

CHAPITRE 3

LANGAGE CSS

PLAN DU CHAPITRE

1. QU'EST CE QUE CSS
2. LIER LE FICHIER HTML AU FICHIER CSS
3. SELECTEUR ET PROPRIETE
4. ATTRIBUT ID ET CLASS
5. TEXTE
6. COMMENTAIRE
7. MARGES ET PADDING
8. BLOCK ET INLINE
9. ELEMENT DIV
10. FLEXBOX
11. PARTIE FIXE

1. QU'EST-CE QUE CSS

- ✗ **CSS** est le diminutif de Cascading StyleSheets, ou feuilles de styles en cascade.
- ✗ Le CSS a été créé en 1996 et a pour rôle de mettre en forme du contenu en lui appliquant ce qu'on appelle des styles.
- ✗ On utilise le CSS sur le code HTML, afin d'enjoliver le résultat visuel final.
- ✗ Le CSS permet par exemple de définir la taille, la couleur ou l'alignement d'un texte.

2. COMMENT LIER LE FICHIER HTML AU FICHIER CSS

Il faut « lier » le fichier HTML au fichier CSS pour indiquer au navigateur qu'il doit appliquer les styles contenus dans le fichier **cours.css** au fichier **cours.html**.

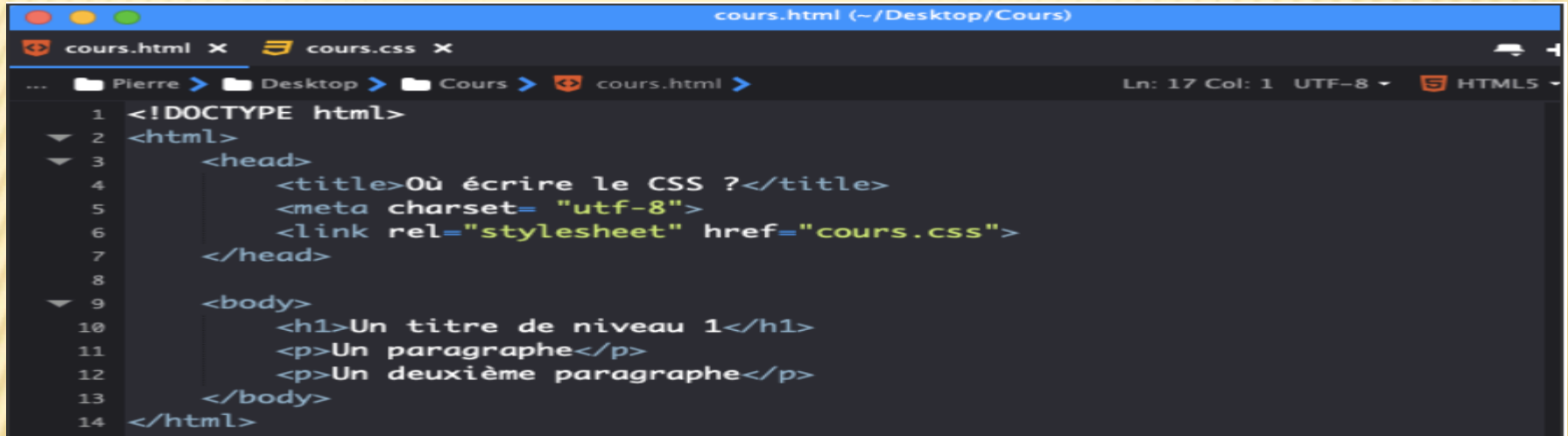
Pour cela, il faut utiliser un nouvel élément HTML : l'élément link (« lien », en français). On va placer l'élément link au sein de l'élément head du fichier HTML.

Cet élément se présente sous la forme d'une balise orpheline :

```
<link rel="stylesheet" href="cours.css">
```

EXEMPLE

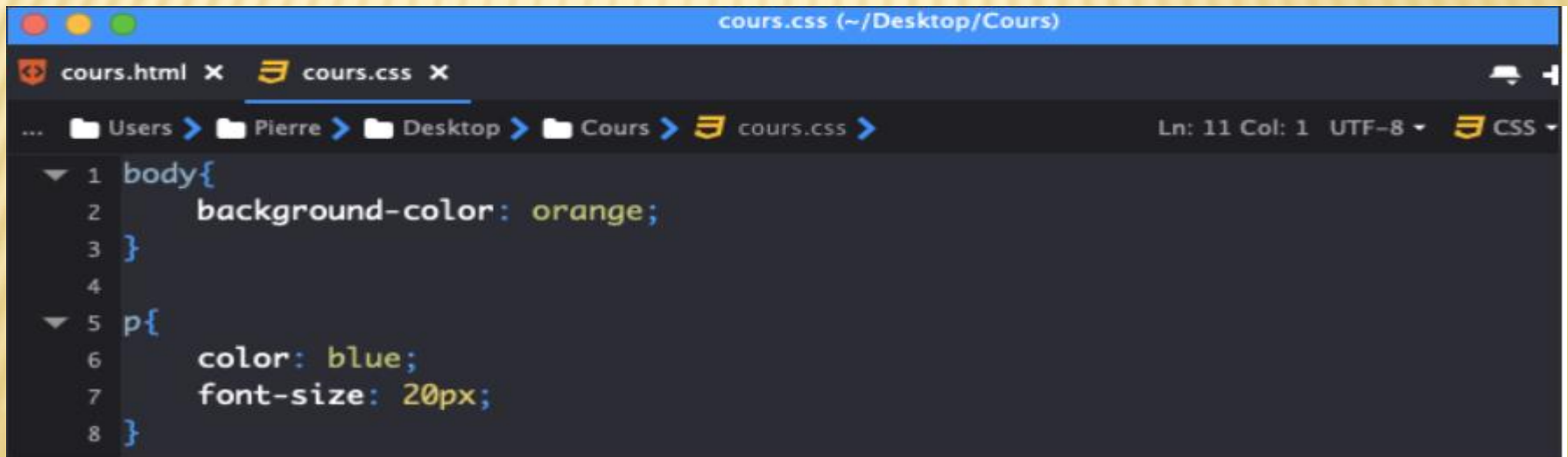
cours.html



The screenshot shows a code editor window titled 'cours.html (~/Desktop/Cours)'. The editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Où écrire le CSS ?</title>
5     <meta charset= "utf-8">
6     <link rel="stylesheet" href="cours.css">
7   </head>
8
9   <body>
10    <h1>Un titre de niveau 1</h1>
11    <p>Un paragraphe</p>
12    <p>Un deuxième paragraphe</p>
13  </body>
14 </html>
```

cours.css



The screenshot shows a code editor window titled 'cours.css (~/Desktop/Cours)'. The editor displays the following CSS code:

```
1 body{
2   background-color: orange;
3 }
4
5 p{
6   color: blue;
7   font-size: 20px;
8 }
```

EXEMPLE



3. SELECTEUR ET PROPRIÉTÉ

- ✖ **Les sélecteurs CSS simples**

Un sélecteur va donc nous permettre de cibler un ou plusieurs éléments HTML afin de leur appliquer un style particulier.

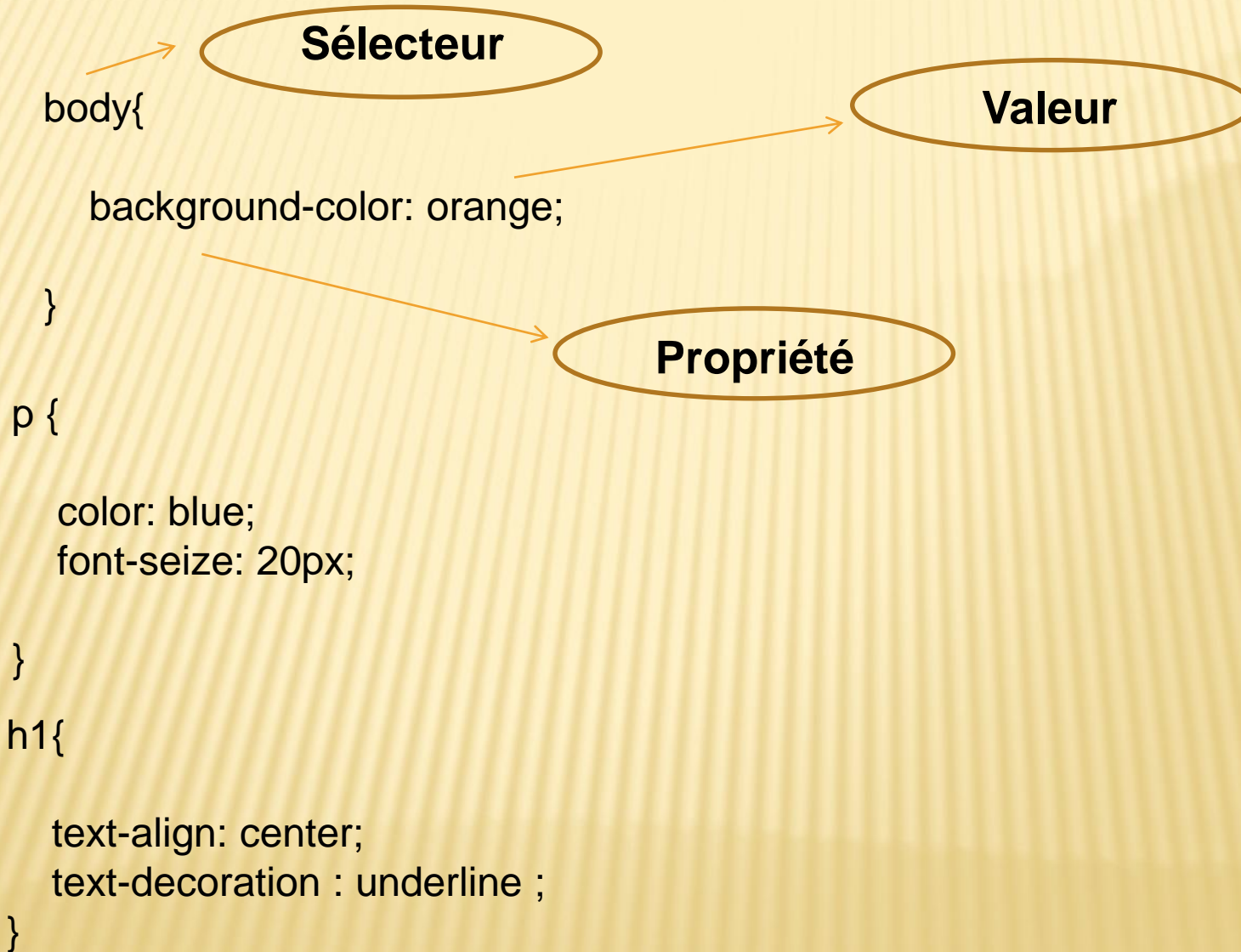
- ✖ **Les propriétés CSS**

Les propriétés vont nous permettre de choisir quel(s) aspect(s) (ou "styles") d'un élément HTML on souhaite modifier.

- ✖ **Les valeurs CSS**

Une propriété va être accompagnée d'une ou plusieurs valeurs qui vont définir le comportement de cette propriété.

EXEMPLE



4. ATRIBUTS ID ET CLASS

Les sélecteurs #id et .class vont nous permettre de cibler un élément en particulier plutôt qu'un type d'élément.

```
/*L'élément portant l'id "p1" sera en bleu*/  
#p1{  
    color: blue;  
}  
  
/*L'élément portant la class "p1" sera en rouge*/  
.p1{  
    color: red;  
}
```

Il existe une différence entre les deux attributs class et id : chaque id doit avoir une valeur unique tandis que plusieurs attributs class peuvent posséder la même valeur.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Sélecteurs CSS</title>  
    <meta charset= "utf-8">  
    <link rel="stylesheet" href="styles.css">  
  </head>  
  
  <body>  
    <!--On donne la même class à notre titre et à un paragraphe-->  
    <h1 class="p1">Un titre de niveau 1</h1>  
    <p id="p1">Un paragraphe contenant</p>  
    <p class="p1">Un deuxième paragraphe</p>  
    <p>Un troisième paragraphe</p>  
  </body>  
</html>
```

CLASS

```
.p1{  
  color: blue;  
  background-color: silver;  
  font-size: 14px;  
  text-align: center;  
}
```

```
.p2{  
  color: blue;  
  background-color: yellow;  
  font-size: 14px;  
  text-align: center;  
}
```

```
.p3{  
  color: blue;  
  background-color: green;  
  font-size: 14px;  
  text-align: center;  
}
```

```
#p1{  
  color: blue;  
  background-color: silver;  
  font-size: 14px;  
  text-align: center;  
}
```

```
#p2{  
  color: silver;  
  background-color: yellow;  
  font-size: 14px;  
  text-align: center;  
}
```

```
#p3{  
  color: purple;  
  background-color: green;  
  font-size: 14px;  
  text-align: center;  
}
```


5. TEXTE

La police, la taille du texte, gras ou pas, tous ces paramètres sont réglables en CSS.

font-family: Famille de police

font-size: Taille de votre police de caractère

font-style:, italique, oblique...

font-weight: Gras,

```
.p1{
```

```
    font-family: "Times New Roman" ;
```

```
    font-size: 14px;
```

```
    font-style: italic;
```

```
    font-weight: bold;
```

```
}
```


5. TEXTE

L'alignement , la mise en majuscules / minuscules, la décoration et l'ombre tous ces paramètres sont réglables en CSS.

text-align: gère l'alignement (left, right, center , justify)

text-transform: gère la mise en majuscules /minuscules (Lowercase, Uppercase, Capitalize, ...)

text-decoration:, gère le soulignement (underline, overline, none,)

text-shadow: gère les ombres

```
.p2{
```

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

```
text-transform: uppercase;
```

```
text-decoration: underline;
```

```
text-shadow: 1px 1px 4px silver;
```

```
}
```

6. COMMENTAIRE

Tout comme nous avons vu qu'on pouvait écrire des commentaires en HTML, nous allons également pouvoir **commenter notre code CSS**.

Il est donc essentiel de bien organiser et de bien commenter son code CSS afin de ne pas faire d'erreur en appliquant par exemple deux styles différents à un même élément

```
/* P1 est utilisé pour faire pour le titre sur la page Accueil */  
.p1{
```

```
text-align: center;  
text-transform: uppercase;  
text-decoration: underline;  
text-shadow: 1px 1px 4px silver;
```

```
}
```

7. LES MARGES ET LES PADDING

En CSS, nous allons devoir distinguer deux types de marges : les marges intérieures (padding) et les marges extérieures (margin).

Les marges intérieures (padding) se trouvent entre le contenu de l'élément et sa bordure. On peut définir des marges intérieures haute, droite, basse et gauche de tailles diverses avec padding-top, padding-right, padding-bottom et padding-left.

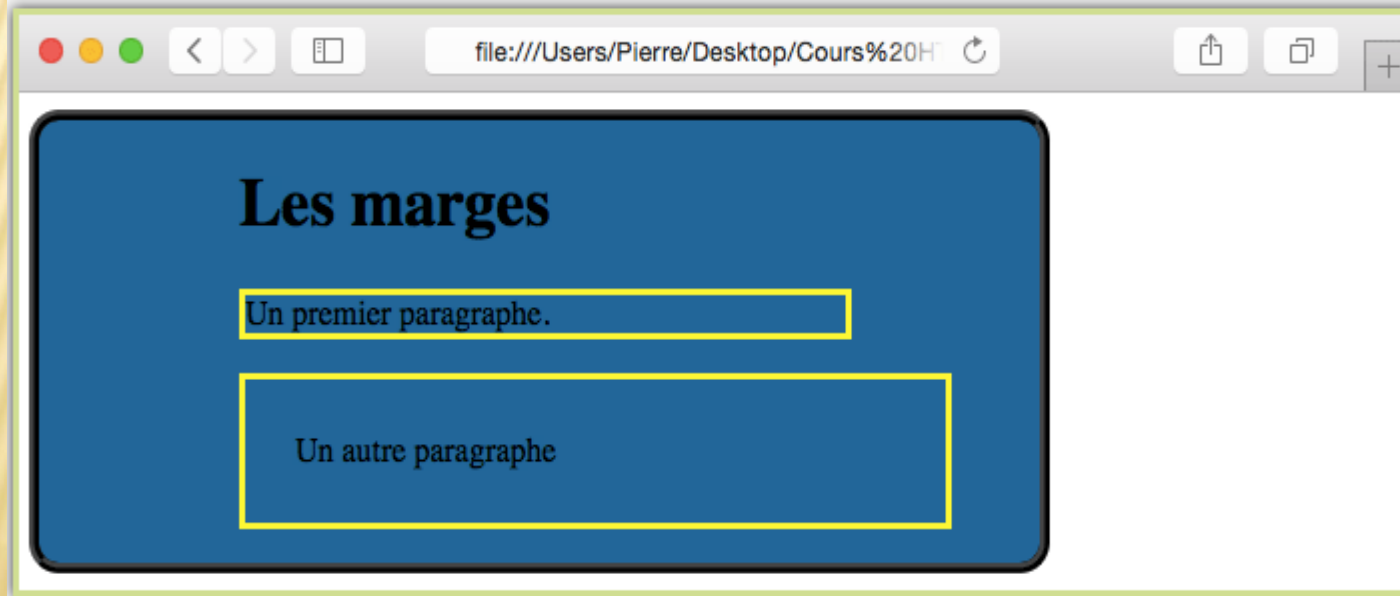
```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les marges</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
  border-radius: 15px;
  padding-left: 100px;
}

.para1{
  width: 300px;
  border: 3px solid yellow;
}

.para2{
  width: 300px;
  border: 3px solid yellow;
  padding: 25px;
}
```



7. LES MARGES ET LES PADDING

Les marges extérieures (margin), au contraire, vont définir l'espace autour d'un élément.

On peut définir des marges extérieures haute, droite, basse et gauche d'importances diverses grâce à margin-top, margin-right, margin-bottom et margin-left.

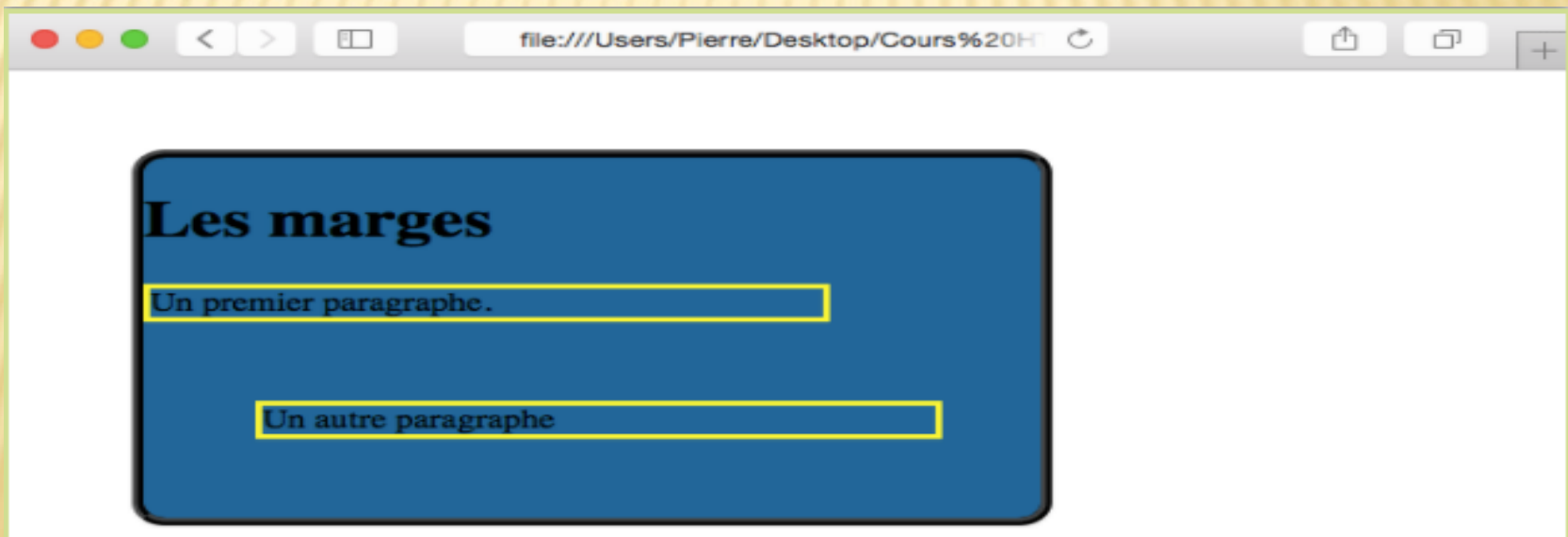
```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les marges</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
  border-radius: 15px;
  margin-top: 50px;
  margin-left: 50px;
}

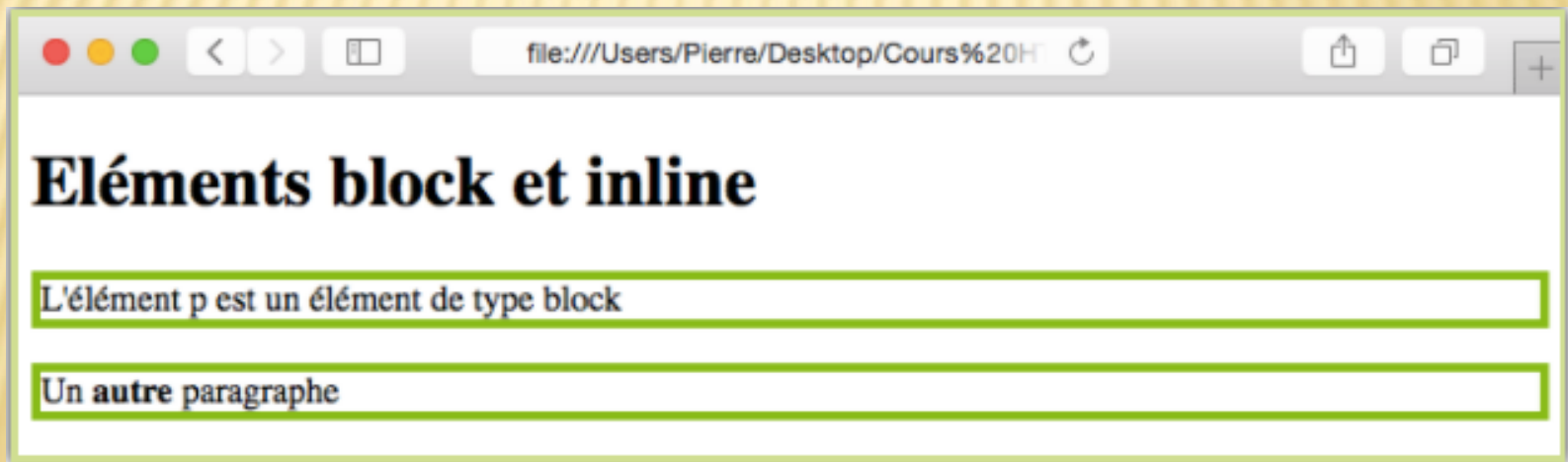
.para1{
  width: 300px;
  border: 3px solid yellow;
}

.para2{
  width: 300px;
  border: 3px solid yellow;
  margin: 50px;
}
```



8. TYPE BLOCK ET TYPE INLINE

- ✗ En HTML, tout élément est soit de type « block » (bloc), soit de type « inline » (en ligne).
- ✗ Les éléments de type block
 - + Un élément de type block va toujours commencer sur une nouvelle ligne et prendre toute la largeur disponible dans la page.
 - + De plus, un élément de type block peut contenir d'autres éléments de type block ainsi que des éléments de type inline.



8. TYPE BLOCK ET TYPE INLINE

- ✕ Les éléments HTML de type block les plus communs sont les suivants :
 - + L'élément p ;
 - + Les éléments h1, h2, etc. ;
 - + Les éléments ol et ul ;
 - + L'élément form (utilisé pour créer des formulaires) ;
 - + L'élément div.

8. TYPE BLOCK ET TYPE INLINE

✗ Les éléments de type inline

- + Au contraire des éléments de type block, les éléments de type inline ne vont pas commencer sur une nouvelle ligne mais s'insérer dans la ligne actuelle.
- + Les éléments de type inline prennent uniquement la largeur qui leur est nécessaire (c'est-à-dire la largeur de leur contenu).



8. TYPE BLOCK ET TYPE INLINE

- ✕ Les éléments HTML de type inline les plus communs sont les suivants :
 - + Les éléments strong et em ;
 - + L'élément a ;
 - + L'élément img ;
 - + L'élément span .

9. ÉLÉMENT DIV

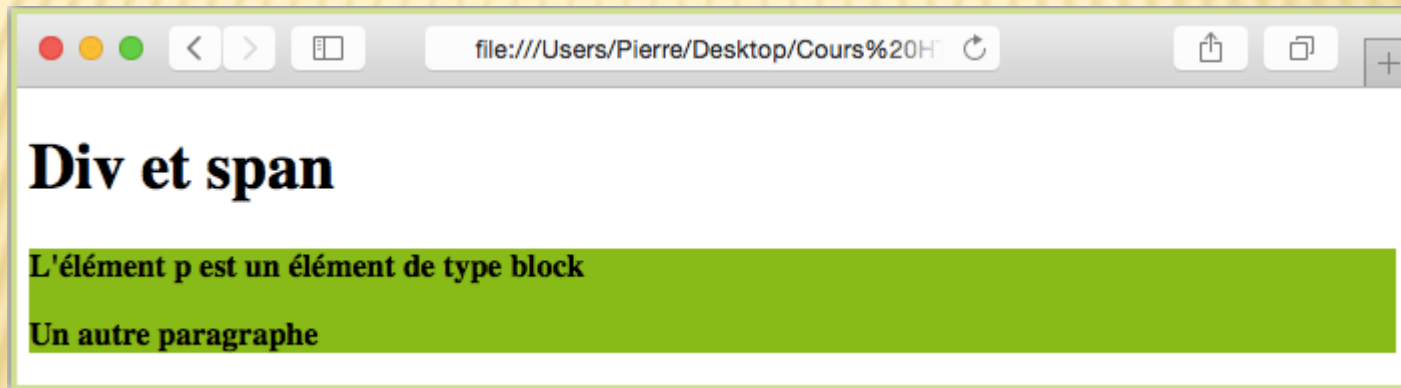
- ✖ L'élément HTML div est un élément de type block. Cet élément va souvent être utilisé comme conteneur pour plusieurs autres éléments HTML.
- ✖ Il est très commun d'attribuer un attribut class à un élément div afin de pouvoir le cibler plus facilement.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Éléments div et span</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Div et span</h1>

    <div class="div-para">
      <p>L'élément p est un élément de type block</p>
      <p>Un <strong>autre</strong> paragraphe</p>
    </div>
  </body>
</html>
```

```
.div-para{  
  background-color: #88BB11;  
  font-weight: bold;  
}
```



10. FLEXBOX

- ✖ Le flexbox est un modèle de disposition très puissant qui permet de contrôler facilement et avec précision l'alignement, la direction, l'ordre et la taille de nos éléments (ou plus précisément de nos boîtes).
- ✖ Un conteneur flexible est défini en attribuant un **display : flex** à un élément. Tous les éléments directement contenus dans ce conteneur (c'est-à-dire tous les enfants directs) vont alors automatiquement devenir des éléments flexibles.
- ✖ La propriété **flex-direction** permet de définir quel va être l'axe principal pour un conteneur flexible et ses éléments flexibles ainsi que la direction des éléments le long de cet axe.
- ✖ La propriété **flex-grow** permet de définir la capacité des éléments à s'étirer dans leur conteneur pour remplir l'espace vide. Nous allons à nouveau l'appliquer aux éléments flexibles.
- ✖ La propriété **flex-wrap** donne la possibilité aux éléments flexibles d'aller à la ligne ou en colonne s'ils dépassent de leur conteneur.

FLEXBOX

HTML

CSS

Result

EDIT ON
CODEPEN

```
.conteneur-flexible{
  display: flex;
  flex-wrap: wrap;
  align-items: center;
  justify-content: space-around;
  align-content: space-around;
  background-color: #0AD; /*Bleu*/
  width: 100%;
  height: 200px;
  border: 2px solid blue;
  box-sizing: border-box;
  margin: 20px 0px;
}
.ligne{
  flex-direction: row; /*Axe principal = axe horizontal*/
}
.colonne{
  flex-direction: column; /*Axe principal = axe vertical*/
}
.element-flexible{
  flex: 1 1 50px;
  width: 40%;
  height: 40%;
  background-color: #0DA; /*Vert*/
  border: 2px solid green;
  box-sizing: border-box;
  margin: 10px;
}
```

Resources

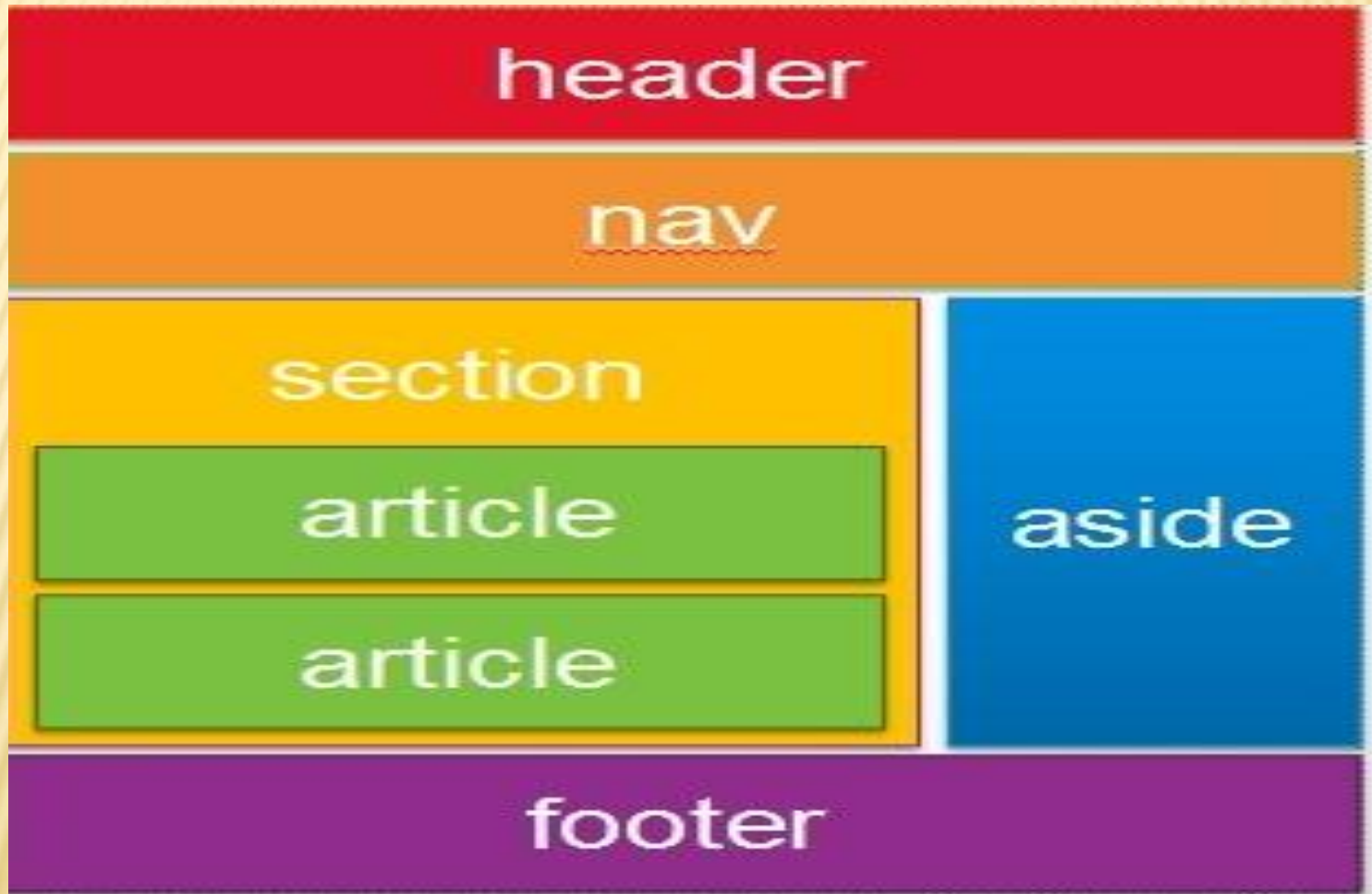
Privacy & Cookies

11. PARTIE FIXE DE SITE WEB

- ✗ Le rôle des éléments structurants est de diviser le contenu d'une page en différentes parties selon un ordre logique.
- ✗ Ces éléments sont de type block et peuvent ainsi remplacer certains de nos éléments div.

Nom de l'élément	Cas d'utilisation
header	Utilisé pour définir le haut de page (menu exclus)
nav	Utilisé pour définir le menu principal de navigation d'une page
section	Représente une section dans une page, c'est-à-dire un groupement thématique de contenu
article	Représente une partie de page qui se suffit à elle même, comme un post sur un blog par exemple
aside	Représente une section de page indirectement liée aux autres, comme une mini-biographie d'un auteur sur un blog par exemple
footer	Utilisé pour définir le pied d'une page HTML

11. PARTIE FIXE DE SITE WEB





LANGAGE HTML

1. Introduction

HTML est le langage universel utilisé pour communiquer sur le Web. Votre information sera ainsi transportée sur cette gigantesque toile de réseaux interconnectés qu'est Internet, pour aboutir sur l'ordinateur de votre lecteur grâce à un programme appelé navigateur ou browser. Ce logiciel, que l'on appelle un browser ou navigateur, vous permet de surfer sur le Net et d'afficher sur votre écran les pages qu'il a interceptées. Il y a, hélas, beaucoup de marques et de types de browsers différents. Des simples, des archaïques ou des sophistiqués... Les plus connus sont FireFox de Mozilla, Internet Explorer de Microsoft et Netscape dans sa version 2, 3 et 4.

2. Qu'est ce que le langage html

Le langage Html étant un ensemble de balises et d'attributs, il nous paraît utile sinon indispensable de les passer en revue et surtout de les visionner au moins une fois car :

- si les éditeurs Html vous faciliteront grandement la tâche, ils ne sont pas toujours parfaits surtout lors des modifications, annulations ou suppressions en cours de travail. Il vous faudra bien alors vous plonger dans le code source pour corriger les dysfonctionnements.
- les codes source de vos pages préférées sont disponibles (et sans copyright). Il est alors possible de s'en inspirer pour reprendre le procédé sans avoir à réinventer.
- ces mêmes éditeurs Html vous proposeront des termes comme "En-tête, Heading, Cell spacing, Numered List..." qui sont propres au langage Html.
- vous aurez besoin d'une connaissance pointue du Html pour inclure les codes du PHP, Javascript ou VBscript dans vos pages.

3. Premières balises

```
<HTML> Ceci est le début d'un document de type HTML.  
</HTML> Ceci est la fin d'un document de type HTML.  
<HEAD> Ceci est le début de la zone d'en-tête (prologue au document proprement dit contenant des informations destinées au browser).  
</HEAD> Ceci est la fin de la zone d'en-tête.  
<TITLE> Ceci est le début du titre de la page.  
</TITLE> Ceci est la fin du titre de la page.  
<BODY> Ceci est le début du document proprement dit.  
</BODY> Ceci est la fin du document proprement dit.
```

- Vous aurez remarqué qu'à chaque balise de début d'une action, soit <...>, correspond (en toute logique) une balise de fin d'une action .
- Vous noterez aussi que les balises ne sont pas "case sensitive". Il est donc équivalent d'écrire , , , etc

4. Premier pas Html:

- Ecrire les codes Html suivants:

```
<HTML>  
<HEAD>  
<TITLE>premier pas </TITLE>  
</HEAD>  
<BODY> bienvenue sur ma page web  
</BODY>  
</HTML>
```

-Enregistrer le document avec l'extension .html ou .htm. -cliquer sur la page web pour visualiser

