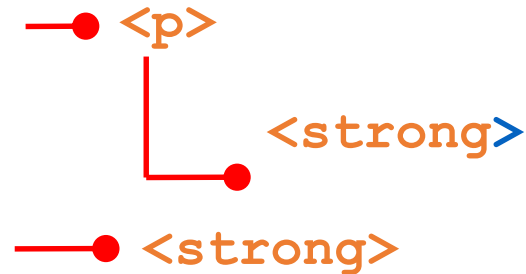
Mais attention ...

```
P>strong {color : red }
```

Le style n'est appliqué à `strong` que s'il est descendant direct de `p`

Exemple ...

```
<p>La couleur <strong>rouge</strong>  
est appliquée ici  
<strong>Et ici</strong>  
</p>
```



De même ...

```
p+strong {text-decoration : underline }
```

Le style n'est appliqué à `strong` que s'il suit immédiatement une balise `p`

Exemple ...

```
p {text-indent : 1em}  
p+p {text-indent : 0}
```

*Le premier paragraphe d'un texte sera indenté,
mais pas les suivants*

Exercice :

`` : balise multi-usage, très pratique pour appliquer un style CSS à une portion de texte.

`` : balise qui permet de mettre un texte en gras (comme la balise ``) mais avec un peu d'importance au niveau sémantique.

Les deux balises `` et `` sont très pratiques pour appliquer un style CSS à une portion de texte (`` donne de l'importance à une portion de texte de façon moins forte que ``).



Exercice :



The image shows a web browser window with a dark blue background. The address bar at the top shows a file path: file:///C:/Users/Noureddine/E... To the right of the address bar is a search bar with the text 'Rechercher'. Below the address bar, the title 'Les plateformes JADE' is displayed in large, bold, yellow letters. Underneath the title, the subtitle 'Comportements des agents dans la plate-forme JADE' is shown in yellow, underlined text. The main content area contains a paragraph in white text explaining the FIPA (Foundation for Intelligent Physical Agents) organization, founded in 1996, and its goal of producing standards for 'l'interopération d'agents logiciels hétérogènes'. This phrase is highlighted in red. Below the paragraph is a bulleted list of five items in yellow text: 'Les applications', 'Les architectures abstraites', 'Les langages d'interaction (ACL)', 'La gestion des agents', and 'Le transport des messages'.

Les plateformes JADE

Comportements des agents dans la plate-forme JADE

La [FIPA](#) (Foundation for Intelligent Physical Agents) est une organisation à but non lucratif fondée en 1996 dont l'objectif est de produire des standards pour **l'interopération d'agents logiciels hétérogènes**.

- Les applications
- Les architectures abstraites
- Les langages d'interaction (ACL)
- La gestion des agents
- Le transport des messages

Les commentaires dans CSS

- les fichiers CSS deviennent rapidement très long (beaucoup plus que les fichiers HTML) et pas efficace

```
/* Un commentaire en CSS */
```

```
H1 {  
  color : red ;  
}
```

Les limites des sélecteurs simples : les attributs class et id

- Jusqu'à présent, nous n'avons manipulé que des sélecteurs que l'on appelle « simple », car ils correspondent à des éléments HTML seuls et sans attributs (par exemple le sélecteur p).
- Question : comment faire pour appliquer un style différent à deux éléments de même type, deux paragraphes par exemple ?
- Réponse : Il faut identifier chaque balise de façon unique.
- Et c'est pour cela que l'on a créé les attributs class et id.
- Class et Id sont deux attributs HTML qui ont été créés pour pouvoir appliquer différents styles à des éléments de même type.
- Question : quelle est la différence entre id et class?

Les attributs : class et id

- Réponse :
 - L'attribut id permet d'identifier chaque élément de façon unique c. à. d. le style sera appliqué à cet élément. Donc l'attribut id avec une valeur précise ne peut être utilisé qu'une seule fois dans une page, au contraire de class.
 - L'attribut class permet également de faire l'inverse et d'appliquer le même style à différents éléments choisis (on parle des éléments qui appartiennent à la même catégorie (même classe)).
- Syntaxe :
 - Premièrement on se place dans la balise ouvrante d'un élément HTML, on écrit le nom de notre attribut (class ou id), et on lui donne une valeur cohérente.

L'attribut : id

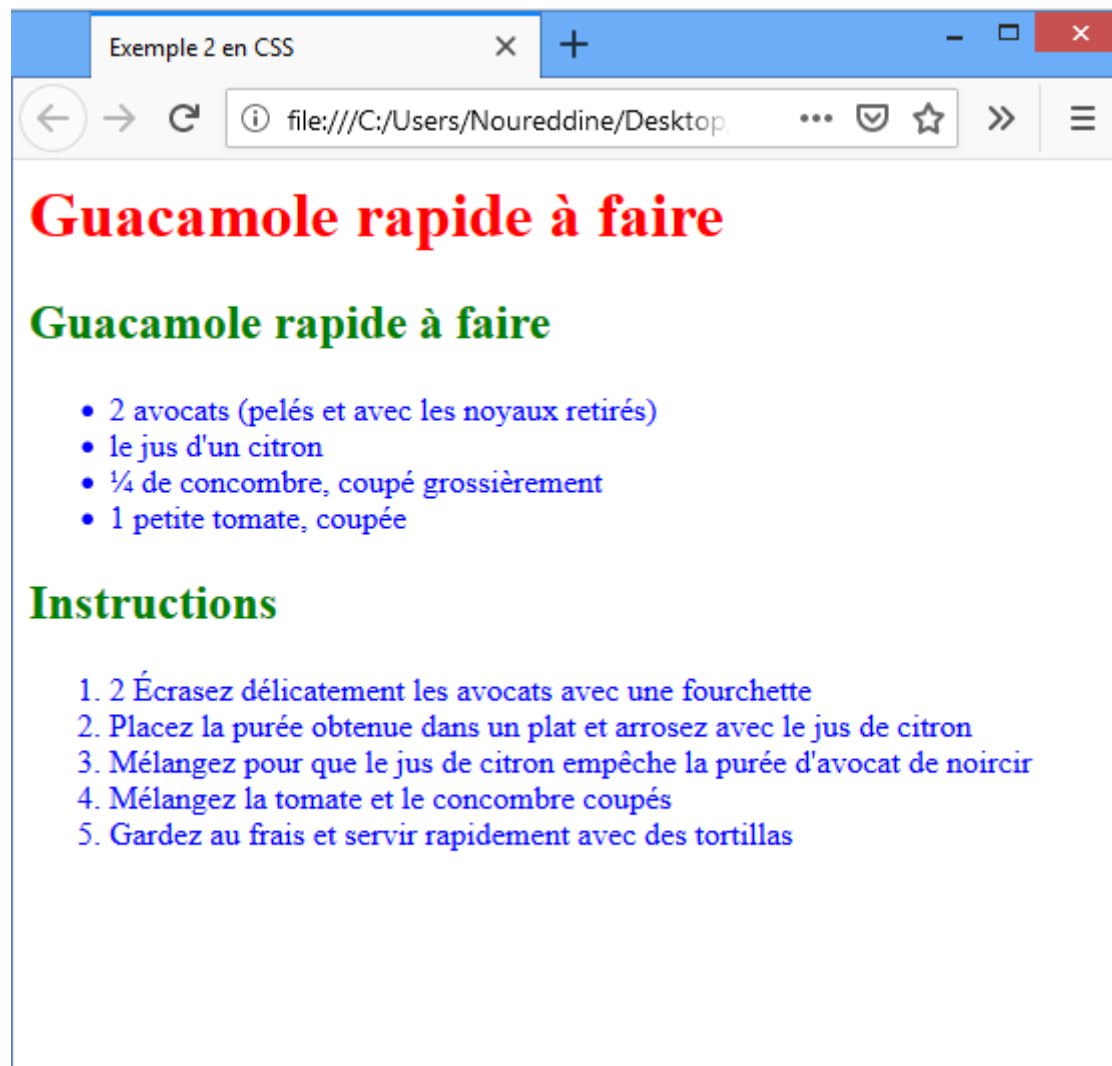
- Exemple :

```
<html>
  <head>
    <title>Exemple 2 en CSS</title>
    <link rel="stylesheet" type="text/css" href="mystyle.css" />
  </head>
  <body>
    <h1 id="id1">Guacamole rapide à faire</h1>
    <h2 id="id2">Guacamole rapide à faire</h2>
    <ul id="id3">
      <li>2 avocats (pelés et avec les noyaux retirés)</li>
      <li>le jus d'un citron</li>
      <li>¼ de concombre, coupé grossièrement</li>
      <li>1 petite tomate, coupée</li>
    </ul>
    <h2 id="id4">Instructions</h2>
    <ol id="id5">
      <li>2 Écrasez délicatement les avocats avec une fourchette</li>
      <li>Placez la purée obtenue dans un plat et arrosez avec le jus de
citron</li>
      <li>Mélangez pour que le jus de citron empêche la purée d'avocat de
noircir</li>
      <li>Mélangez la tomate et le concombre coupés</li>
      <li>Gardez au frais et servir rapidement avec des tortillas</li>
    </ol>
  </body>
</html>
```

```
#id1 {
  color : red ;
}
#id2 {
  color : green ;
}
#id3 {
  color : blue ;
}
#id4 {
  color : green ;
}
#id5 {
  color : blue ;
}
```


L'attribut : id

- Résultat :



L'attribut : class

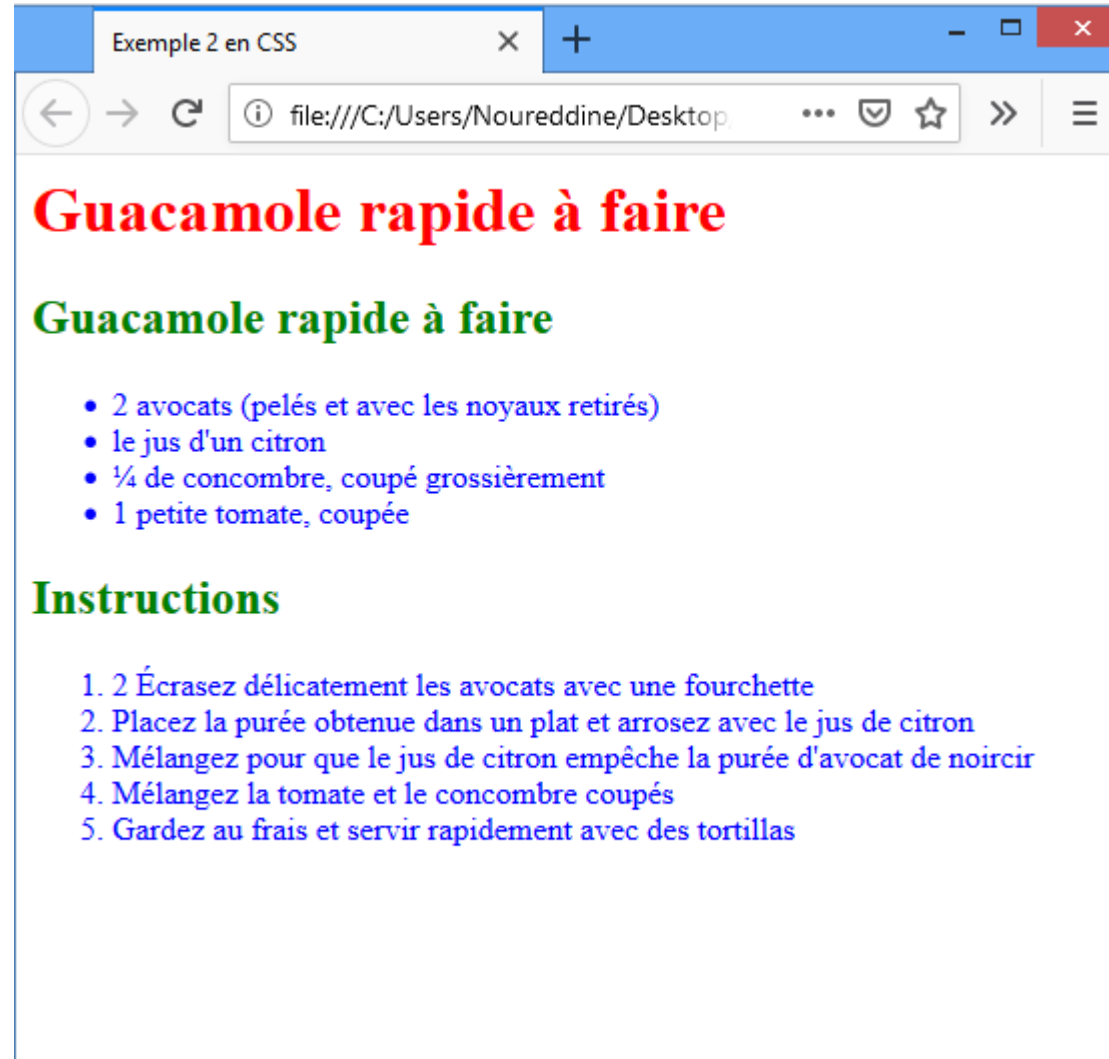
- Exemple :

```
<html>
  <head>
    <title>Exemple 2 en CSS</title>
    <link rel="stylesheet" type="text/css" href="mystyle.css" />
  </head>
  <body>
    <h1 class="titre">Guacamole rapide à faire</h1>
    <h2 class="titre">Guacamole rapide à faire</h2>
    <ul class="liste">
      <li>2 avocats (pelés et avec les noyaux retirés)</li>
      <li>le jus d'un citron</li>
      <li>¼ de concombre, coupé grossièrement</li>
      <li>1 petite tomate, coupée</li>
    </ul>
    <h2 class="titre">Instructions</h2>
    <ol class="liste">
      <li>2 Écrasez délicatement les avocats avec une fourchette</li>
      <li>Placez la purée obtenue dans un plat et arrosez avec le jus de
citron</li>
      <li>Mélangez pour que le jus de citron empêche la purée d'avocat de
noircir</li>
      <li>Mélangez la tomate et le concombre coupés</li>
      <li>Gardez au frais et servir rapidement avec des tortillas</li>
    </ol>
  </body>
</html>
```

```
.classe1 {
  color : red ;
}
.classe2 {
  color : green ;
}
.classe3 {
  color : blue ;
}
```

L'attribut : class

- Même résultat :



Exercice :

- Sachant que la propriété [list-style-type](#) définit le type de la puce :

```
list-style-type: circle;  
list-style-type: square;  
list-style-type: upper-roman;  
list-style-type: lower-alpha;
```

- Créer les listes imbriquées suivantes :

- I. Introduction
- II. Chapitre1
 - Eléments et attributs
 - a. Elements
 - b. Attributs
 - Titres et formatage du texte
 - a. Titres
 - b. Formatage du texte
- III. Chapitre2
 - Insérer une photo ou une vidéo dans une page web
 - a. Une photo
 - b. Une vidéo