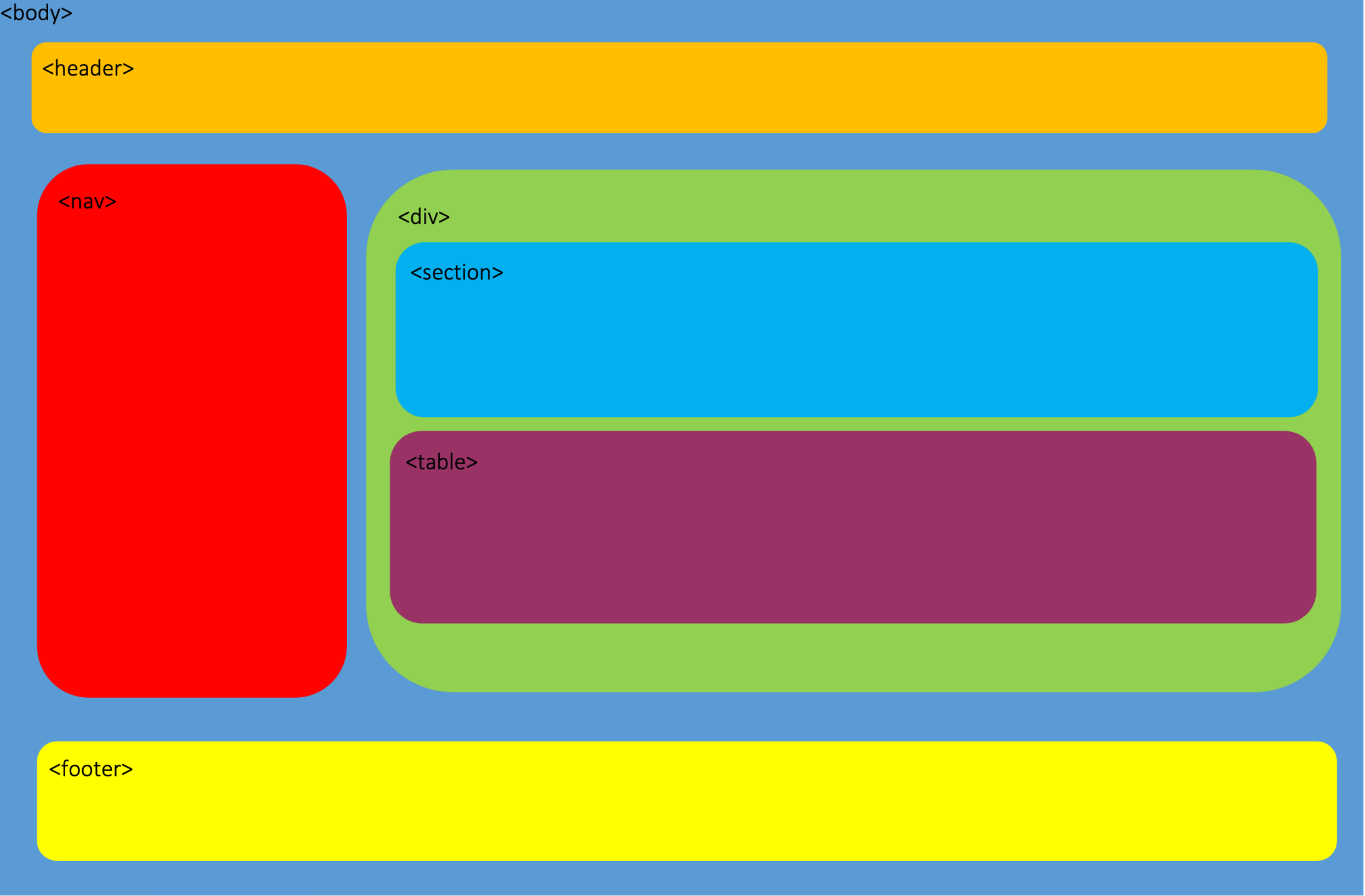


CSS 3

- Feuilles de style en cascade | Cascading Style Sheets

LE MODELE DES BOITES



Plan

- CSS

- 1. Généralités**

2. Règles et sélecteurs

3. Propriétés utiles

4. Placement des styles

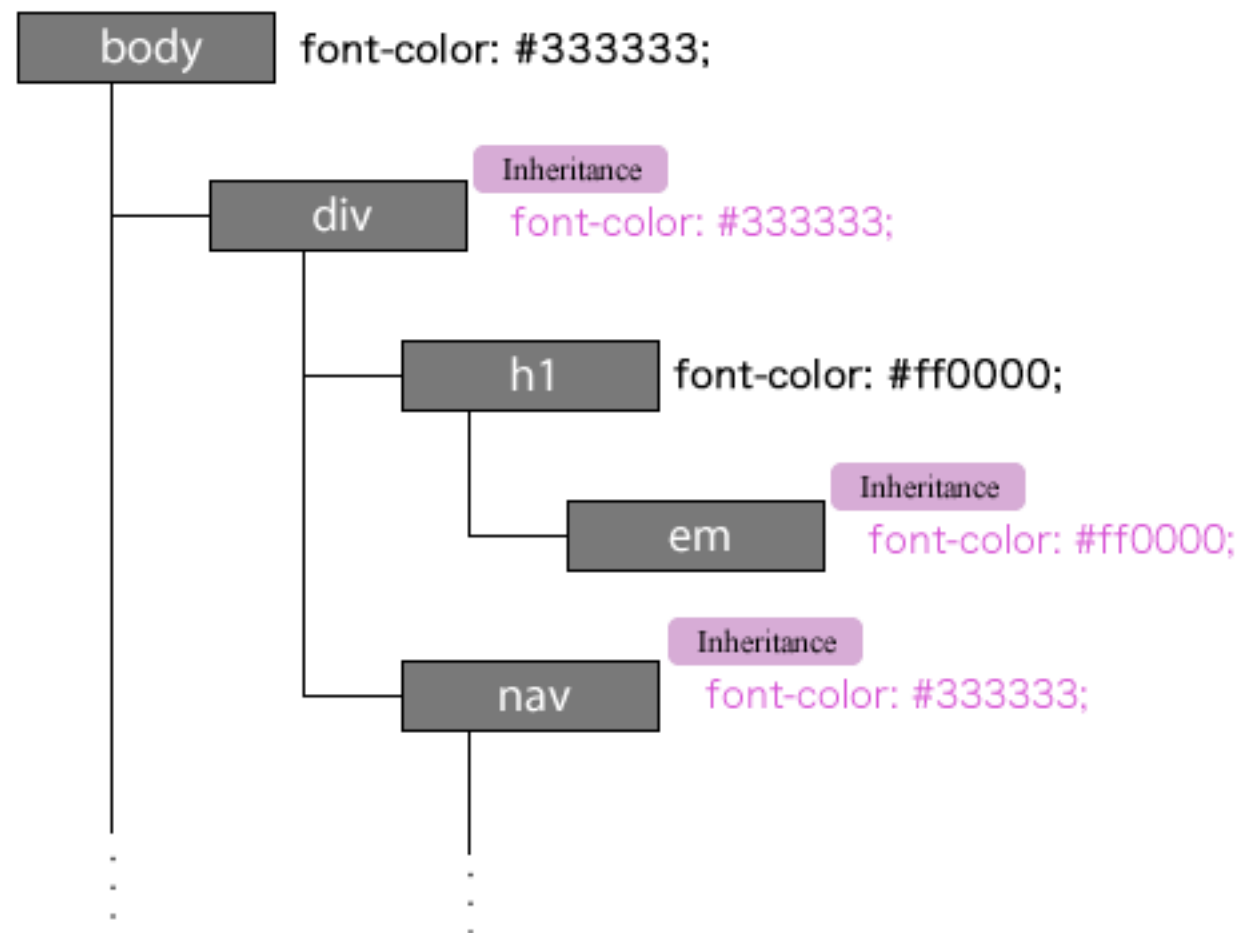
5. Héritage et cascade

1. CSS, généralités

- Objectif
 - Décrire comment un document HTML doit être affiché
 - Remplace les éléments d'affichage des anciennes versions de HTML (séparation réelle du contenu et de sa présentation)
 - L'affichage est pris en charge par le navigateur (normalement)
- Principe
 - Décoration de l'arbre des éléments HTML
 - Associer un certain nombre d'attributs de style à un élément

Example

<http://www.w3.org/wiki/CSS/Training/Inheritance>



Feuille de Style CSS

- Une feuille de style est composée d'un certain nombre de règles (rules)
- Une règle se compose
 - d'un sélecteur
 - d'une déclaration
- Une déclaration se compose d'un ensemble de propriétés/valeurs
- Remarque
 - L'ordre des règles est indifférent
 - Tous les styles ne peuvent pas s'appliquer à tous les éléments
 - Ex. : une image n'a pas de style de police

Plan

- CSS

1. Généralités

- 2. Règles et sélecteurs**

3. Propriétés utiles

4. Placement des styles

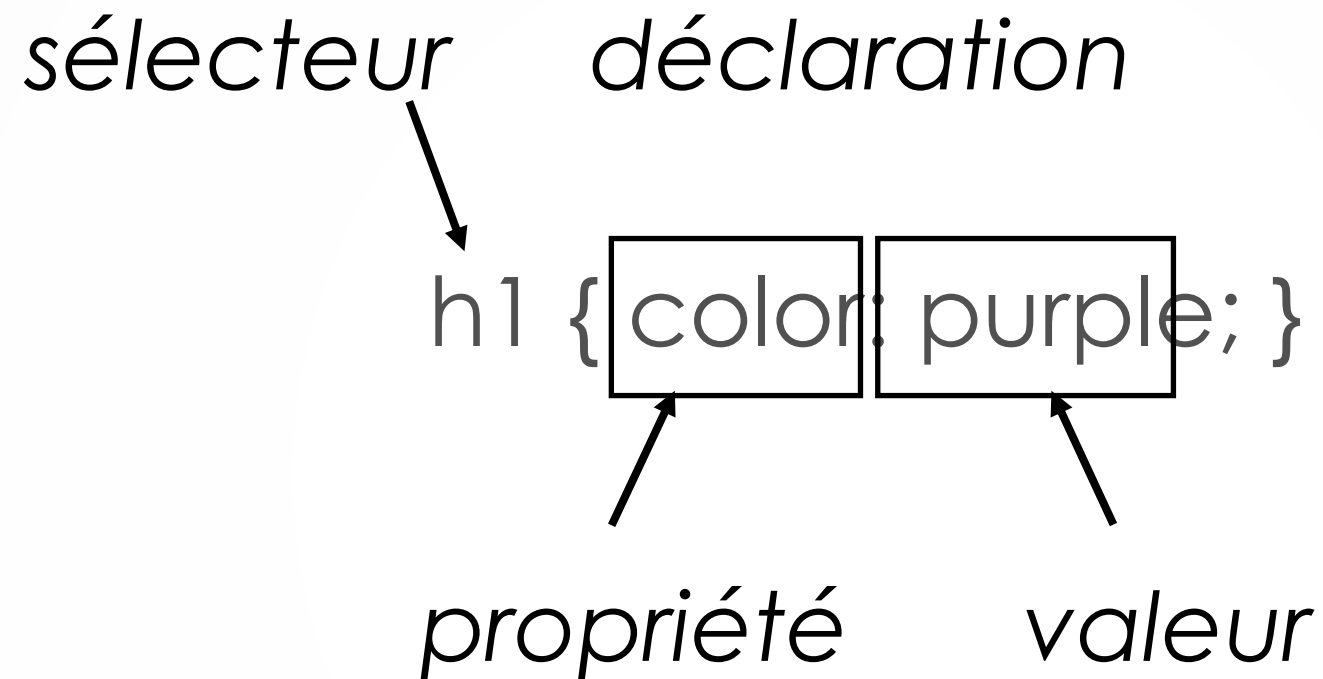
5. Héritage et cascade

1 - Structure des règles

sélecteur *déclaration*

h1 { color: purple; }

propriété *valeur*



h1 { font-family: Arial, sans-serif; font-style: italic }

séparateur de propriétés/valeurs



1 - Structure des règles

```
/* Auteur : BTS SIO */  
a {  
  color:#333;      /* gris sombre */  
}  
/* -----  
Typographie et couleurs  
----- */  
p {  
  font-size:12px;  
}  
#menu li {  
  color:#ff0000;  
}  
/* -----  
Structure du site  
----- */  
div {...  
}
```

Commentaire sur une ligne

Commentaire sur plusieurs lignes

Exemple

```
body {  
    background: #FFFFFF;  
    color: black; /* commentaire */  
    margin-left: 5%;  
    margin-right: 5%;  
    font-family: Tahoma, Georgia, Arial;  
}
```

Plan

- CSS

1. Généralités

- 2. Règles et sélecteurs**

3. Propriétés utiles

4. Placement des styles

5. Héritage et cascade

Sélecteurs CSS : simples et groupes

- Simple
 - Lié à un type d'élément HTML
 - Utilisation de son nom
 - Exemple : `h1 { text-align: center; }`
- Groupe
 - Regroupement de règles qui s'appliquent à plusieurs éléments
 - Exemple : `h2, p { font-family: Georgis, Arial; }`
- Impossibilité de considérer différemment des éléments de même type

Sélecteurs CSS : classes

On peut assigner une classe à un élément HTML

```
<h1 class="centrer" >
```

Celle-ci spécifie un sélecteur particulier dans le feuille de style

```
h1.centrer { text-align: center; }
```

Une classe peut s'appliquer à de multiples éléments

```
.centrer { text-align: center; }
```

s'appliquera aussi à <h2 class="centrer">, etc.

Sélecteur CSS : pseudo-classes

- Sélecteurs qui sélectionnent des éléments en fonction de leur état à un moment donné
- Exemple et intérêt principal
 - **a:link** — lien non visité et inactif
 - **a:hover** — lien sur lequel passe le pointeur de la souris
 - **a:active** — lien sur lequel on clique
 - **a:visited** — lien déjà visité
- Exemple

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

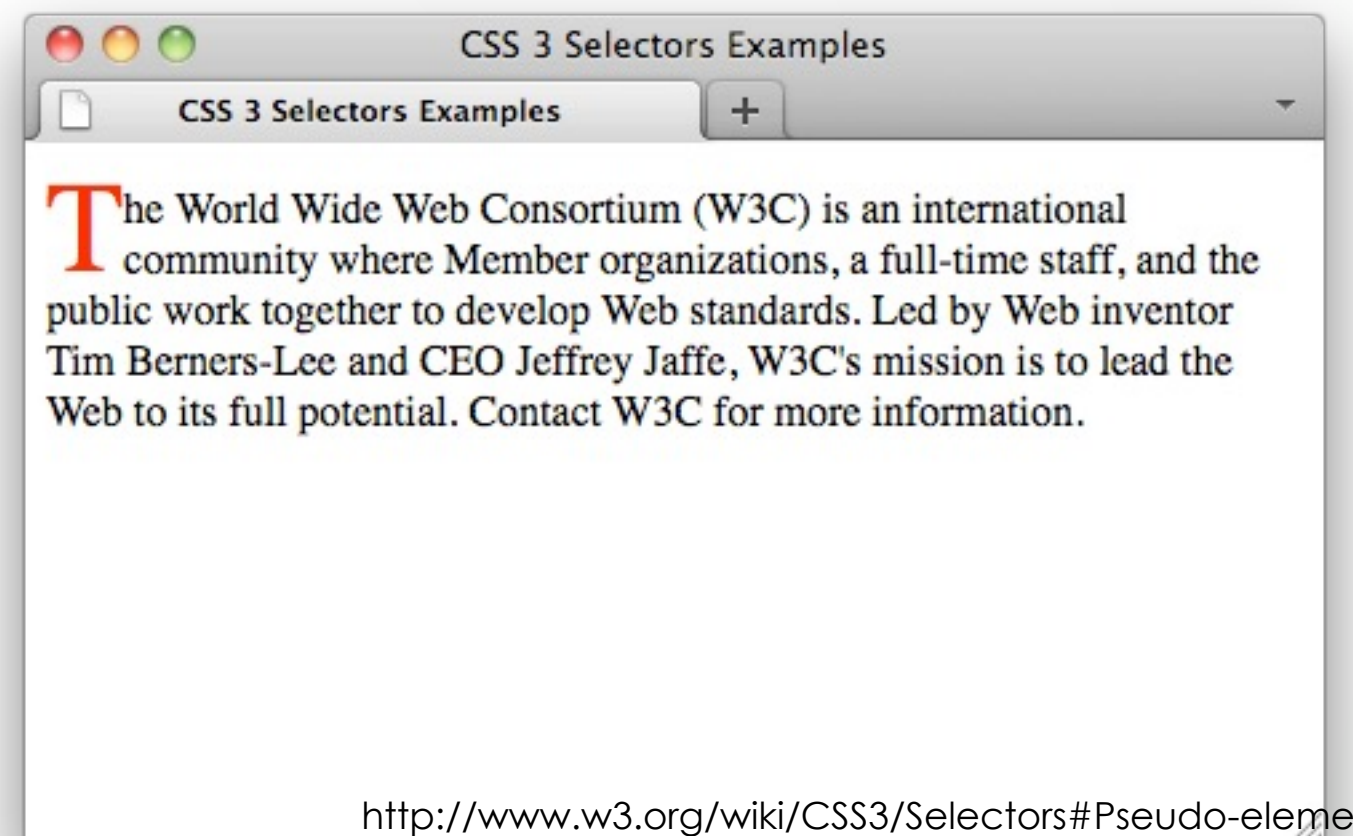
```
a:visited {color: magenta;}
```

```
a:hover {  
    color: red;  
    text-decoration:none;  
    font-weight: bold;}
```

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

Sélecteurs CSS : pseudo-éléments

- **::first-letter**
 - première lettre dans un élément bloc (ex. p, h1, ...)
- **::first-line**
 - première ligne dans un élément bloc (ex. p, h1, ...)
- **::after**
- **::before**
 - avant et après du contenu



```
p::first-letter {  
  font-size: 200%;  
  color: red;  
}
```

Sélecteurs CSS : contextuels

- Sélecteurs qui ne sélectionnent que des éléments dans un certain contexte
 - Les plus complexes mais aussi les plus riches.
- Exemple de style contextuel :

CSS :

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

HTML :

```
<h1>Ceci est un texte de  
header <em>ce texte est mis en  
évidence</em> celui-ci ne l'est  
pas.</h1>
```

```
<p>Dans ce paragraphe,  
<em>ceci est mis en  
évidence</em>  
</p>
```

Ceci est un texte de header *ce texte est mis en évidence* celui-ci ne l'est pas.

Dans ce paragraphe, *ceci est mis en évidence*

Sélecteurs CSS : id

Le sélecteur id a presque la même fonction que class, à la différence qu'on ne peut l'utiliser **qu'une seule fois dans la page**, contrairement au sélecteur class.

C'est pour cela qu'il est plutôt utilisé à la mise en page qu'à la mise en forme de caractères.

```
#menu { background-color:silver; width:100px; float:left; }
```

```
#contenu { margin-left:110px; }
```

```
<div id="menu">  
  <ul>  
    <li>Menu 1</li>  
    <li>Menu 2</li>  
    <li>Menu 3</li>  
    <li>Menu 4</li>  
    <li>Menu 5</li>  
  </ul>  
</div>  
<div id="contenu">  
... 
```

Plan

- CSS

1. Généralités
2. Règles et sélecteurs
- 3. Propriétés utiles**
4. Placement des styles
5. Héritage et cascade

Propriétés de texte

- font-size: small | medium... | % | x pt
- font-family: fontname1, fontname2 (*si la première n'est pas disponible*), familyname (serif, sans-serif, etc.)
- font-weight: bold | lighter
- font-style: italic, oblique
- text-align: left | center | right | justify
- text-indent: (*retrait de première ligne*) % | 2 px

Les couleurs

- color:
 - red | blue... | hexcode
- background-color:
 - red | blue... | hexcode

Hex Code	Color	Hex Code	Color	Hex Code	Color
#FFFFFF		#CCFFFF		#99FFFF	
#FFFFCC		#CCFFCC		#99FFCC	
#FFFF99		#CCFF99		#99FF99	
#FFFF66		#CCFF66		#99FF66	
#FFCCFF		#CCCCFF		#99CCFF	
#FFCCCC		#CCCCCC		#99CCCC	
#FFCC99		#CCCC99		#99CC99	
#FFCC66		#CCCC66		#99CC66	
#FF99FF		#CC99FF		#9999FF	
#FF99CC		#CC99CC		#9999CC	

Hex Code	Color	Hex Code	Color	Hex Code	Color
#FF9999		#CC9999		#999999	
#FF9966		#CC9966		#999966	
#FF66FF		#CC66FF		#9966FF	
#FF66CC		#CC66CC		#9966CC	
#FF6699		#CC6699		#996699	
#FF6666		#CC6666		#996666	
#FF6633		#CC6633		#996633	
#FF6600		#CC6600		#996600	
#FF33FF		#CC33FF		#9933FF	
#FF33CC		#CC33CC		#9933CC	

Types d'éléments

- Éléments blocs
 - Prennent la forme d'un bloc dans la page
 - ensemble de lignes
 - ne peuvent être contenus que dans d'autres éléments blocs
 - Exemple : p, img, ul, table, h1, div...
- Éléments inline
 - S'inscrivent dans la continuité des éléments
 - ne forcent pas un changement de ligne
 - peuvent être inclus dans n'importe quel élément
 - Exemple : a, em, span...
- Éléments de listes
 - éléments HTML qui ont un marqueur (ex: puce) et un ordre

Rappel sur div et span

- div
 - élément contenant d'autres éléments, servant à définir un bloc
 - servira essentiellement à faire du positionnement de bloc
- span
 - élément contenant d'autres éléments, inline
 - servira essentiellement à regrouper des caractéristiques globales pour des éléments textuels

Propriétés de positionnement

- S'appliquent aux éléments blocs
- Attribut position
 - static
 - bloc placé en fonction de sa position dans l'ordre des balises,
 - ne peut pas être modifiée dynamiquement
 - absolute
 - bloc placé précisément par rapport aux bords de la fenêtre d'affichage
 - la position peut être modifiée dynamiquement (déplacement)
 - relative
 - bloc positionné par rapport à sa position normale, dans le flux (ex. décalage de 20 pts à droite)

Propriétés graphiques : arrière plan

- Arrière-plan
 - background-color (couleur de fond)
 - background-image (image de fond)
 - background-repeat (types de répétition de l'image)
 - background-attachment (arrière-plan fixe quand on utilise l'ascenseur)
 - background-position (position par rapport au coin supérieur gauche)
 - background (attribut de résumé)
- Exemple : `p { background: image.gif fixed repeat-y }`

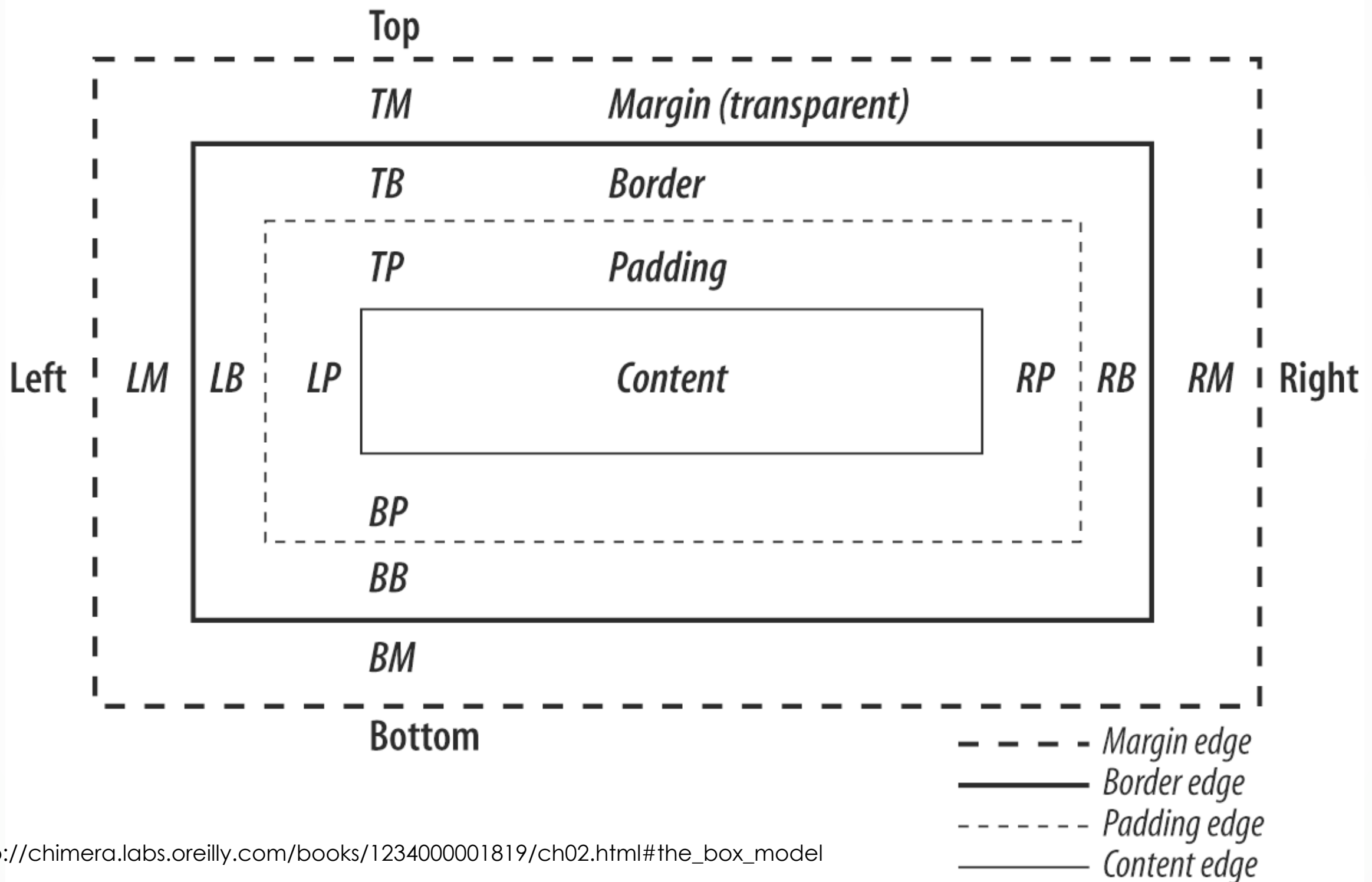
Propriétés graphiques : visibilité

- Attribut visibility et display
 - indiquent si (visible | hidden) et comment un élément est affiché
- Attribut z-index
 - en cas de superposition de blocs d'affichage, indique l'ordre dans lesquels le navigateur doit les afficher.

Remarques

- les distances s'expriment en pixels (px), points (pt), unités métriques (cm, mm), ou pourcentages de la taille de la fenêtre (%) ou en unité relative (em, rem)

Le modèle de boîte (box model)



Plan

- CSS
 - Généralités
 - Règles et sélecteurs
 - Propriétés utiles
 - **Placement des styles**
 - Héritage et cascade

Styles intégrés

- Déclarés comme attribut style d'un élément
- Exemple

```
<h1 style="font-family: Arial; font-style: italic;">
```

Un texte qui se retrouvera en arial italique

```
</h1>
```

Styles intégrés

- La feuille de style est déclarée dans l'en-tête (head) du document HTML
- Elle s'appliquera aux éléments dans body
- Exemple

```
<html>
<head>
  <title>Texte du titre</title>
  <style type="text/css">
    .important {color: red;}
    h1 {font-family: Arial; font-style: italic; }
    em { color: green }
  </style>
</head>
<body>... </body>
</html>
```

Feuilles de styles liées

Les règles se trouvent dans un fichier extérieur

On indique au navigateur où se trouve cette ressource de style, en utilisant un élément link dans l'en-tête <head>

```
<link rel="stylesheet" type="text/css" href="fichier-de-style.css" >
```

Intérêt

Plusieurs documents HTML peuvent faire appel à la même feuille de style

Plan

- CSS
 - Généralités
 - Règles et sélecteurs
 - Propriétés utiles
 - Placement des styles
 - **Héritage et cascade**

Héritage de styles

- Par défaut

- Les styles d'un élément sont héritées par ses éléments descendants
- Exemple :
`<p style="color: red;"> Du texte mis en évidence, pas mis en évidence.</p>`

Du texte mis en évidence, pas mis en évidence.

- A condition que l'héritage ait un sens

- i.e. que les caractéristiques soient applicables à l'élément enfant
 - un positionnement de bloc n'a pas d'intérêt pour un élément em qui y est contenu
- si un style est défini spécialement pour un élément
- ex: `em { color: blue; }`, l'héritage ne se fait pas

Du texte mis en évidence, pas mis en évidence.

Cascading style sheets : cascade

- On peut avoir concurrence entre plusieurs styles définis dans de multiples endroits
 - styles par défaut (1- navigateur)
 - fichiers CSS externes (2- spécifications globales au site)
 - élément head du document HTML (3- spécification globales au doc.)
 - attributs style des éléments (4- spécification locales)
 - style utilisateur (5- spécification de l'utilisateur)

Cascading style sheets : priorités

- Notion de cascade ou ordre de priorité des styles
 - trouver toutes les déclarations qui s'appliquent à un élément
 - les classer par spécificité
 - système de poids qui s'ajoutent
 - les classer par ordre d'apparence
 - plus une déclaration apparaît tard, plus elle a de poids
- Exemple
 - style (4) > style (4 hérité) > style (3) > style (2) > style (1)

l'instruction **!important** permet de donner la priorité à une style sur tous les autres malgré la position.