

Test

- 1. What is the previous version of HTML, prior to HTML5?
 - A. HTML 4.1
 - B. HTML 4
 - C. HTML 4.01
 - D. HTML 4.9

Test

- 2. Which doctype is correct for HTML5?
 - A. `<!DOCTYPE HTML5>`
 - B. `<!DOCTYPE html>`
 - C. `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0//EN" "http://www.w3.org/TR/html5/strict.dtd">`

Test

- 3. Who is making the Web standards?
 - A. Mozilla
 - B. The World Wide Web Consortium
 - C. Google
 - D. Microsoft

Test

- 4. Choose the correct HTML tag for the largest heading
 - A. `<h1>`
 - B. `<heading>`
 - C. `<head>`
 - D. `<h6>`

Test

- 5. What is the correct HTML tag for inserting a line break?
 - A. `<break>`
 - B. `
`
 - C. `<lb>`

Test

- 6. Choose the correct HTML tag to make a text bold
 - A. ``
 - B. `<bold>`

Test

- What is the correct HTML for creating a hyperlink?
 - A. `W3Schools.com
`
 - B. `<a>http://www.w3schools.com`
 - C. `W3Schools.c
om`
 - D. `W3Schools`

Test

- 8. Which of these tags are all <table> tags?
 - A. <table><head><tfoot>
 - B. <table><tr><td>
 - C. <thead><tbody><tr>
 - D. <table><tr><td>

Test

- 9. What is the correct HTML for inserting an image?
 - A. ``
 - B. `<image src="image.gif" alt="MyImage">`
 - C. `image.gif`
 - D. ``

Test

- 10. What is the preferred way for adding a background color in HTML?
 - A. `<body style="background-color:yellow;">`
 - B. `<background>yellow</background>`
 - C. `<body background="yellow">`

Ismail hadjadj

Log.ismail111@gmail.com

Programmation Web Coté Client : Feuilles de style CSS



31/01/2017

Feuilles De Style CSS



11

Introduction

- Pour maîtriser l'aspect final à l'écran on utilise un autre langage que le HTML: le langage CSS
- version actuelle : CSS3
- CSS = Cascading Style Sheets
= feuilles de style en cascade

Introduction

- Exemple d'un site sans css ni mise en forme (mise en page par tableau)



Introduction

- On veut obtenir quelque chose comme



Introduction

- Le code html est modifié
- Il faut coller ce code dans toutes les pages du site
- On recommence pour les fonds de tableaux etc.
- Les espacements, les fontes, les couleurs de texte, ...

Introduction

- Base de HTML :
 - Documents : fond, peu de mise en forme
- Mise en forme des débuts
 - Couleurs, alignements, ..., décrits dans les balises
 - Texte transformé en image
 - Images pour gérer les espacements
 - Tableaux utilisés à outrance
- ➔ Mise en forme figée, très lourde à modifier
- ➔ Mise en forme peu portable
- ➔ Pages complexes
- ➔ Nécessité d'une méthode alternative

Introduction

- Comment modifier toutes les polices de caractères de la page index.htm en une seule opération ?
- Comment appliquer ces modifications automatiquement aux autres pages ?
- Réponses (imparfaites) :
 - En utilisant l'outil de recherche/remplacement automatique de Dreamweaver sur l'ensemble du site (pas toujours facile)
 - En utilisant un logiciel de recherche/remplacement automatique qui peut travailler avec plusieurs fichiers texte.
- **Bonne réponse: utiliser une feuille de style (CSS)**

CSS : Cascading Style Sheets

- Positionnement flexible des styles:
 - Fichier séparé
 - Au début du document
 - ~~Dans les balises à mettre en forme~~
- Héritage et cascade
- Dépendance au média
 - Écran, imprimante, Braille, ...
- Styles alternatifs
- Rendu plus rapide que mise en forme historique
- Modification de style très aisée

Modification de style très aisée

■ Exemple de deux feuilles de styles appliquées



Appliquer un style à un élément

- Appliquer à une balise
 - Redéfinir le style de toutes les instances
- Selon les liens de parenté
 - Descendant, enfant, ...
- Pseudo-classes
 - Liées à l'état de l'élément
- En fonction des attributs de l'élément
 - Valeur de l'attribut
 - Attribut **class**: Affecter des éléments
- Élément identifié
 - Identifier un élément HTML ?

Vision basique de la mise en forme

Relations entre structure du

Vision légèrement dynamique et contextuelle

Affiner la vision basique de la mise en forme

Classe, identification et description

■ Affecter une balise à une classe

- Attribut **class**

- `<p class="ma_classe">`

Sous-groupe de l'ensemble des balises `<p>`

■ Donner un identifiant à un

- Attribut **id**

- Identifiant doit être unique (charge du concepteur)

- `<p id="mon_id_unique">`

Une balise `<p>` repérée de façon unique

■ Décrire un élément

- Attribut **title** pour la plupart
- Texte affiché sous forme d'info-bulle (tooltip)

Ajouter du style à un document HTML

■ Dans le document

```
>head <
>"style type="text/css" <
--|>
Mon style
<--
>style </
```

■ Fichier externe

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

Feuille de style

```
sélecteur {  
    propriété : valeur ;  
[ propriété : valeur ; ]  
    ...  
    /* Commentaire */  
}  
  
p {  
    color : blue ;  
}
```

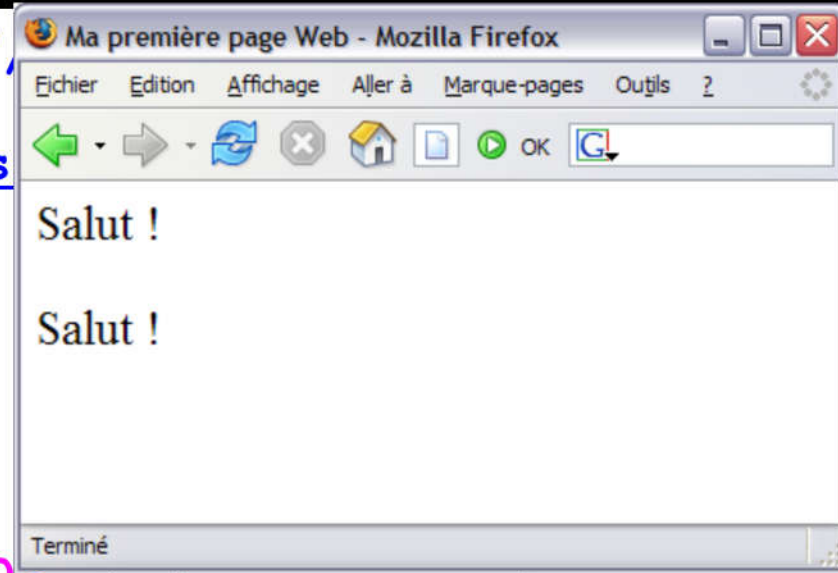

Héritage

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Ma première page Web</title>
  </head>
  <body>
    <p>Salut !</p>
  </body>
</html>
```

- `<html>` parent de `<body>` parent de `<p>`
- `<p>` hérite de `<body>` hérite de `<html>`
- L'enfant hérite des propriétés de ses parents

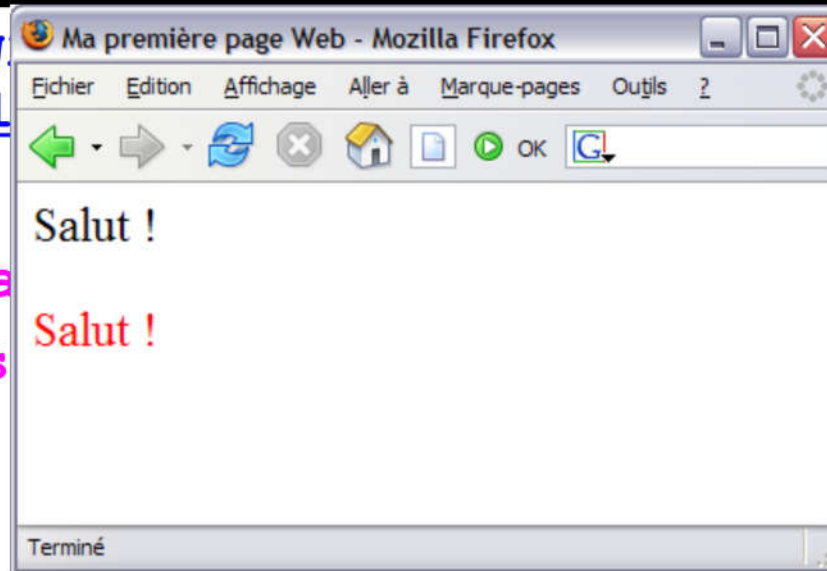
Héritage

```
<!DOCTYPE HTML PUBLIC "-//W3C//  
4.01//EN"  
"http://www.w3.org/TR/html4/s  
<html>  
  <head>  
    <title>Ma première page  
    <style type="text/css">  
      <!--  
        html { font-size : 150% ;  
      -->  
    </style>  
  </head>  
  <body>  
    Salut !<p>Salut !</p>  
  </body>  
</html>
```



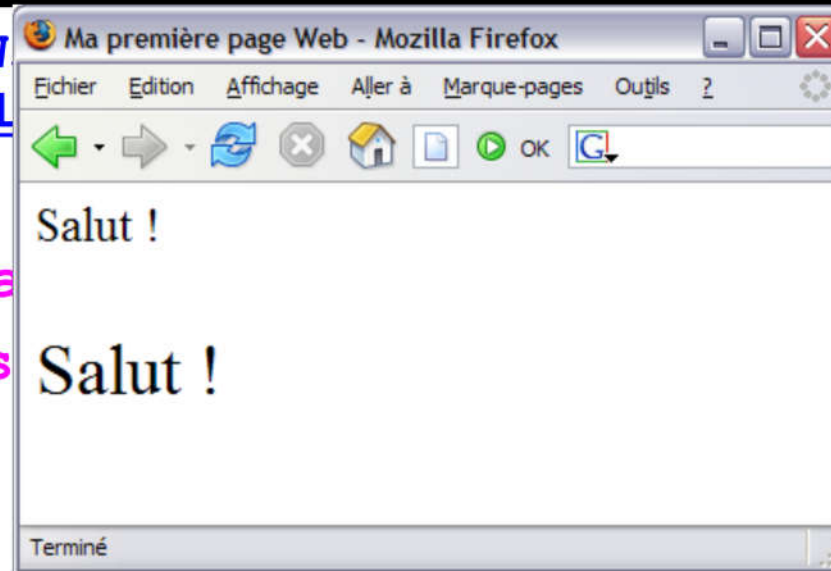
Héritage

```
<!DOCTYPE HTML PUBLIC "-//W
"http://www.w3.org/TR/html
<html>
  <head>
    <title>Ma première pa
    <style type="text/css
      <!--
        body { font-size :
        p    { color      : red ; }
      -->
    </style>
    </head>
    <body>
      Salut !<p>Salut !</p>
    </body>
  </html>
```



Héritage

```
<!DOCTYPE HTML PUBLIC "-//W
"http://www.w3.org/TR/html
<html>
  <head>
    <title>Ma première pa
    <style type="text/css
      <!--
        html { font-size :
        p    { font-size : 150% ; }
      -->
    </style>
    </head>
    <body>
      Salut !<p>Salut !</p>
    </body>
  </html>
```



Sélecteurs CSS

Motif	Signification
*	Tout élément.
E	Tout élément E (un élément de type E).
E F	Tout élément F descendant d'un élément E
E > F	Tout élément F enfant d'un élément E
E:first-child	Un élément E premier enfant de son parent
E:link E:visited	L'élément E , ancre source d'un hyperlien, lorsque la cible n'est pas visitée (:link) ou déjà visitée (:visited)
E:active E:hover E:focus	L'élément E durant certaines actions de l'utilisateur : est actif, est survolé, a le focus.
E:lang(L)	Un élément E s'il emploie la langue L

Sélecteurs CSS

Motif	Signification
E + F	Un élément F immédiatement précédé par un élément E. E et F ont le même parent.
E [foo]	Tout élément E avec l'attribut <i>foo</i> (peu importe la valeur).
E [foo="val"]	Tout élément E dont l'attribut <i>foo</i> à la valeur <i>val</i> .
E [foo~="val"]	Tout élément E dont l'attribut <i>foo</i> est une liste de valeurs séparées par des blancs et dont une à la valeur <i>val</i> .
E [lang ="en"]	Tout élément E dont l'attribut <i>lang</i> est une liste de valeurs séparées par des tirets, celle-ci commençant par <i>val</i> .
E.val	<i>Spécifique à HTML</i> . Identique à E[class~="val"]
E#mon_id	Tout élément E dont l'ID est <i>mon_id</i> .

Exemple de sélecteurs courants

p : toutes les balises p

b, u, i : les balises b, u et i

p b : les balises b dans une balise p

a:link:hover : les ancres sources (d'un lien)
non visitées lors de leur survol

p:first-letter : Première lettre des
paragraphes

Exemple de sélecteurs courants

div.ma_classe : les balises **div** dont l'attribut **class** contient au moins *ma_classe*

.ma_classe : les balises dont l'attribut **class** contient au moins *ma_classe*

div#mon_id : la balise **div** dont l'attribut **id** est *mon_id*

ul.menu li : Les items des listes à puces appartenant à la classe 'menu'

Valeurs des propriétés

- Taille de police :
 - **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**
- Métrique :
 - **in** : pouce (25,4 mm), **cm**, **mm**
 - **pt** : point (1/72^e de pouce)
 - **pc** : pica (12 points)
- Pixels : **px**
- Relatif à la taille de la police : **em**
- Pourcentage : %

Valeurs des propriétés

- URI : `url("mon_uri")`
- Couleurs :
 - `black, white, orange, yellow, green, red, ...`
 - `#RRGGBB` (00→FF), `#RGB` \equiv `#RRGGBB`
 - `rgb(red, green, blue)` (0→255)
 - `rgb(red%, green%, blue%)` (0→100%)
- Chaînes de caractères :
 - `"chaîne 'bonjour' "`
 - `"chaîne \"bonjour\""`
 - `'chaîne "bonjour" '`
 - `'chaîne \'bonjour\' '`

Exemples de propriétés

```
color      : blue ;
font-size  : 12pt ;
font-weight : bold ;
list-style-type : square ;
margin-top : 20px ;
padding-left : 3em ;
text-align : justify ;
text-decoration : none ;
visibility : hidden ;
```

Cascade de feuilles de style

■ Origine des feuilles de style :

- auteur
- utilisateur
- agent utilisateur

■ Règles de priorité :

- Tri par média
- Tri par origine
 - agent utilisateur
 - utilisateur
 - auteur
 - auteur (!important)
 - utilisateur (!important)



Ordre d'application
Écrasement des précédents

Cascade de feuilles de style

■ Règles de priorité (suite) :

■ Tri par priorité (spécificité)

- 4 chiffres calculés pour chaque sélecteur (A,B,C,D)
- A=1 si attribut style (`<p style="...">`)
- Si A==0, B=nombre d'**id** dans le sélecteur `#ident { ... }`
- Si A==0, C=nombre de **class** dans le sélecteur `.ma_classe { ... }`
- Si A==0, D=nombre de balises et pseudo-éléments dans le sélecteur `p b a:hover { ... }`

■ Tri par ordre d'apparition

- ordre des feuilles de style
- ordre des styles incorporés

Dépendance au média

■ Types :

- all : tous
- braille : système tactile
- embossed : imprimante braille
- handheld : système portable
- print : imprimante
- projection : système de projection
- screen : moniteur
- speech : synthétiseur vocal
- tty : terminal texte
- tv : télévision

Dépendance au média

- Spécifier un média :

```
@media media1 [, media2] {  
    /* feuille de style */  
}
```

```
@import url("url_css") [media1  
    [, media2]] ;
```

Modèle de boîtes

- Tous les éléments HTML/XHTML sont considérés comme des boîtes
- Les deux grands modes d'affichage des boîtes sont :
 - Le mode ligne (display : inline)
 - Le mode bloc (display : block)
- La taille finale des boîtes dépend de :
 - Taille
 - Remplissages
 - Bords
 - Marges

Point sur les boîtes

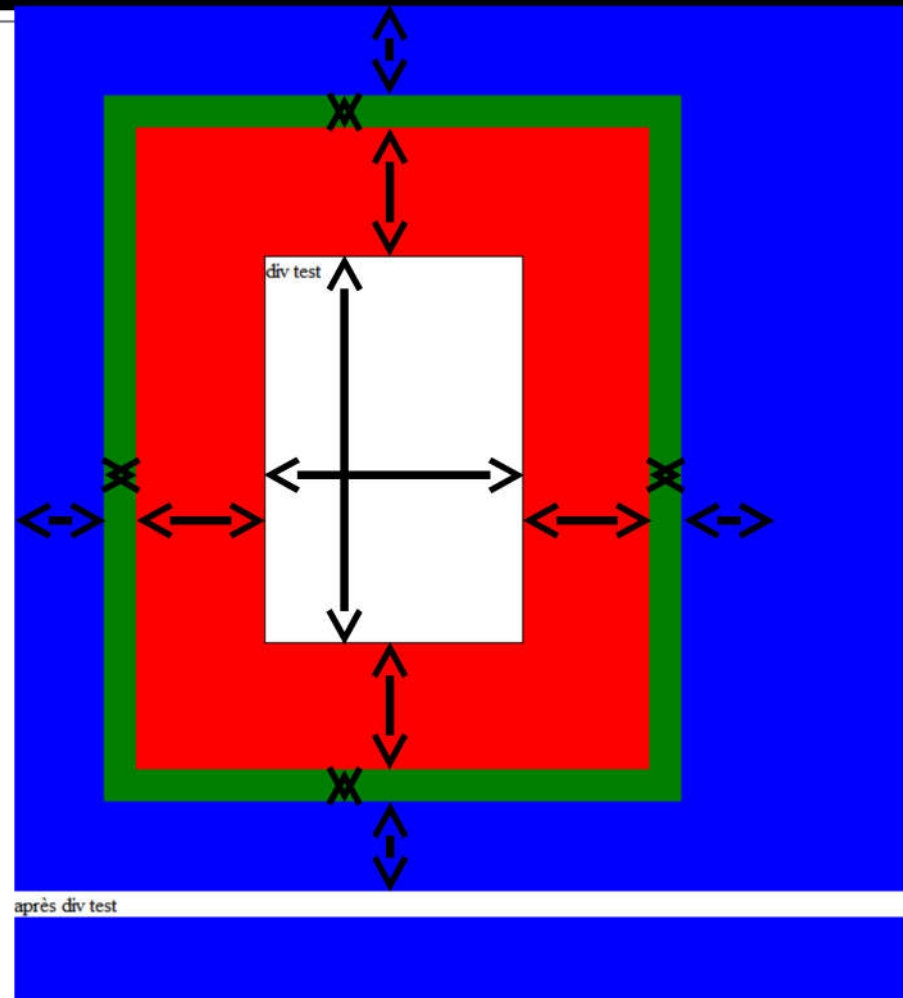
```
<html><head>
<style type='
body { background-color : blue ; }
#a { background-color : red ;
  border : solid 2px green ;
  width : 200px ;
  height : 300px ;
  padding : 100px ;
  margin : 70px ; }
#a div { width : 100px ;
  height : 100px ;
  border : 1px solid black ;
  background-color : white ; }
#b { background-color : white ; }
</style> </head>
<div id='a'>
</div>
<div id='b'>après div test</div>
</body> </html>
```

largeur, hauteur
= width, height

marge interne
= padding

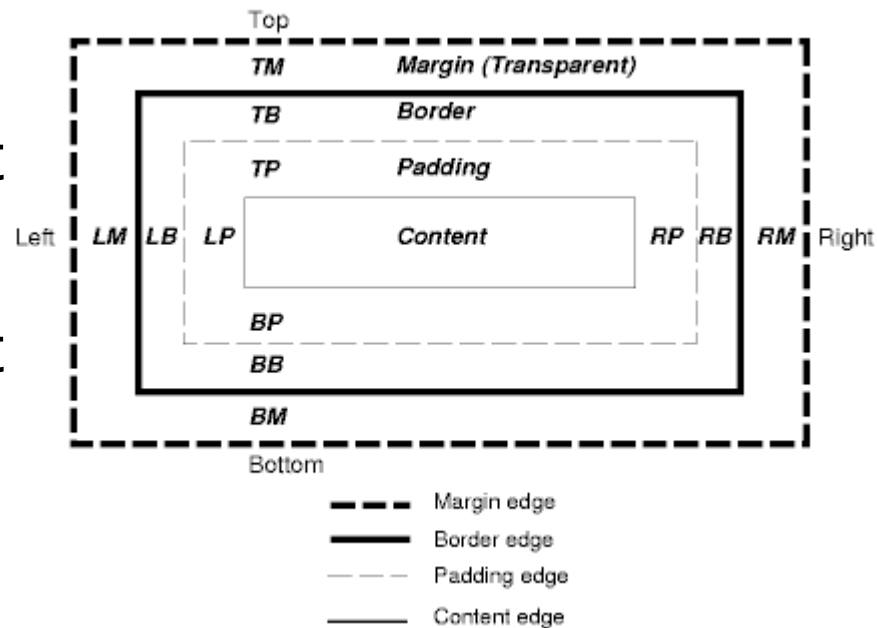
bord
= border

marge externe
= margin



Point sur les boîtes

- margin
 - top, bottom, left, right
- border
 - top, bottom, left, right
- padding
 - top, bottom, left, right



Source : www.w3.org

Ouvrons la boîte

Tout n'est que **boîte**. La compréhension des mécanismes de boîtes est la clé de la mise en forme des éléments structurés de la page Web

```
<!DOCTYPE h
<html> <hea
  <meta http
    <style
```

```
</styl
<title>Unc
</head> <b
<div c
<p>Tc
<em>l
e
c
</div>
</body>
</html>
```

Ouvrons la boîte

Tout n'est que **boîte**. *La compréhension des mécanismes de boîtes est la clé de la mise en forme des éléments structurés de la page Web*

```
<!DOCTYPE h  
<html> <hea  
<meta http  
<style  
body
```

```
</styl  
<title>Un  
</head> <br  
<div c  
<p>Tc  
<em>l  
€  
C  
</div>  
</body>  
</html>
```

Ouvrons la boîte

display : inline

```
<!DOCTYPE h
<html> <head
<meta http
<style
*
body
div
p
strong
em
span
</style>
<title>Une
</head> <bo
<div c
<p>To
<em>L
e
c
</div>
</body>
</html>
```

Tout n'est que **boîte**. La compréhension
des mécanismes de boîtes est la clé de la
mise en forme des éléments structurés de la
page Web

display : block

Ouvrons la boîte

display : inline

```
<!DOCTYPE h
<html> <head
<meta http
<style
*
body
div
p
strong
em
span
</style>
<title>Un
</head> <bo
<div c
<p>To
<em>L
e
c
</div>
</body>
</html>
```

Tout n'est que **boîte**.

La compréhension des mécanismes de boîtes

est la **clé** de la mise en forme des éléments

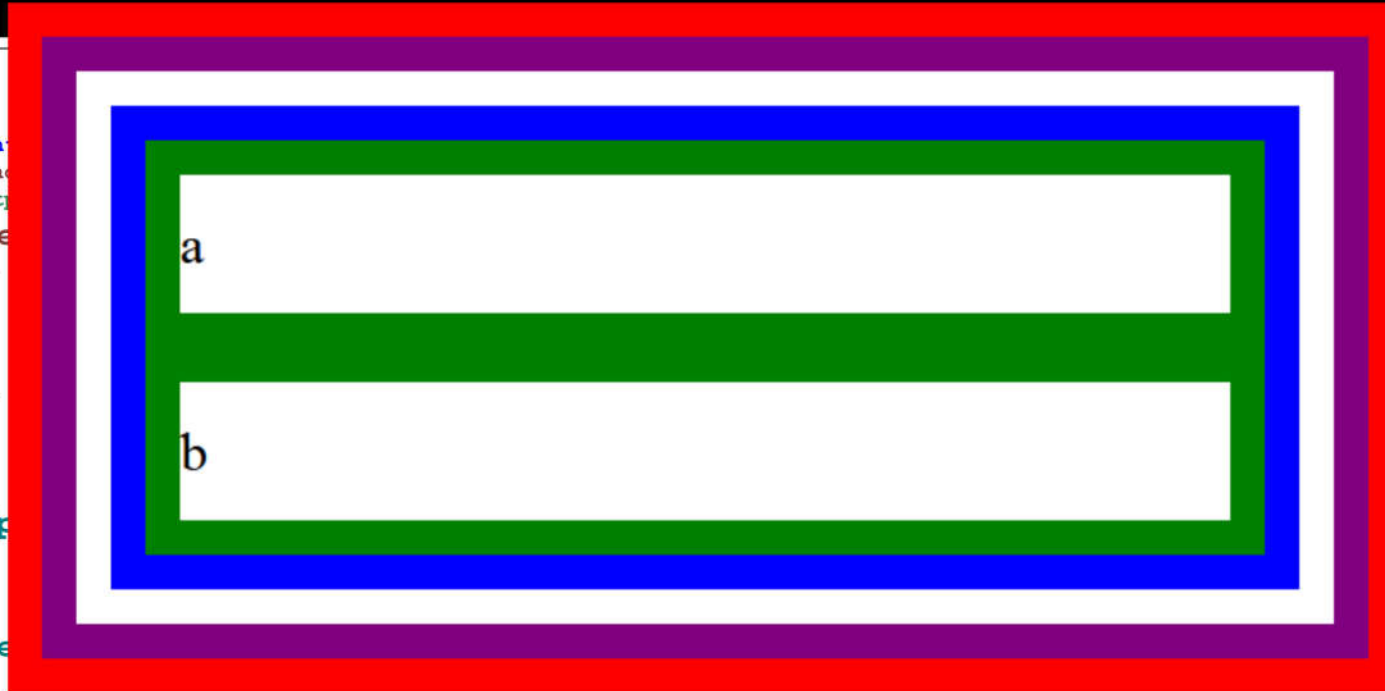
structurés de la page Web

display : block

Autour de et dans la boîte

```
<!DOCTYPE h
<html> <head
<meta http
<style
*
body
div#p
div.e
```

```
</styl
<title>Un
</head> <b
<div i
<c
<c
</div>
</body>
</html>
```



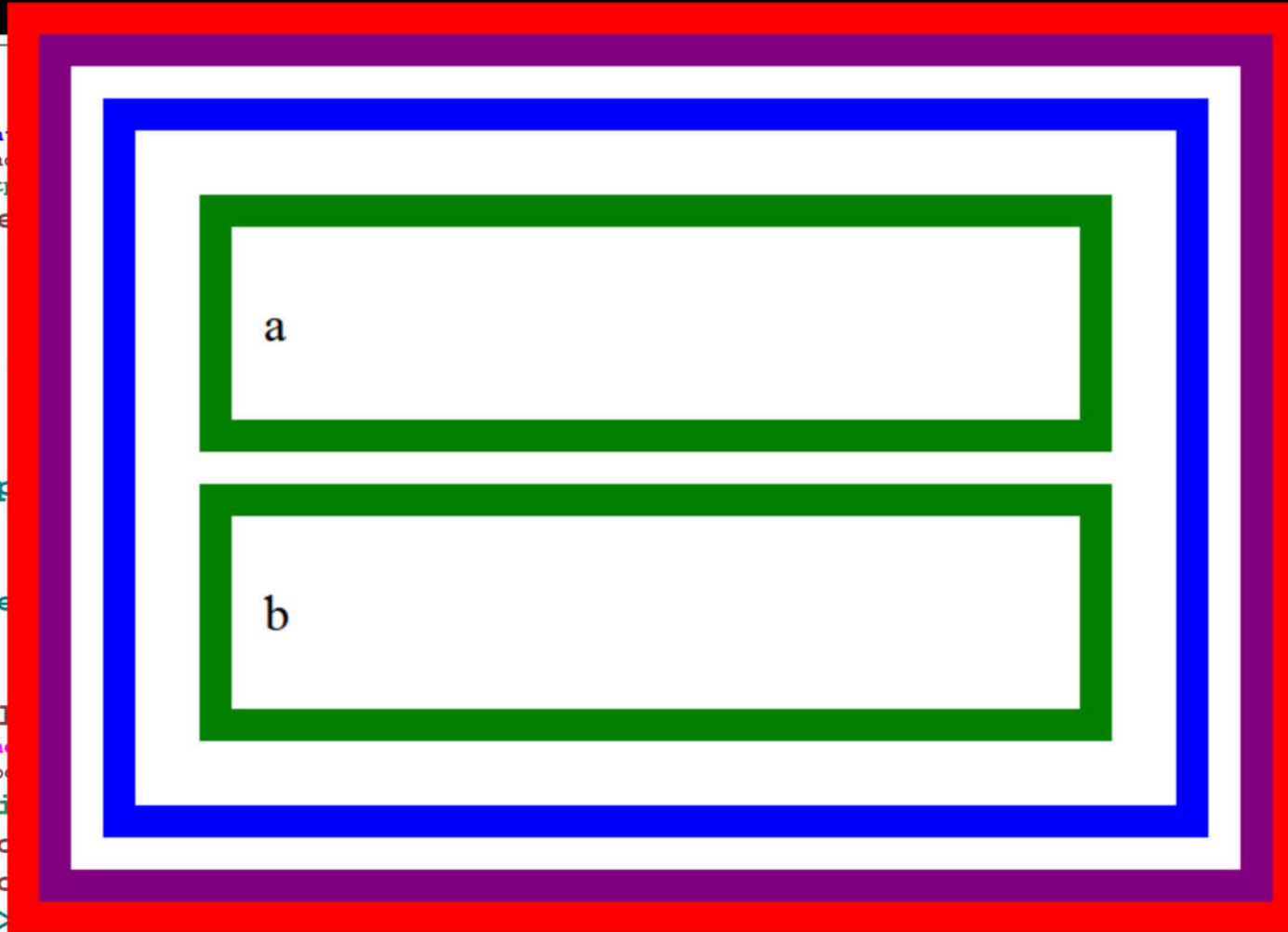
Autour de et dans la boîte

```
<!DOCTYPE h
<html> <head
<meta http
<style
*
body
div#p
div.e
</styl
<title>Un
</head> <b
<div i
<c
<c
</div>
</body>
</html>
```



Autour de et dans la boîte

```
<!DOCTYPE html>
<html> <head>
  <meta http-equiv="charset" content="UTF-8">
  <style>
    *
    body
    div#page
    div.container
  </style>
</head> <body>
  <div id="page">
    <div class="container">
      <div class="content">
        <div class="text">
          a
        </div>
        <div class="text">
          b
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```



Les boîtes qui flottent

```
<!DOCTYPE  
<html> <he  
<meta  
<styl  
div  
body  
  
#end  
#end  
#end  
</sty  
<title>C  
</head> <  
<div  
  <  
  <  
  <  
  
</div  
</body>  
</html>
```

aaaaaaaaaa

bbbbbbbbbb

ccccccccc

[dtd">](#)

:-8">

Les boîtes qui flottent

```
<!DOCTYPE  
<html> <he  
<meta  
<styl  
div  
body  
#enf  
#enf  
  
#enf  
</sty  
<title>U  
</head> <  
<div  
  <  
  <  
  <  
</div  
</body>  
</html>
```



`.d">`

`3">`

Les boîtes qui flottent

```
<!DOCTYPE  
<html> <he  
<meta  
<styl  
div  
body  
#enf  
#enf  
#enf  
  
</sty  
<title>U  
</head> <  
<div  
  <  
  <  
  <  
</div  
</body>  
</html>
```

aaaaaaaaaa

bbbbbbbbbb

cccccccccc

d">

3">

Les boîtes qui flottent

```
<!DOCTYPE  
<html> <he  
<meta  
<styl  
div  
body  
#enf  
#enf  
#enf  
  
</sty  
<title>U  
</head> <  
<div  
  <  
  <  
  <  
</div  
</body>  
</html>
```

aaaaaaaaaa

bbbbbbbbbb

cccccccccc

d">

3">

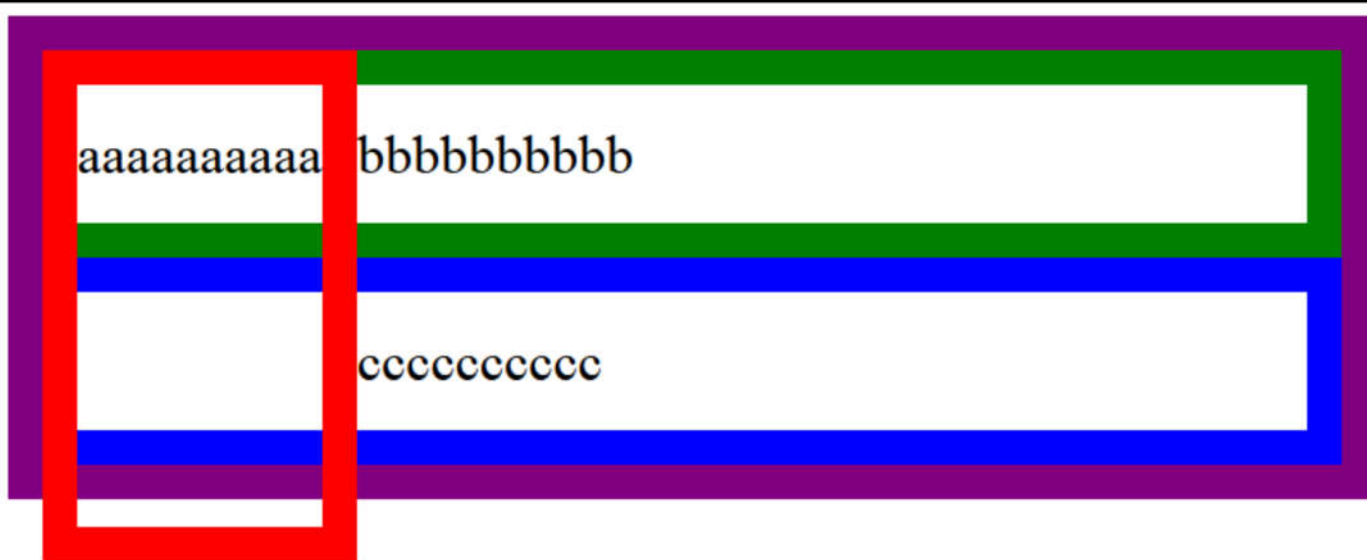
Les boîtes qui flottent et les autres

```
<!DOCTYPE h
<html> <hea
<meta htt
<style>
div
body
#enf
#enf
#enf
</sty
<title>Un
</head> <b
<div :
<
<
<
</div>
</body>
</html>
```



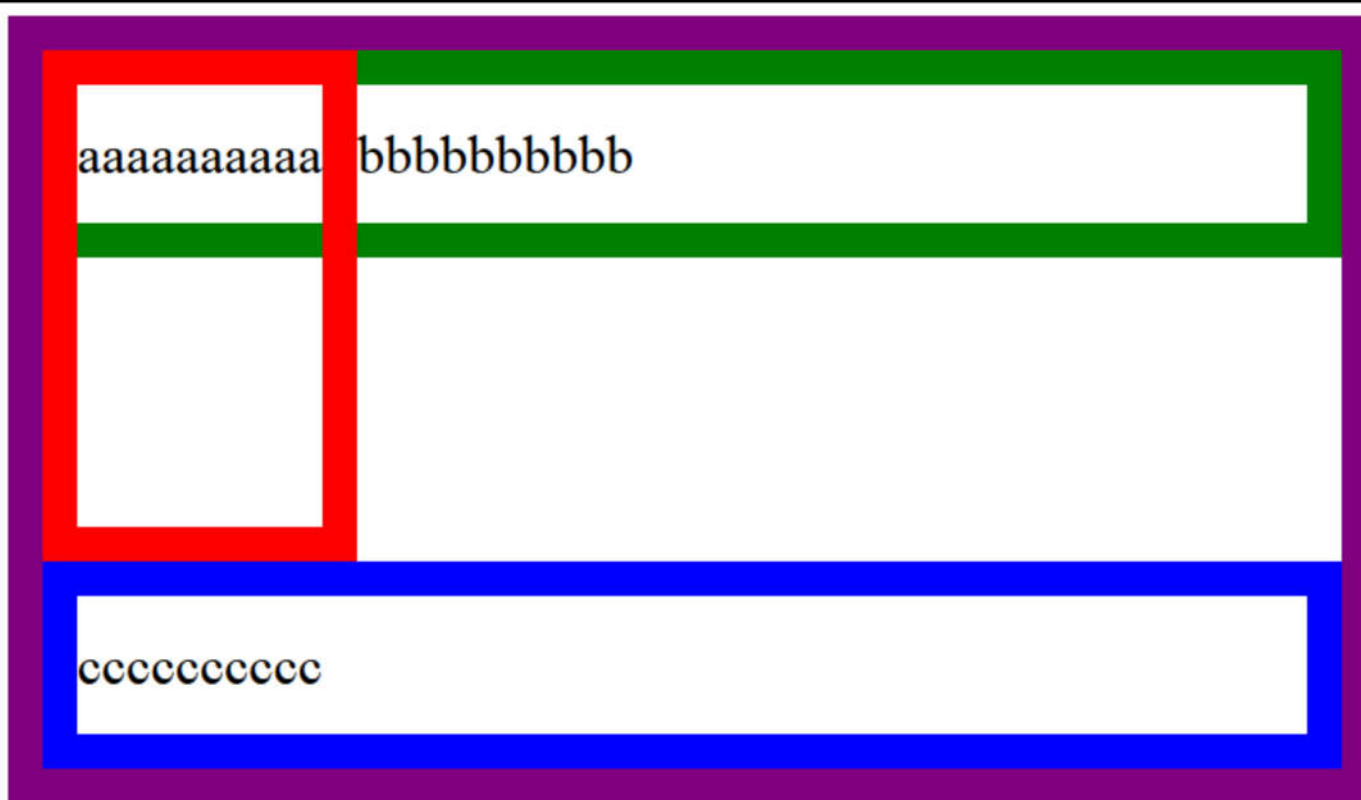
Les boîtes qui flottent et les autres

```
<!DOCTYPE h
<html> <hea
<meta htt
<style>
div
body
#enf
#enf
#enf
</sty
<title>Un
</head> <b
<div
<
<
<
</div>
</body>
</html>
```



Les boîtes qui flottent et les autres

```
<!DOCTYPE h
<html> <hea
<meta htt
<style>
div
body
#enf
#enf
#enf
</sty
<title>Un
</head> <b
<div :
<
<
<
</div>
</body>
</html>
```



Quirks Mode

- Mode de rendu des nouveau navigateurs lorsqu'ils tentent de rester compatibles avec les erreurs du passé
- Mode déclenché en fonction de la DTD
 - DTD complète → mode de rendu normal
 - DTD partielle ou absente → mode de rendu dégradé
- Importance de fixer correctement la DTD !
- Différences entre les modes :
 - Taille des boîtes
 - Alignements verticaux
 - Héritage des polices dans les tableaux

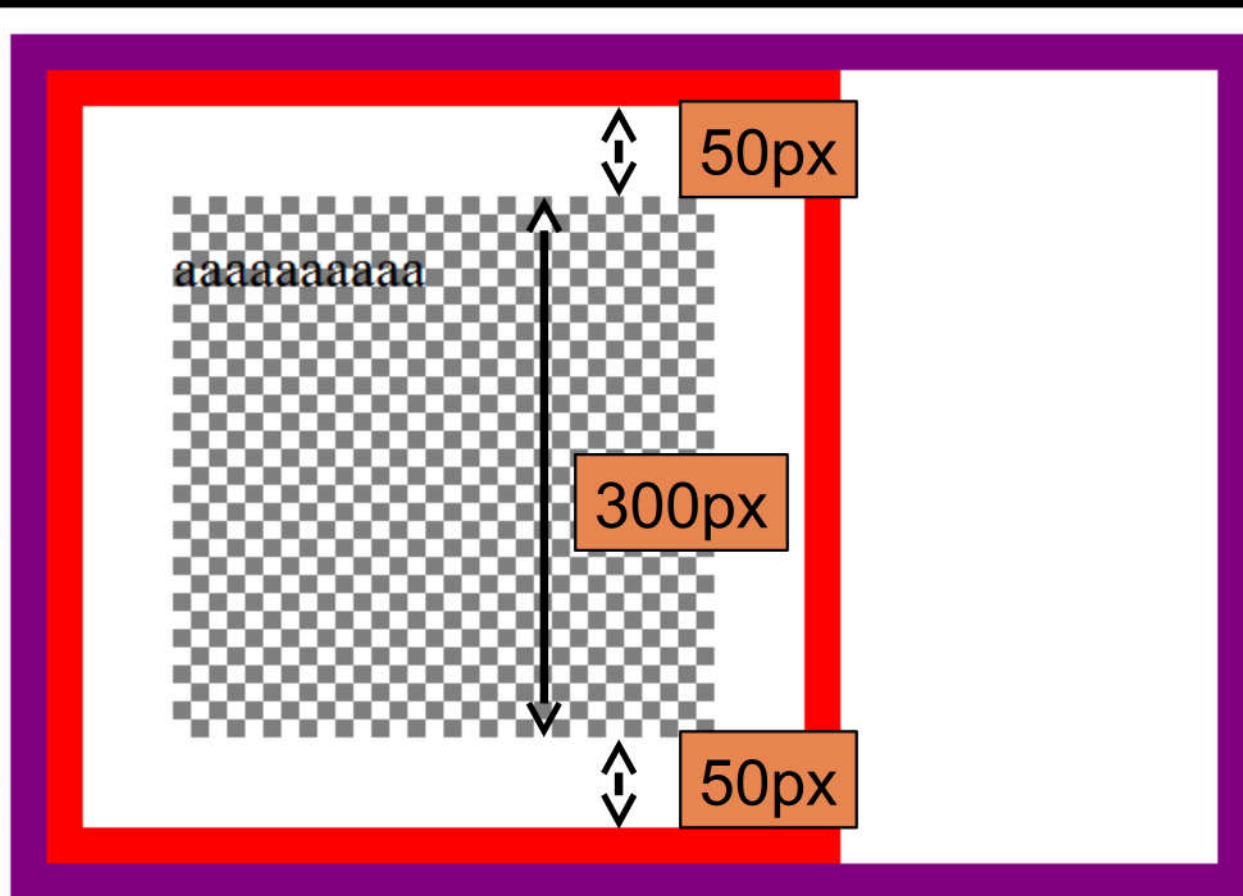
Quirks Mode : exemple normal

```
<!DOCTYPE html>
<html> <head>
  <meta ht
  <style t
    div

  body

  #enfant

</style>
<title>Une bo
</head>
<body>
  <div id=
    <div
  </div>
</body>
</html>
```



Quirks Mode : exemple dégradé

```
<html> <head>
  <meta http
  <style typ
  div

body

#enfant1
```

```
</style>
<title>Une boîte
</head>
<body>
  <div id='p
    <div c
  </div>
</body>
</html>
```

