

CSS: l'entrée en matière

Bases de CSS

Ressources: spécifications CSS, c.f.
<http://www.w3.org/Style/CSS/current-work>

Histoire de CSS (1/2)

- *c.f. article wikipedia (en français ou en anglais)*
- Créé vers 1995
 - *Cascading HTML Style Sheets* de Håkon Wium Lie (créateur du test Acid2, impliqué dans Opera)
 - « Stream-based Style Sheet Proposal » (SSP) de Bert Bos (actuellement au W3C)
- Définir le rendu par les auteurs, et non plus par les navigateurs
- Plus général que l'HTML

Histoire de CSS (2/2)

- Chaque version de CSS inclut les précédentes
- Fin 1996 : version finale de CSS1
 - 8 sélecteurs et une cinquantaine de propriétés
- 2007 : Version « standard » de CSS2.1
 - Début de CSS2 en 1998
 - Ajoute 11 sélecteurs et 70 propriétés
- CSS3 débuté en 1999, « CSS4 » a débuté en 2010. Il n'existe pas vraiment, c'est une version itérative de modules
- Actuellement des « snapshot » (« en gros », une version par an)
- Support de CSS par les navigateurs : de mieux en mieux (cela n'a pas toujours été le cas)

Une évolution constante

- Du fait de spécifications parallèles
 - Parfois des préfixes (-moz- ou -webkit- etc.)
 - => écrire plusieurs fois le css (pour chaque préfixe...)
 - Qui disparaissent aux fils des versions des navigateurs
- Il y a des spécifications non implémentées
 - voire l'inverse
- État des lieux de la spécification :
<http://www.w3.org/Style/CSS/current-work>
 - avec un état des lieux des implémentations
 - <http://caniuse.com/>
 - Sur les pages des navigateurs :
<https://developer.mozilla.org/fr/docs/Web/CSS>
etc.

Base de css

Un peu de vocabulaire

Une page sans css

Une page web, c'est :

- un code source html
= un ensemble de tag et de texte
- vu comme un arbre
- Rendu lisible par le navigateur web

```
<h1>Le <strong>dormeur du val</strong></h1>
<p>C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.</p>

<p>Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.</p>

<p>Les pieds dans les glaieuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.</p>

<p><strong>Arthur RIMBAUD</strong><em>(1854-1891)</em></p>
```

Le dormeur du val

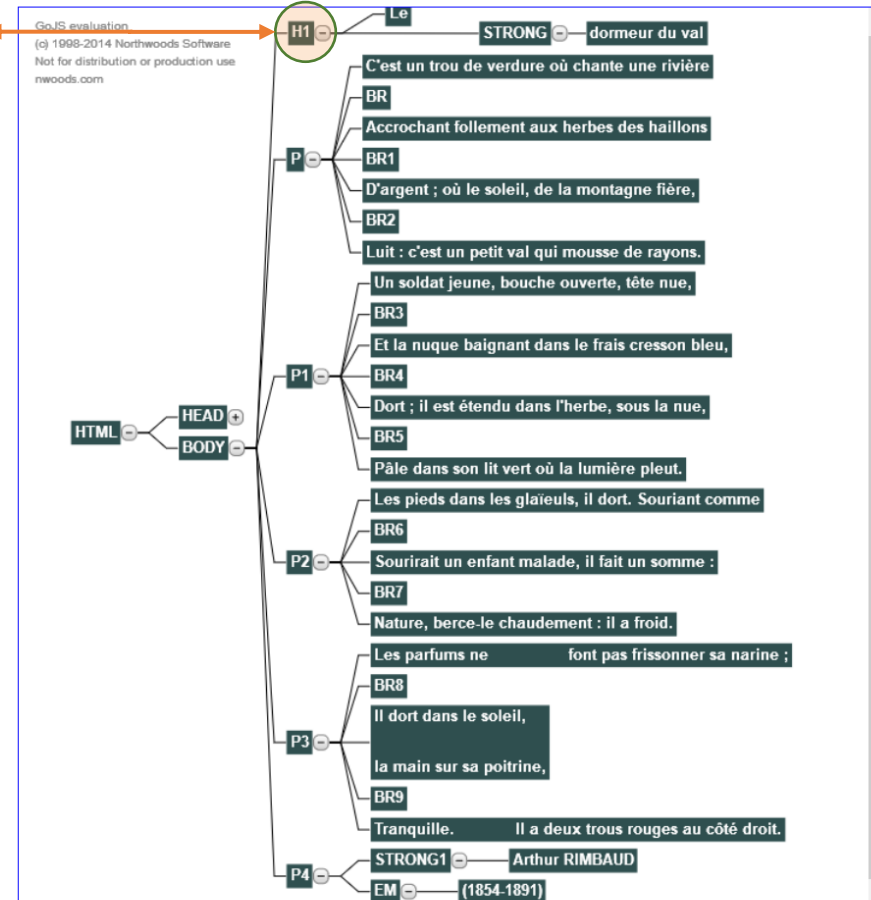
C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaieuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

Arthur RIMBAUD (1854-1891)



Comment fonctionne le CSS ?

Ce que doit exprimer le CSS

- Le CSS doit pouvoir dire comment chaque éléments (de la page) apparait
- Exemple : « le titre est écrit en rouge »
 - Le titre = quel(s) élément(s) est(sont) concerné(s) = le **sélecteur**
 - Écrit = la caractéristique (visuelle ou ...) = la **propriété**
 - En rouge = la valeur que doit prendre la propriété = la **valeur**

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

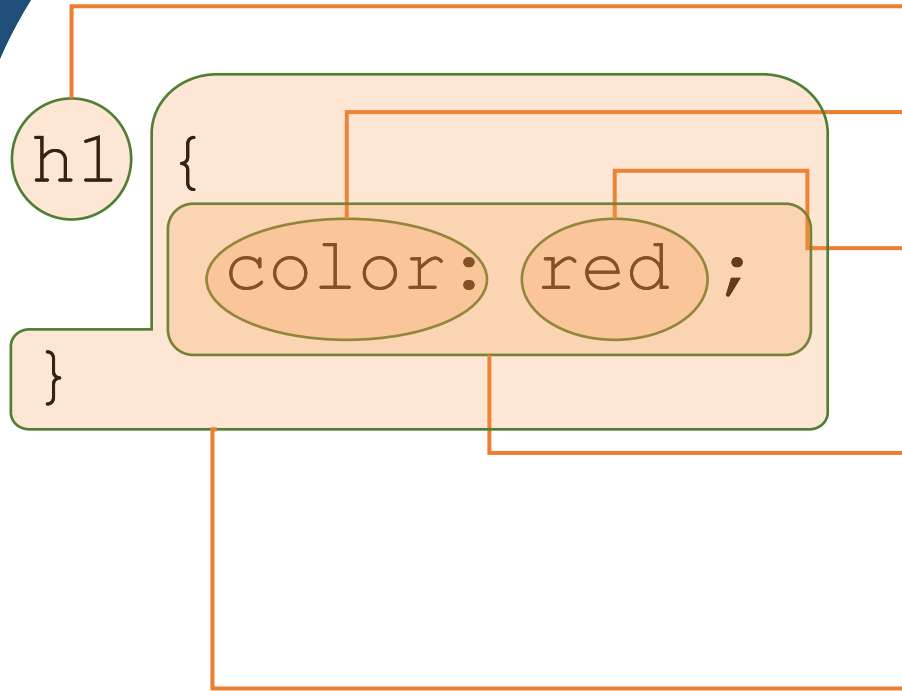
Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

Arthur RIMBAUD (1854-1891)

Comment écrire du CSS



- Le sélecteur (quel(s) élément(s) impacté(s))
- La propriété (modifiée) des éléments désignés
- La nouvelle valeur de la propriété (modifiée) des éléments désignés
- Une déclaration = « propriété : valeur ; » ,
« : » entre la propriété et la valeur
« ; » est un séparateur
- Un bloc de déclaration = ensemble de déclarations (une ou plusieurs) entre { et }

Vocabulaire, en résumé

- Un élément (***element***) = un tag html (<p>, <article>, , etc.)
- Un sélecteur (***selector***) = Quel(s) élément(s) sur la page seront impactés
- Une propriété (***property***) = Quelles caractéristiques des éléments désignés seront modifiées
- Des valeurs (***values***) = Comment les caractéristiques choisies des éléments désignés seront impactées
- Une déclaration (***declaration***) est l'association : « property: value(s) ; »
 - Le « ; » sépare les déclarations (à mettre à la fin de chaque déclaration)
- Un bloc de déclaration (***declaration bloc***) est un regroupement de déclaration entre { }
- Un bloc de déclaration doit être précédé par un (ou plusieurs) sélecteur
- La totalité du CSS que vous écrivez est appelée règles CSS (***rules or rules set***)

Doit-on tout spécifier en CSS ?

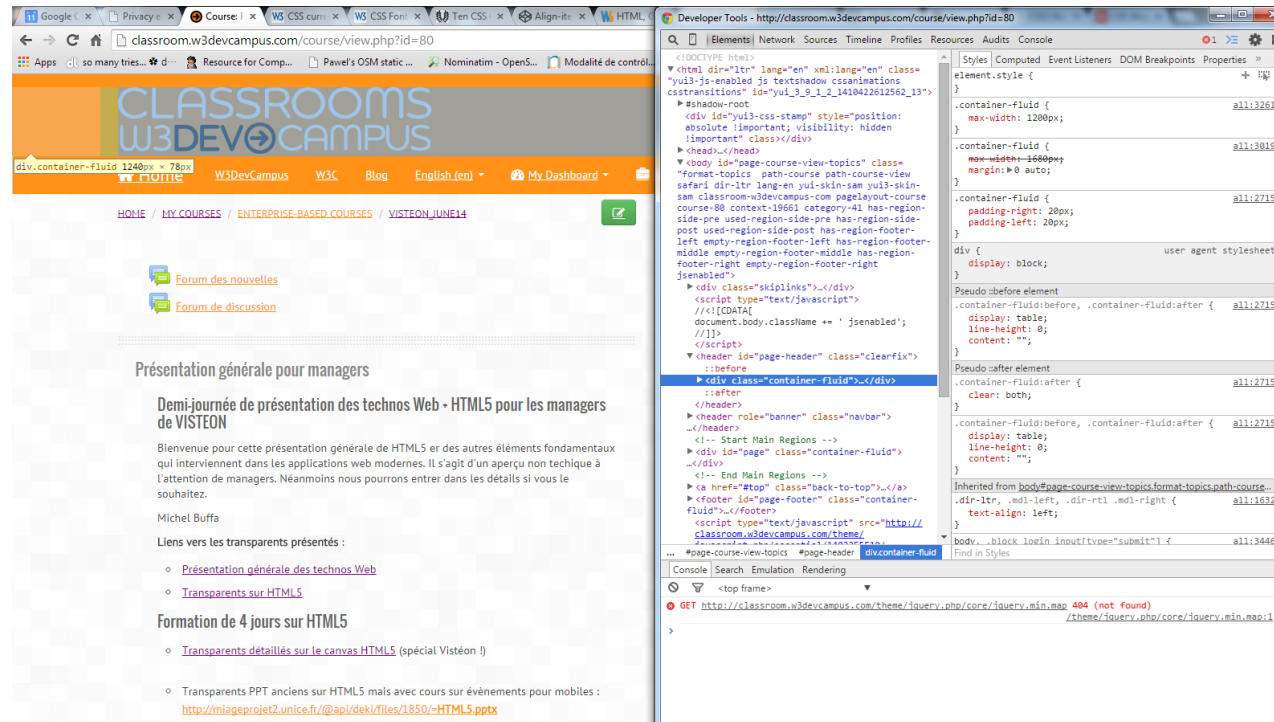
- Dans l'exemple précédent, il n'a pas été préciser (spécifier) quelle était la police de caractère des paragraphes, ni leur couleur d'écriture, etc.
- Chaque propriété à une valeur par défaut (propre à chaque navigateur)
- Si l'auteur de la page ne spécifie pas (directement ou non), alors la valeur par défaut est utilisée pour rendre la page lisible (le rendu)
- La réponse est donc : non (si la valeur par défaut convient)

Éditer du css

- Avec un éditeur de texte tout simple (notepad++ par exemple)
 - En local ou sur un serveur
- Avec un environnement de développement (vscode, webstorm, etc.)
- Edition en ligne
 - dabblet.com (cache les préfixes)
<http://dabblet.com/gist/9314223>
 - jsbin.com
 - Etc.

Les « inspecteurs » intégrées dans les navigateurs

- Clic-droit « inspect » (ou inspecter)
 - F12 sous windows (exemple ici avec chrome)
- Permettent de tester sur la page (plutôt pour « essayer »)
 - <https://developer.chrome.com/devtools/docs/elements-styles>



Une erreur dans une déclaration ?

- La ligne/déclaration est ignorée
 - et peut-être la suivante ou la précédente, voire toute une partie du fichier CSS si l'erreur est un oubli de " ou de)
 - Exemple : oubli d'un « ; » terminal : les deux propriétés (celle de l'oubli et la suivante s'il y en a une avant le }) sont ignorées
 - Exemple: un espace de trop (nous verrons plus tard les unités)
 - « 2rem » est valide
 - « 2 rem » ne l'est pas
- Le navigateur passe outre et ignore donc les lignes impactées...
 - Ouf, sinon les préfixes seraient ingérables

Intégrer le css dans la page web ?

- Plusieurs possibilités :
 - Balise link : lier un fichier contenant que du CSS avec la page
 - Balise style : écrire du CSS dans la page web
 - Css dans l'attribut « style » d'une balise : écrire des déclarations directement sur l'élément concerné
- **Balise link la plus pratique**
partage d'un style entre plusieurs pages
- Les autres plutôt utilisés par des programmes qui génèrent des pages web (Content Management System)

Mettre du CSS dans une balise

- Css dans l'attribut « style » d'une balise
- Ne contient que des déclarations, sans sélecteurs
- Ces déclarations ne s'appliquent qu'à l'élément à qui appartient l'attribut style

- Exemples :

```
<h1 style="color: red;"> ... </h1>
```

```
<p style="color: blue; font-weight: bold;"> ...  
</p>
```

Intégrer le css dans la page web avec une (ou plusieurs) balise <style>

- Balise style : le css est dans le fichier html

```
<style>  
    p { background-color: grey; }  
</style>
```

- Généralement dans le <head> (en fait là où des méta-données peuvent être utilisées)
- Quelques attributs particuliers :
 - media : précise sur quel dispositif le style s'applique
Par défaut c'est pour tous les dispositifs
c'est une « media query » (c.f. plus tard)
 - disabled (si présent, sous la forme disabled="disabled"), le style n'est pas pris en compte

Intégrer le css dans la page web avec une (ou plusieurs) balise <link>

- Balise link : lien vers un fichier
`<link rel="stylesheet" type="text/css" media="screen,print" href="fleur.css" />`
 - **<link> avec un rel forcément placée dans <head>**
 - utilisable dans autant de pages que voulues
 - Plusieurs balises link possibles (le navigateur fait des requêtes supplémentaires pour chaque balise link)
 - <https://html.spec.whatwg.org/multipage/semantics.html#the-link-element>
- Généralement dans le <head> (en fait là où des méta-données peuvent être utilisées)
- 4 attributs (+ l'attribut title)
 - Attribut **rel="stylesheet"** **obligatoire** (pour lier une feuille de styles)
 - type : indique le format du style.
Par défaut c'est type="text/css"
 - media : précise sur quel dispositif le style s'applique
Par défaut c'est pour tous les dispositifs
c'est une « media query » (c.f. plus tard)
 - href : indique le chemin (local ou via http://) vers le fichier css

Illustration : image d'une page sans CSS



```
1 <!doctype html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>first contact with CSS</title>
7 </head>
8 <body>
9
10 <header>
11   <h1>My first contact with CSS</h1>
12 </header>
13
14 <section>
15
16   <article>
17     Michel Buffa is an associate professor in computer science at Université Nice Sophia
18     Antipolis. He is specialized in web and web semantics (collaborative web, social
19     networks, web of things). In the W3C, he is in charge of the HTML5 courses.
20   </article>
21
22   <aside>
23     Michel on tweeter : <a href="https://twitter.com/micbuffa">@micbuffa</a>
24   </aside>
25
26   <article>
27     Philippe Renevier Gonin is also an associate professor in computer science at
28     Université Nice Sophia Antipolis. He is specialized in HCI.
29   </article>
30 </section>
31 </body>
32 </html>
```

Hyper Text Markup Language file | length: 1106 | lines: 45 | Ln: 14 Col: 1 Sel: 0 | Dos\Windows | UTF-8 | INS

La même page avec du css

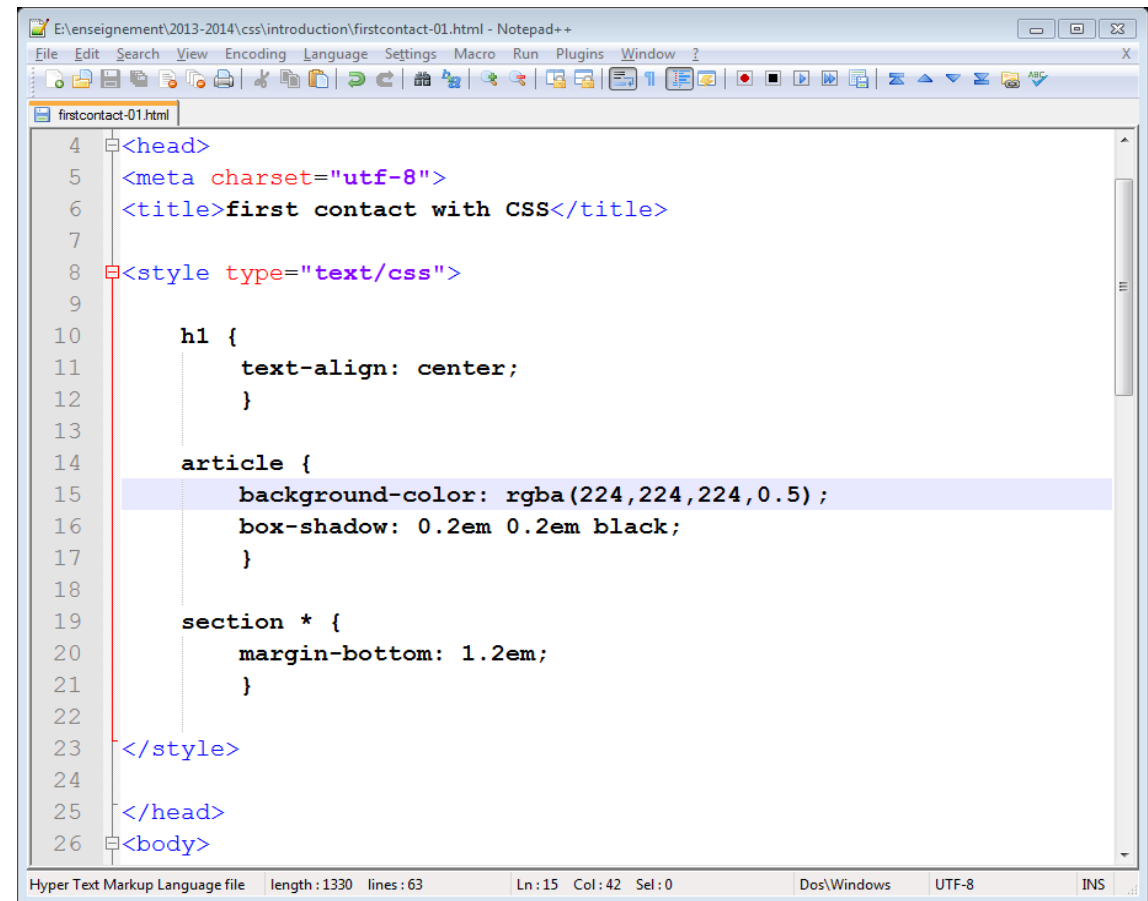
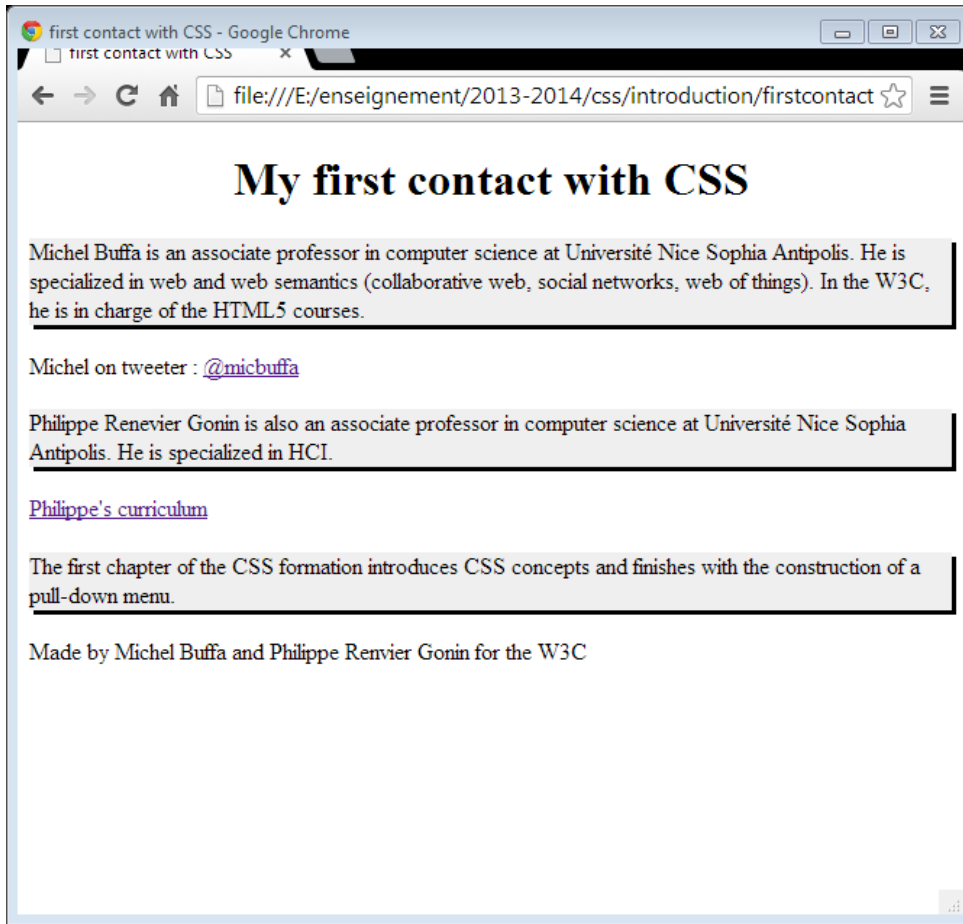


Illustration d'un sélecteur

```
<style type="text/css">
```

```
h1 {  
  text-align: center;  
}  
  
article {  
  background-color: rgba(224,224,224,0.5);  
  box-shadow: 0.2em 0.2em black;  
}  
  
section * {  
  margin-bottom: 1.2em;  
}  
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renevier Gonin for the W3C

Le même sélecteur utilisé pour d'autres éléments

```
<style type="text/css">
```

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

```
article {  
  background-color: rgba(224,224,224,0.5);  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

Une combinaison de sélecteurs

```
<style type="text/css">
```

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

```
article {  
  background-color: rgba(224,224,224,0.5);  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

mise en valeur (propriété: valeur;)

```
<style type="text/css">
```

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

```
article {  
  background-color: rgba(224,224,224,0.5) ;  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

mise en valeur (propriété: valeur;)

```
<style type="text/css">
```

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

```
article {  
  background-color: rgba(224,224,224,0.5);  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

mise en valeur (propriété: valeur;)

```
<style type="text/css">
```

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

```
article {  
  background-color: rgba(128,255,128,1);  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the ~~HTML5~~ courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

mise en valeur (propriété: valeur;)

```
<style type="text/css">
```

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

```
article {  
  /* changing the property */  
  /* from background-color */  
  /* to : */  
  color: rgba(128,255,128,1);  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

Des propriétés pour certains éléments

```
<style type="text/css">
```

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

```
article {  
  background-color: rgba(224,224,224,0.5);  
  box-shadow: 0.2em 0.2em black;  
}
```

```
section * {  
  margin-bottom: 1.2em;  
  color: #880000;  
}
```

```
</style>
```

My first contact with CSS

Michel Buffa is an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in web and web semantics (collaborative web, social networks, web of things). In the W3C, he is in charge of the HTML5 courses.

Michel on tweeter : [@micbuffa](#)

Philippe Renevier Gonin is also an associate professor in computer science at Université Nice Sophia Antipolis. He is specialized in HCI.

[Philippe's curriculum](#)

The first chapter of the CSS formation introduces CSS concepts and finishes with the construction of a pull-down menu.

Made by Michel Buffa and Philippe Renvier Gonin for the W3C

color s'applique à tous les éléments
margin-bottom ne s'applique pas à <a> (comme la plus part des éléments *phrasing*)

Quelques sélecteurs (simples)

Pour jouer rapidement avec le css

5 sélecteurs « de bases »

Selecteur	Signification	Lien et nom	Version CSS
*	any element	Universal selector	2
E	an element of type E	Type selector	1
E.className	an E element whose class is "className" (the document language specifies how class is determined).	Class selectors	1
E#myid	an E element which "id" attribute is equal to "myid".	ID selectors	1
E F	an F element descendant of an E element	Descendant combinator	1

rappel

Un document html = un arbre

`<h1>Le dormeur du val</h1>`

`<p>C'est un trou de verdure où chante une rivière
`

Accrochant follement aux herbes des haillons

D'argent ; où le soleil, de la montagne fière,

Luit : c'est un petit val qui mousse de rayons.</p>

`<p>Un soldat jeune, bouche ouverte, tête nue,
`

Et la nuque baignant dans le frais cresson bleu,

Dort ; il est étendu dans l'herbe, sous la nue,

Pâle dans son lit vert où la lumière pleut.</p>

`<p>Les pieds dans les glaïeuls, il dort. Souriant comme
`

Sourirait un enfant malade, il fait un somme :

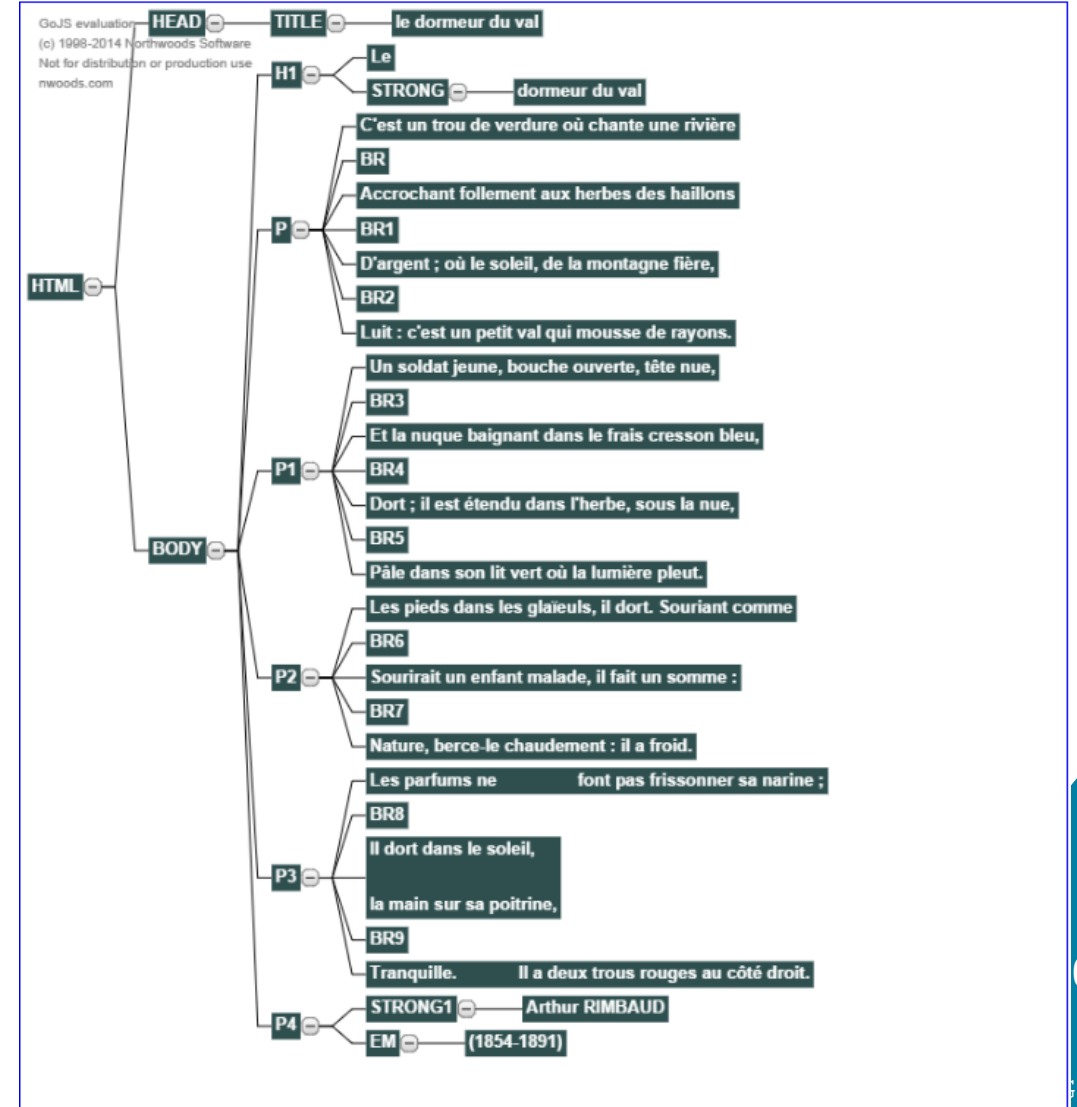
Nature, berce-le chaudement : il a froid.</p>

`<p>Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil,`

la main sur sa poitrine,

Tranquille. Il a deux trous rouges au côté droit.</p>

`<p>Arthur RIMBAUD(1854-1891)</p>`



* (universal selector)

Le dormeur du val

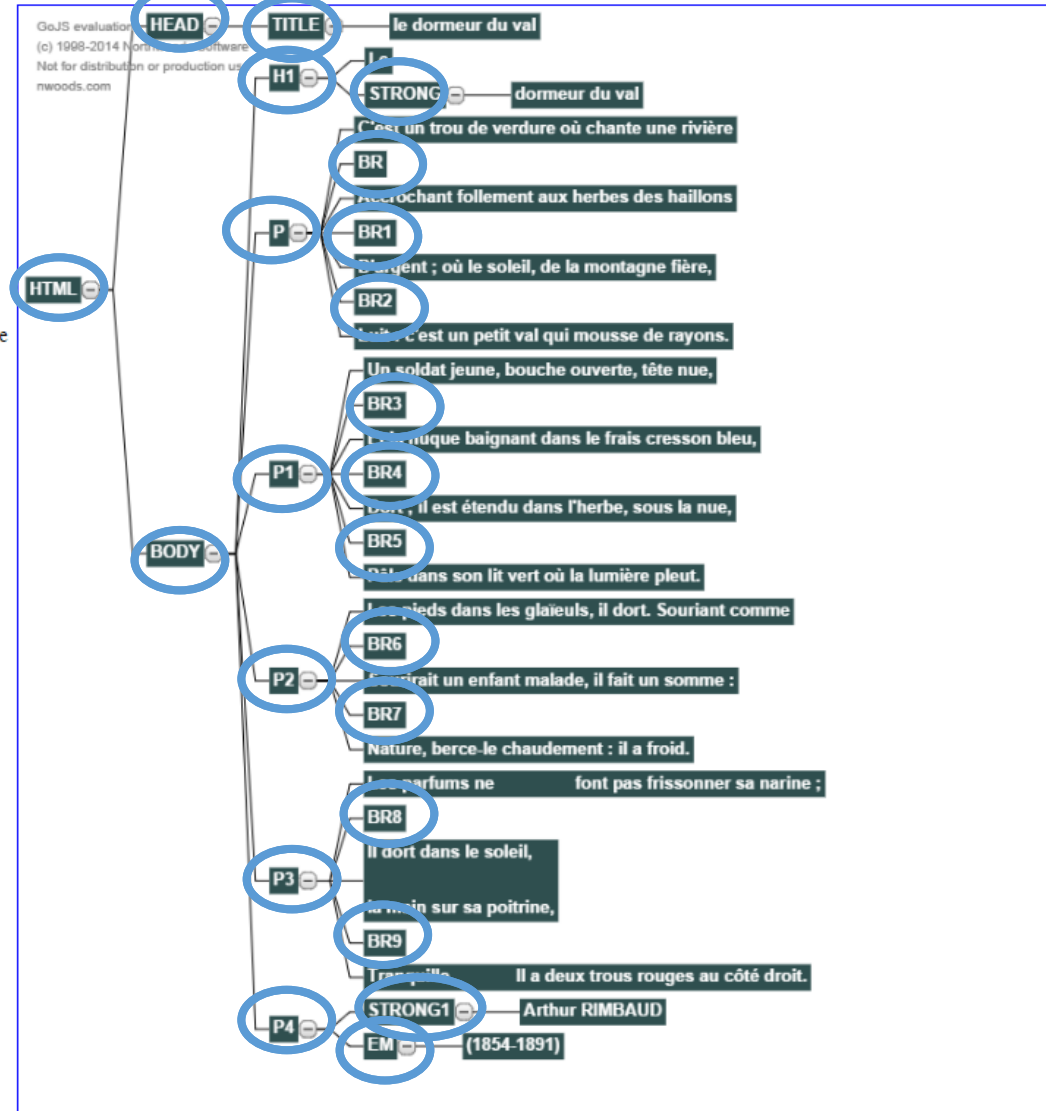
C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

Arthur RIMBAUD (1854-1891)



*** : toutes les balises sont concernées**
(mais toutes les propriétés ne s'appliquent pas à toutes les balises)

E (Type Selector)

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

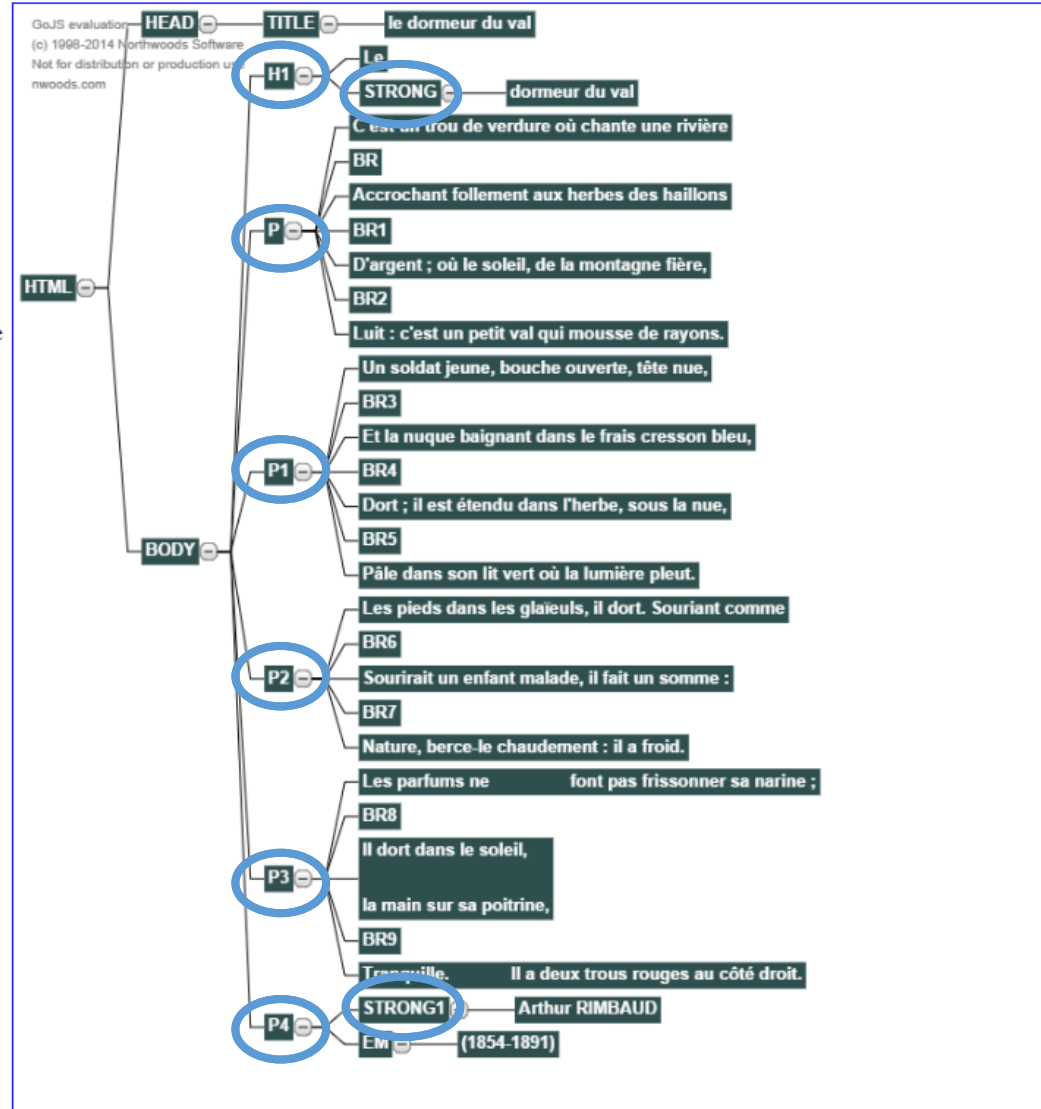
Arthur RIMBAUD (1854-1891)

E : désigne toutes les occurrences d'un élément (= une balise) où qu'elle soit

h1 : désigne tous les <h1>

p : désigne tous les <p>

strong : désigne tous les



E.className (Class Selectors)

```
1 <p class="classname">...</p>
2 <h1 class="classname">...</h1>
3 <p class="classname">... <em class="classname">emphasis text</em>...</p>
4 <h2>...</h2>
5 <h1 class="anotherclass">...</h1>
```

- Pour désigner des balises qui ont un **attribut « class » (de html)**
- Se combine avec le « type selector » et l'« universal selector »
- Exemples :
 - *.classname => désigne les <p> lignes 1 et 3, le <h1> ligne 2 et le ligne 3
 - p.classname => désigne les les <p> lignes 1 et 3
 - p.anotherclass => rien, aucune balise <p> n'a son attribut "class" dont la valeur est "anotherclass".
- a class name *"can contain only the characters [a-zA-Z0-9] and ISO 10646 characters U+00A0 and higher, plus the hyphen (-) and the underscore (_); they cannot start with a digit, two hyphens, or a hyphen followed by a digit."*

Plusieurs classes

```
<p class="first">A paragraph of the class "first"</p>  
<p class="second">A paragraph of the class "first"</p>  
<p class="first second">A paragraph of the class "first" and  
of the class second</p>
```

<code>p.first</code>	=> le premier et le troisième <p>
<code>p.second</code>	=> le deuxième et le troisième <p>
<code>p.first.second</code>	=> le troisième <p>
<code>p.second.first</code>	=> le troisième <p>

<http://dabblet.com/gist/7687090f00b2fe73c920>

E#myId (id selector)

- Pour désigner **LA** balise par la valeur de son **attribut id**
- « id » est un attribut général (à toutes les balises).
- Chaque « id » doit être unique

E F (descendant combinator)

- Pour désigner dans l'arbre tous éléments F (= une balise) qui sont en dessous d'un autre E
- E est un ancêtre de F
- Pour qu'un « F » soit désigner par « E F », il faut que si on « remonte » (si la racine est en haut) dans l'arbre à partir d'un F vers la racine, il faut passer par un E

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaieuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

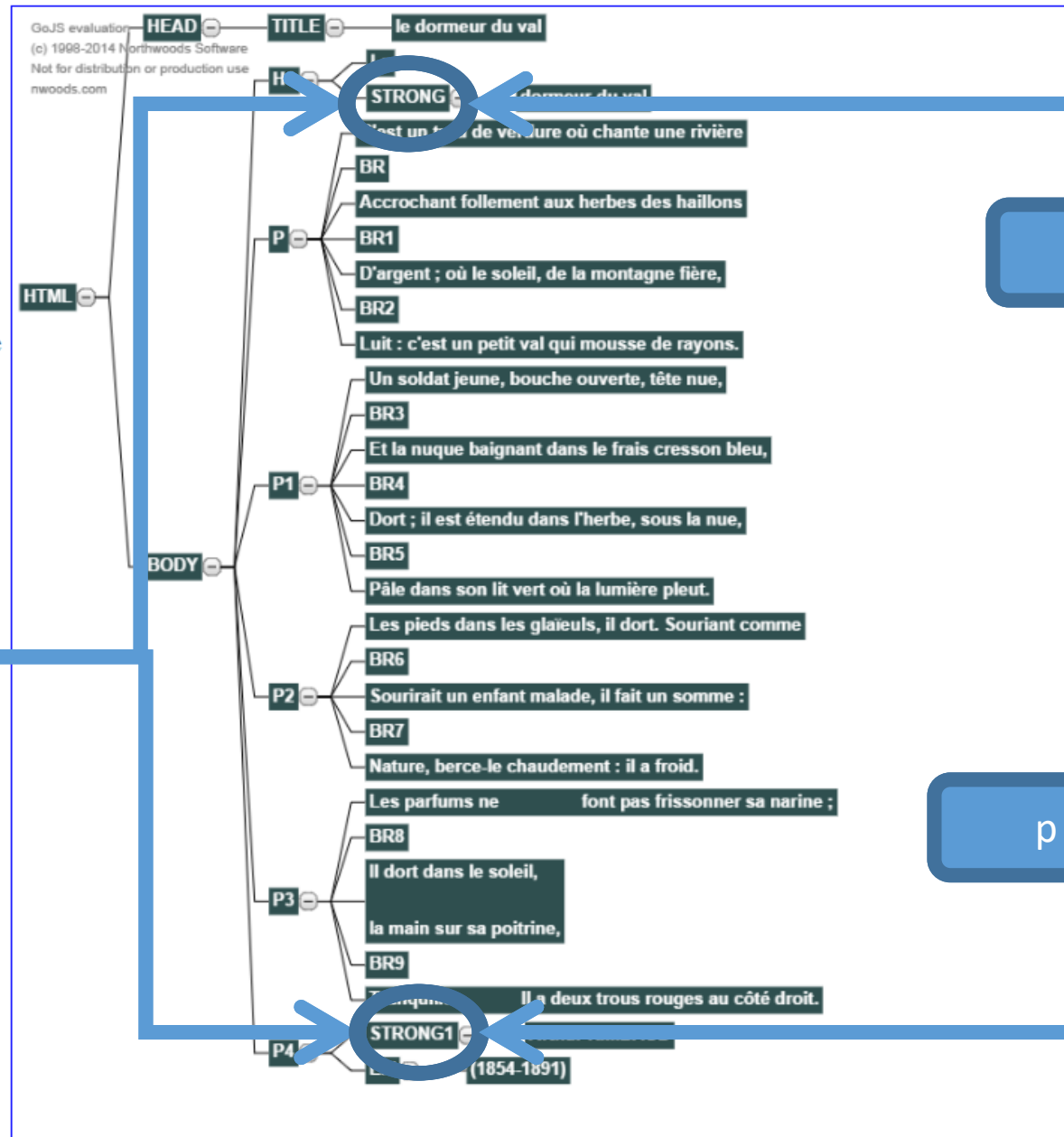
Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

Arthur RIMBAUD (1854-1891)

body strong

h1 strong

p strong



Les sélecteurs se combinent

- `p *`
 - Combinaison de « descendant combinator », « type selector » (`p`) et de « universal selector » (*)
 - Désigne tous les éléments contenus dans un `P`
- `p#premier *.special`
 - Combinaison de « descendant combinator », « type selector » (`p`), de « id selector » (`#premier`), de « class selector » (`.special`) et de « universal selector » (*)
 - Désigne toutes les balises dont l'attribut class contient le mot « special » et qui sont un descendant d'une balise `p` dont l'id vaut « premier »

Exemple de combinaison

Le dormeur du val

C'est un trou de verdure où chante une rivière
Accrochant follement aux herbes des haillons
D'argent ; où le soleil, de la montagne fière,
Luit : c'est un petit val qui mousse de rayons.

Un soldat jeune, bouche ouverte, tête nue,
Et la nuque baignant dans le frais cresson bleu,
Dort ; il est étendu dans l'herbe, sous la nue,
Pâle dans son lit vert où la lumière pleut.

Les pieds dans les glaïeuls, il dort. Souriant comme
Sourirait un enfant malade, il fait un somme :
Nature, berce-le chaudement : il a froid.

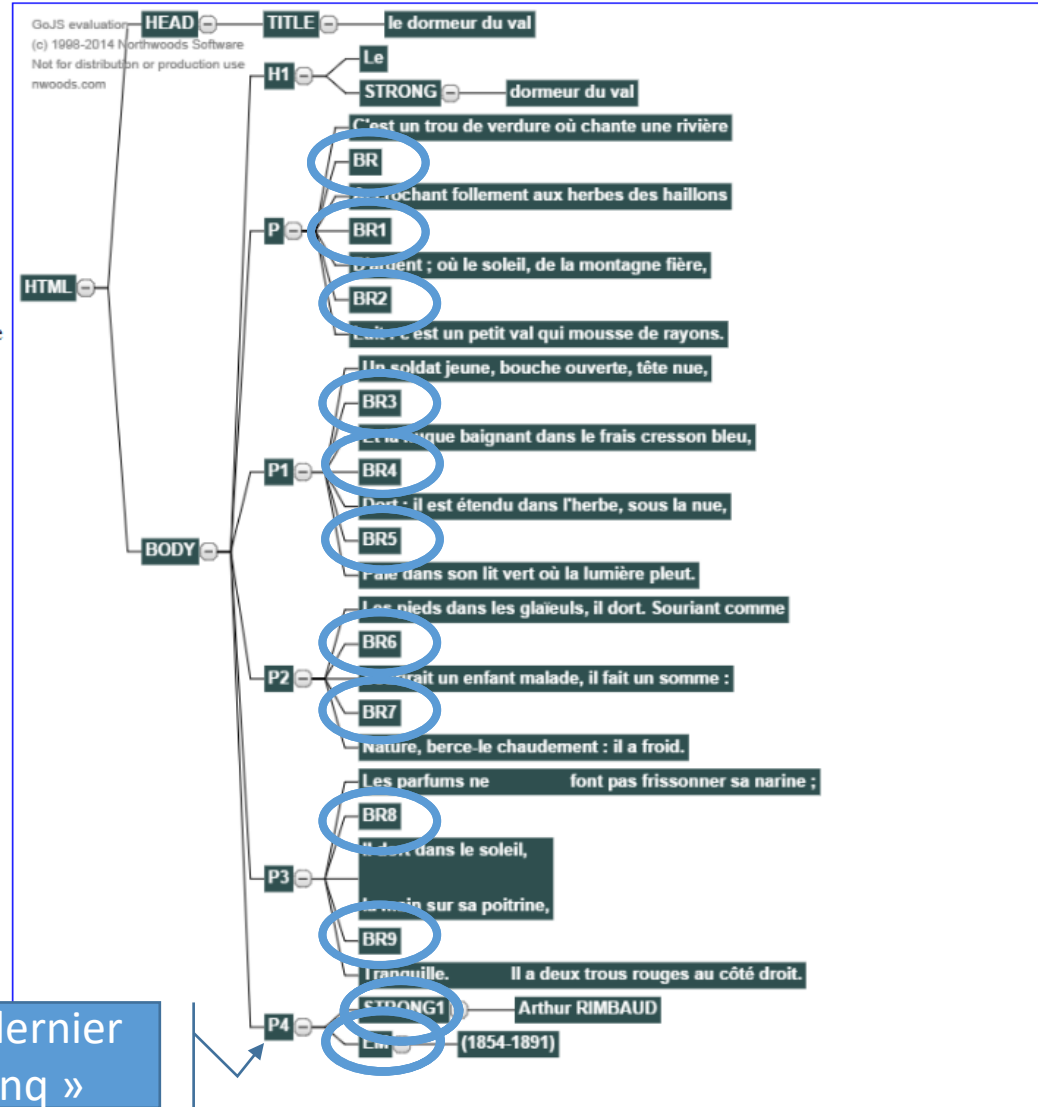
Les parfums ne font pas frissonner sa narine ;
Il dort dans le soleil, la main sur sa poitrine,
Tranquille. Il a deux trous rouges au côté droit.

Arthur RIMBAUD (1854-1891)

p *

p#cinq *

L'id de ce dernier
p est « cinq »



Quelques propriétés CSS... ... pour pouvoir jouer avec...

color et background-color

Color & images values

border, border-radius, box-shadow

Length

Calc

margin & padding

Quelques éléments à propos de font

Quelques éléments pour « jouer »

- Des propriétés très fréquentes et plutôt simple
- Couleur de texte - valeur des couleurs - et les fonds – valeur des images...
- Les encadrements (border) – valeur des dimensions - , les coins arrondis et les ombres
- Les marges internes et externes
- Quelques éléments sur les polices de caractères

color

- <http://www.w3.org/TR/css3-color/>
- Indique la couleur texte
- S'applique à tous les éléments
- Valeur par défaut : dépend du navigateur (et de l'élément)
- Il y a héritage de la couleur du texte du parent
- Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles : une couleur

Des couleurs en valeurs

- inherit | initial
- currentColor (la couleur en cours, plutôt pour border)
- transparent : une couleur invisible (complètement transparente)
- Un nom anglais de la couleur (c.f. <http://www.w3.org/TR/css3-color/#svg-color>)
- RGB
 - Un code rgb : #FF000 ou rgb(255,0,0) ou rgb(100%,0,0) pour du rouge
 - Un code rgba : rgba(255,0,0,0.5) pour un rouge semi transparent

Background : vue d'ensemble

- Une notation raccourcie
- Pour spécifier le fond : couleur ou image
- Spécifications complètes et complexes
- Attention, cela peut rendre la page illisible
- <http://dabblet.com/gist/c1e399c986b3fbf9a223>

background

- Indique le fond de l'élément, en précisant plusieurs propriétés à la fois
- S'applique à tous les éléments
- Valeur par défaut : pas d'image et un fond transparent
- Il n'y a pas d'héritage du fond du parent (on le voit généralement car transparent)
- Cette propriété est prise en compte (partiellement) dans les animations CSS
- Valeurs possibles :
 - `<bg-image> || <position> [/ <bg-size>]? || <repeat-style> || <attachment> || <box> || <box> || <'background-color'>`
- Plusieurs images, avec la dernière comme ci-dessus, et les autres quasi-identiques (sauf qu'il n'y a pas de `background-color`), séparées par des « , »
 - La première déclaration est le fond du dessous, la dernière est le fond du dessus
- Les valeurs possibles
 - \longleftrightarrow correspondance avec une propriété spécialisée
 - `<bg-image>` : une valeur « image »
 \longleftrightarrow `background-image`
 - `<position>` la position de la première image (si répété) par rapport à l'élément
 \longleftrightarrow `background-position`
 - `<bg-size>` taille de l'image
 \longleftrightarrow `background-size`
 - `<repeat-style>` fond de répété l'image si besoin
 \longleftrightarrow `background-repeat`
 - `<attachment>` comportement en cas de scroll
 \longleftrightarrow `background-attachment`
 - 1er `<box>` pour dire où commence le repère (contour, padding, content)
 \longleftrightarrow `background-origin`
 - 2ième `<box>` pour dire comment le fond est « rempli » (contour, padding, content)
 \longleftrightarrow `background-clip`
 - `<'background-color'>` la couleur de fond (s'il n'y a pas d'image ou si elle est transparente)
 \longleftrightarrow `background-color`

Une image comme valeur

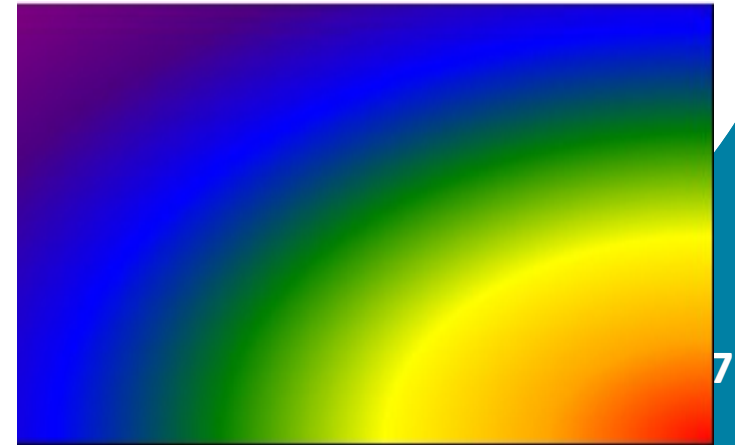
- Une url : `url(" https://www.w3.org/assets/logos/w3c/w3c-no-bars.svg")`
- Une image calculée
 - linear-gradient
 - On indique une direction (ou un angle) => passe par le centre
 - Des couleurs
 - À des distances précisées et/ou réparties équitablement
 - radial-gradient
- <http://dabblet.com/gist/bbbb6b5361fbb0e0da8c>

Exemple de gradient

```
background-image: linear-  
gradient(darkblue, cornflowerblue  
50%);
```

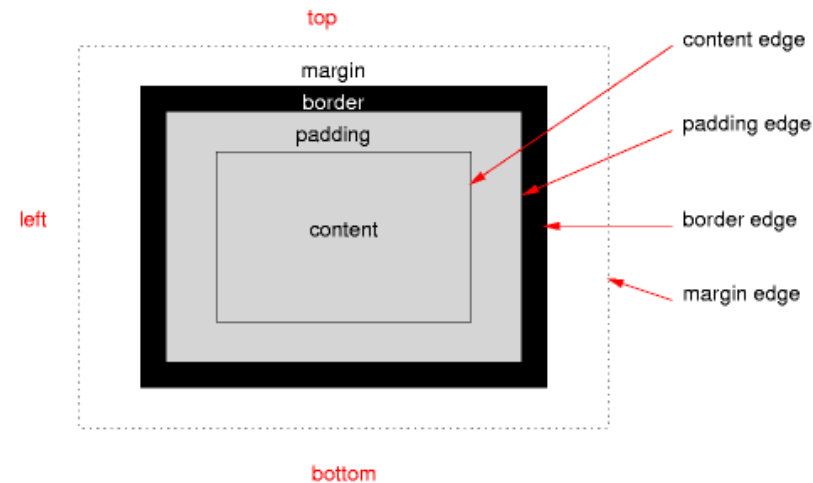


```
background-image: radial-gradient(at  
right bottom, red, orange, yellow,  
green, blue, indigo, purple)
```



Encadrer les éléments: border

- La propriété *border* permet de dessiner (spécifier) l'apparence du contour des éléments
- Se décompose en 4 traits et en 3 propriétés (couleur, épaisseur, style)
- Il y a donc 12 propriétés et 8 notations raccourcies



Couleur du contour : border-color

- C'est un des raccourcis
- Permet de spécifier les couleurs du contours pour le haut, la droite, le bas et la gauche

border-top-color | border-right-color |
border-bottom-color | border-left-color

`border-top-color | border-right-color |
border-bottom-color | border-left-color`

- Indique la couleur du trait du contour, du côté spécifié
- S'applique à tous les éléments
- Valeur par défaut : `currentColor` (pour texte, la couleur d'écriture, par défaut généralement du noir)
- Il n'y a pas d'héritage des couleurs de traits
- Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles :
 - Une couleur (c.f. `background`)

border-color

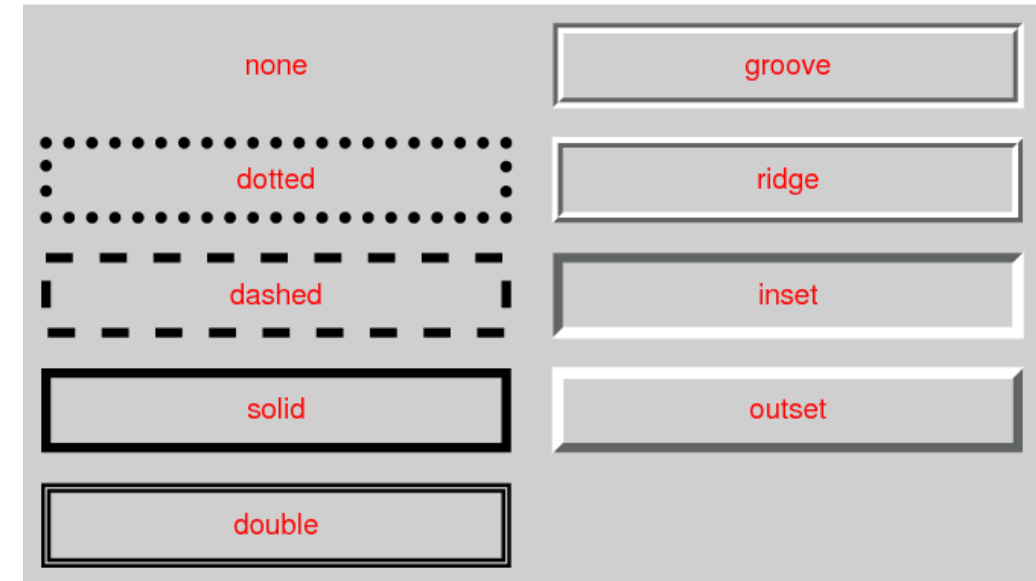
- Raccourcit pour déclarer les 4 couleurs de traits en une fois , avec 4, 3, 2 ou 1 valeur(s)
- **border-color: val1 val2 val3 val4;**
est équivalent à
border-top-color: val1;
border-right-color : val2;
border-bottom-color : val3;
border-left-color : val4;
- **border-color: val1 val2 val3;**
est équivalent à border-color: val1 val2 val3 val2;
soit:
border-top-color : val1;
border-right-color : val2;
border-bottom-color : val3;
border-left-color : val2;
- **border-color: val1 val2;**
est équivalent à border-color: val1 val2 val1 val2;
soit:
border-top-color : val1;
border-right-color : val2;
border-bottom-color : val1;
border-left-color : val2;
- **border-color: val1;**
est équivalent à
border-color: val1 val1 val1 val1; soit:
border-top-color : val1;
border-right-color : val1;
border-bottom-color : val1;
border-left-color : val1;

Quel(s) trait(s) pour le contour ? border-style

- Fonctionne comme border-color
- C'est donc le raccourci pour
border-top-style | border-right-style |
border-bottom-style | border-left-style

border-top-style | border-right-style | border-bottom-style | border-left-style

- Indique le style du trait du contour, du côté spécifié
- S'applique à tous les éléments
- Valeur par défaut : none
- Il n'y a pas d'héritage des couleurs de traits
- Cette propriété n'est pas prise en compte dans les animations CSS
- Valeurs possibles :
none | hidden | dotted | dashed | solid |
double | groove | ridge | inset | outset



border-style

- Raccourcit pour déclarer les 4 style de traits en une fois , avec 4, 3, 2 ou 1 valeur(s)
- **border-style: val1 val2 val3 val4;**
est équivalent à
border-top-style: val1;
border-right-style : val2;
border-bottom-style : val3;
border-left-style : val4;
- **border-style: val1 val2 val3;**
est équivalent à border-style: val1 val2 val3 val2;
soit:
border-top-style : val1;
border-right-style : val2;
border-bottom-style : val3;
border-left-style : val2;
- **border-style: val1 val2;**
est équivalent à border-style: val1 val2 val1 val2;
soit:
border-top-style : val1;
border-right-style : val2;
border-bottom-style : val1;
border-left-style : val2;
- **border-style: val1;**
est équivalent à
border-style: val1 val1 val1 val1; soit:
border-top-style : val1;
border-right-style : val1;
border-bottom-style : val1;
border-left-style : val1;

L'épaisseur du trait du contour : border-width

- Fonctionne comme border-color et border-style
- C'est donc un raccourci pour

border-top-width | border-right-width |
border-bottom-width | border-left-width

border-top-width | border-right-width |
border-bottom-width | border-left-width

- Indique l'épaisseur du trait du contour, du côté spécifié
- S'applique à tous les éléments
- Valeur par défaut : medium (mais comme style est à « none », l'épaisseur retenue dans les calculs est 0)
- Il n'y a pas d'héritage des couleurs de traits
- Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles :
 - thin | medium | thick
 - une longueur (positive)

Les longueurs en css

- 0 (distance nulle)
- Sinon, une longueur est un nombre auquel on accole une unité
- Deux types d'unité
 - Unités relatives
 - Unités absolues

Quelques unités relatives

unit	relative to
<code>'em'</code>	font size of the element
<code>'ex'</code>	<i>x-height</i> of the element's font
<code>'ch'</code>	width of the "0" (ZERO, U+0030) glyph in the element's font
<code>'rem'</code>	font size of the root element
<code>'vw'</code>	1% of viewport's width
<code>'vh'</code>	1% viewport's height
<code>'vmin'</code>	1% of viewport's smaller dimension
<code>'vmax'</code>	1% of viewport's larger dimension

- em, ex, ch se réfère à la police de caractère de l'élément même
 - em, unité d'imprimerie
en général, 1.2em = hauteur totale d'une ligne
- rem se réfère à la police de caractère de l'élément racine (html) => la même pour tous
- Viewport : zone dans laquelle est affichée le document
 - vmin : le plus petit de vw et vh
 - vmax : le plus grand de vw et vh

Unités absolues

unit	definition
'cm'	centimeters
'mm'	millimeters
'in'	inches; 1in is equal to 2.54cm
'px'	pixels; 1px is equal to 1/96th of 1in
'pt'	points; 1pt is equal to 1/72nd of 1in
'pc'	picas; 1pc is equal to 12pt

- Mesures surtout pratique pour l'impression
- Variation d'un dispositif à un autre
- px à la fois pratique (pour les images, etc.) mais à la fois problématique (changement de taille, de taille de police, etc.)

calc: une expression pour calculer

- Pour les opérations + ; - ; * ; /
- Remplace un nombre (et si l'unité est une longueur, une longueur)
- Avec des unités hétérogènes
- calc(opération)
l'opération est une suite de nombres (avec unité ou pas, et même des pourcentages) et d'opérateurs
 - avec des espaces après les unités
- Exemple :
calc(1rem - 2px)
calc(50% - 1rem + 10px) **// pas de % pour border-width**
- Attention donc aux unités et aux résultats (négatif...)

border-width

- Raccourcit pour déclarer les 4 épaisseurs de traits en une fois , avec 4, 3, 2 ou 1 valeur(s)
- **border-width: val1 val2 val3 val4;**
est équivalent à
border-top-width: val1;
border-right-width : val2;
border-bottom-width : val3;
border-left-width : val4;
- **border-width: val1 val2 val3;**
est équivalent à border-width: val1 val2 val3 val2;
soit:
border-top-width : val1;
border-right-width : val2;
border-bottom-width : val3;
border-left-width : val2;
- **border-width: val1 val2;**
est équivalent à border-width: val1 val2 val1 val2;
soit:
border-top-width : val1;
border-right-width : val2;
border-bottom-width : val1;
border-left-width : val2;
- **border-width: val1;**
est équivalent à
border-width: val1 val1 val1 val1; soit:
border-top-width : val1;
border-right-width : val1;
border-bottom-width : val1;
border-left-width : val1;

border-top | border-right | border-bottom | border-left | border

- border-top | border-right |
border-bottom | border-left
 - Définissent l'épaisseur et/ou la couleur et/ou le style du trait du côté qu'ils spécifient
 - L'ordre n'influence pas
- border
 - Définir l'épaisseur et/ou la couleur et/ou le style du trait pour les 4 côtés en même temps
 - L'ordre n'influence pas
- Attention à l'ordre d'utilisation
 - par exemple, si border est utilisé après border-top, la propriété « écrassera » border-top.
 - Et réciproquement
 - C.f. règle de priorité avec les sélecteurs

Les coins arrondis

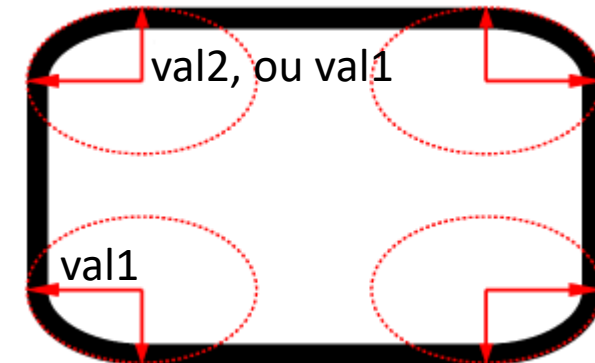
- S'applique aux contours (*border*), au fond (*background*) et aux ombres (*box-shadow*)
- Réduit la taille du contenu... des choses peuvent dépasser (c.f. overflow au moment du menu) : il faut utiliser du padding (c.f. plus loin) pour éviter un débordement
- Une fois encore : une propriété pour chaque coin, résumé en une :
 - border-top-left-radius | border-top-right-radius | border-bottom-right-radius | border-bottom-left-radius
 - border-radius
- Les effets d'épaisseur de trait différent de chaque côté ou de trait trop épais (plus que l'arrondi) sont prévu



Source : W3C

border-top-left-radius | border-top-right-radius | border-bottom-right-radius | border-bottom-left-radius

- Indique l'incurvation du coin concerné
- S'applique à tous les éléments
- Valeur par défaut : 0
- Il n'y a pas d'héritage de marge interne
- Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles :
 - Une ou deux valeurs (positives)
 - Qui peuvent être une longueur ou un pourcentage
 - % de la largeur du contour si c'est la 1^{ère} valeur
 - % de la hauteur du contour si c'est la 2^{ième}
 - 50% sur deux coins consécutifs : forme ovale (partielle)
 - Qui définissent l'ellipse dessinant l'arrondi
 - Si une des valeurs est 0 : le coin est carré



Source : W3C

border-radius

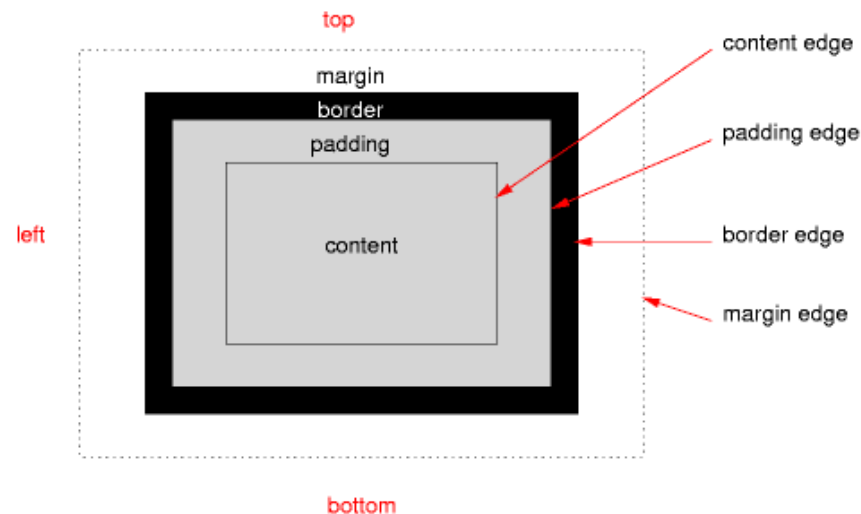
- Revient à déterminer en une fois
border-top-left-radius | border-top-right-radius |
border-bottom-right-radius | border-bottom-left-radius
- Il peut y avoir entre 1 et 8 valeurs, avec ou pas un « / » séparateur
 - S'il n'y a pas de / : il y a au maximum 4 valeurs, à l'instar des border-color, border-width, etc.
 - S'il y a un / : le qui est avant concerne la 1ère valeur pour chaque coin (axe horizontal de l'ellipse) et ce qui est après l'axe vertical
 - toujours en 1 à 4 « couple », en commençant par le coin en haut à gauche puis en allant vers la droite
- <http://dabblet.com/gist/e2b5dbea5be73286f32d>

Les ombres : box-shadow

- Permet de dessiner une ou plusieurs ombres à une boîte
- S'applique à tous les éléments
- Valeur par défaut : none
- Il n'y a pas d'héritage d'ombre
- Cette propriété est prise en compte dans les animations CSS
- <http://dabblet.com/gist/a1a32066c5f56e72eda8>
- Valeurs possibles :
 - none | une ombre | une suite d'ombres séparées par une « , »
 - une ombre est (pas nécessairement dans l'ordre) :
 - facultatif : le mot-clef « inset » (change l'effet de l'ombre)
 - Sans ce mot-clef, l'ombre fait « ressortir » l'élément
 - Avec ce mot-clef, l'ombre fait « s'enfoncer » l'élément (change l'ombre de côtés)
 - Entre 1 et 4 longueurs, les valeurs non spécifiées seront remplacées par 0, qui sont, dans l'ordre
 - Le décalage horizontal (positif => vers la droite) (donne aussi l'épaisseur)
 - Le décalage vertical (positif => vers le bas) (donne aussi l'épaisseur)
 - Le flou (blur) (valeur >= 0) : 0, la fin de l'ombre est nette, sinon il y a une zone de flou
 - La diffusion (*spread*) : étend (> 0) ou rétracte (< 0) l'ombre dans toutes les directions (y compris contraires à celles indiquées)
 - Une couleur

Marges internes et externes

- Marges externes = *margin* = Espacement « vide » autour de l'élément, mesuré à partir de son contour (visible ou pas)
- Marges internes = *padding* = Espacement entre le contenu de l'élément et le trait du contour



Source : W3C

padding

- Utile pour espacer avec les traits du contour
- Ou pour éviter d'écrire dans la partie arrondie (voire à l'extérieur) d'un coin
- La propriété padding est un raccourci pour padding-top | padding-right | padding-bottom | padding-left
- Chacune des 4 propriétés correspond à l'espacement interne du côté spécifié

padding-top | padding-right | padding-bottom | padding-left

- Indique un espacement intérieur (entre le contenu et le contour)
- S'applique à tous les éléments
- Valeur par défaut : 0
- Il n'y a pas d'héritage de marge interne
- Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles :
 - Une longueur (10rem ou 5cm ou 12px etc.)
 - Un pourcentage de l'élément de la largeur (*block*) contenant l'élément pour lequel vous voulez modifier les marges internes
 - Dépend généralement de la largeur de la fenêtre...
 - % par rapport à la largeur du contenant (élément parent généralement) même pour les marges verticales
 - Vrai pour une écriture horizontale
 - Si l'écriture est vertical, c'est un pourcentage de la hauteur, et la « direction » des éléments est alors verticale

padding

- Raccourcit pour déclarer les 4 marges internes en une fois , avec 4, 3, 2 ou 1 valeur(s)
- **padding: val1 val2 val3 val4;**
est équivalent à
padding-top: val1;
padding-right: val2;
padding-bottom: val3;
padding-left: val4;
- **padding: val1 val2 val3;**
est équivalent à padding: val1 val2 val3 val2; soit:
padding-top: val1;
padding-right: val2;
padding-bottom: val3;
padding-left: val2;
- **padding: val1 val2;**
est équivalent à padding: val1 val2 val1 val2; soit:
padding-top: val1;
padding-right: val2;
padding-bottom: val1;
padding-left: val2;
- **padding: val1;**
est équivalent à
padding: val1 val1 val1 val1; soit:
padding-top: val1;
padding-right: val1;
padding-bottom: val1;
padding-left: val1;
- <http://dabblet.com/gist/266f3c0c98570bc8c5bb>

margin

- Espacement autour et extérieur à un élément
- Mesurés depuis les contours
- La propriété margin est un raccourci pour
margin-top | margin-right | margin-bottom | margin-left
- Chacune des 4 propriétés correspond à l'espacement externe du côté spécifié

margin-top | margin-right | margin-bottom | margin-left

- Indique un espacement extérieur (depuis le contour)
 - S'applique à tous les contenants (box) à quelques exceptions près
 - Valeur par défaut : 0
 - Il n'y a pas d'héritage de marge interne
 - Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles :
 - Une longueur (10rem ou 5cm ou 12px etc.)
 - Un pourcentage de l'élément de la largeur (*block*) contenant l'élément pour lequel vous voulez modifier les marges internes
 - Dépend généralement de la largeur de la fenêtre...
 - % par rapport à la largeur du contenant (élément parent généralement) même pour les marges verticales
 - Vrai pour une écriture horizontale
 - Si l'écriture est vertical, c'est un pourcentage de la hauteur, et la « direction » des éléments est alors verticale
 - auto
 - Le navigateur calcule les marges externes
 - Généralement, si l'élément ne prend pas toute la largeur possible, alors il sera centré horizontalement et il n'y aura pas de marge ni en haut ni en bas
 - Si l'élément prend toute la largeur (comme un <p>), alors il n'y aura pas d'effet

margin

- Raccourcit pour déclarer les 4 marges externes en une fois , avec 4, 3, 2 ou 1 valeur(s)
- **margin: val1 val2 val3 val4;**
est équivalent à
margin-top: val1;
margin-right: val2;
margin-bottom: val3;
margin-left: val4;
- **margin: val1 val2 val3;**
est équivalent à margin: val1 val2 val3 val2; soit:
margin-top: val1;
margin-right: val2;
margin-bottom: val3;
margin-left: val2;
- **margin: val1 val2;**
est équivalent à margin: val1 val2 val1 val2; soit:
margin-top: val1;
margin-right: val2;
margin-bottom: val1;
margin-left: val2;
- **margin: val1;**
est équivalent à
margin: val1 val1 val1 val1; soit:
margin-top: val1;
margin-right: val1;
margin-bottom: val1;
margin-left: val1;
- <http://dabblet.com/gist/4a89bc28519517908236>

Présentation partielle de font

- Mise en gras
- Mise en italic
- Changement de taille
- Changer de police (avec les polices installées pour le navigateur / dans l'OS)
- Spécifications complètes : <http://www.w3.org/TR/css3-fonts/>

font-weight

- Indique l'épaisseur de la police
- S'applique à tous les éléments
- Valeur par défaut : normal
- Chaque élément hérite de l'épaisseur d'écriture de l'élément parent
- Cette propriété est prise en compte dans les animations CSS
- Valeurs possibles :
normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900
 - 100 : le plus fin
 - 900 : le plus épais
 - normal = 400
 - bold = 700
 - bolder ou lighter est par rapport à l'épaisseur du trait héritée (i.e., celle du parent)

Inherited value	bolder	lighter
100	400	100
200	400	100
300	400	100
400	700	100
500	700	100
600	900	400
700	900	400
800	900	700
900	900	700

Source : W3C

font-style

- Indique l'inclinaison de la police (droite, italique ou oblique)
- S'applique à tous les éléments
- Valeur par défaut : normal (= « droit »)
- Chaque élément hérite de l'inclinaison d'écriture de l'élément parent
- Cette propriété n'est pas prise en compte dans les animations CSS
- Valeurs possibles :
 - normal | italic | oblique
 - italic est pour les fonts prévues à cet effet (créée pour ou avec)
 - oblique est une inclinaison « calculées »
 - Les rendus css utilise italic ou oblique à la place de l'autre si la font demandée n'existe pas
 - Au besoin, si aucun des deux n'existent, la police est calculée

font-size

- Indique la taille de la police
- S'applique à tous les éléments
- Valeur par défaut : medium (valeur par défaut du navigateur = 1em pour les polices retatables)
- Chaque élément hérite de la taille d'écriture de l'élément parent
- Cette propriété est prise en compte dans les animations CSS
- Les valeurs possibles :
 - xx-small | x-small | small | medium | large | x-large | xx-large en référence à une table interne au navigateur
 - larger | smaller en référence à la taille de police de l'élément parent. Opère un décalage par rapport aux mots-clefs ci-dessus (de 1 vers la gauche si c'est smaller, de 1 vers la droite si c'est larger). Le navigateur peut arrondir (comme il veut) si la taille du parent est entre deux mots-clefs
 - Une longueur (e.g., 1.5em | 2cm | 10px)
 - Un pourcentage par rapport à la taille de police de l'élément parent
- xx-large = 200% = 2em

CSS absolute-size values	xx-small	x-small	small	medium	large	x-large	xx-large
scaling factor	3/5	3/4	8/9	1	6/5	3/2	2/1
HTML headings	h6		h5	h4	h3	h2	h1

<http://dabblet.com/gist/d8f90dd17c425669f24f>

font-family

- Indique la taille de la police
- S'applique à tous les éléments
- Valeur par défaut : valeur par défaut du navigateur
- Chaque élément hérite de la taille d'écriture de l'élément parent
- Cette propriété n'est pas prise en compte dans les animations CSS
- Les valeurs possibles, une ou plusieurs, séparées par un « , » :
 - nom de police de caractère (qui doit être présente dans le navigateur), e.g., "Times New Roman"
 - Un nom générique parmi (le navigateur choisira) :

Aa

Constantia

Aa

Verdana

Aa

Corsiva

Aa

Comic Sans MS

Aa

Courier

- Plusieurs valeurs, car si la première n'est pas disponible, le navigateur passera à la suivante
 - Il est conseillé de placer une valeur générique à la fin (pour la robustesse)

text-align

- Dans <http://www.w3.org/TR/css3-text/#text-align-property>
- Indique l'alignement du texte
- S'applique à tous les éléments de niveau « block »
- Valeur par défaut : normal (= « droit »)
- Chaque élément hérite de l'alignement du texte du parent
- Cette propriété n'est pas prise en compte dans les animations CSS
- Valeurs possibles :
left | right | center | justify (classique)
start | end (par rapport aux début / fin d'écriture) | match-parent (par rapport aux bords / et la direction d'écriture)

Sélecteurs et priorités

Les différents sélecteurs et les règles pour savoir résoudre les conflits lorsque des déclarations sont en concurrences.

Sélecteurs

- Listes
 - <http://www.w3.org/TR/selectors/#selectors>
- Une liste de sélecteur est possible, en séparant ceux-ci par « , »
 - Permet de ne pas ré-écrire des blocs de déclaration
 - Potentiellement un usage de sélecteurs dans des sélecteurs (à partir du level 3)
- On va retenir :
 - :hover (quand la souris survole)
 - :nth-child / :nth-of-type pour choisir quelques uns (un sur deux par exemple)

Les sélecteurs (1/3)

Sélecteur	Signification	Lien et Nom	Version CSS
*	any element	Universal selector	2
E	an element of type E	Type selector	1
E[foo]	an E element with a "foo" attribute	Attribute selectors	2
E[foo="bar"]	an E element whose "foo" attribute value is exactly equal to "bar"	Attribute selectors	2
E[foo~="bar"]	an E element whose "foo" attribute value is a list of whitespace-separated values, one of which is exactly equal to "bar"	Attribute selectors	2
E[foo^="bar"]	an E element whose "foo" attribute value begins exactly with the string "bar"	Attribute selectors	3
E[foo\$="bar"]	an E element whose "foo" attribute value ends exactly with the string "bar"	Attribute selectors	3
E[foo*="bar"]	an E element whose "foo" attribute value contains the substring "bar"	Attribute selectors	3
E[foo ="en"]	an E element whose "foo" attribute has a hyphen-separated list of values beginning (from the left) with "en"	Attribute selectors	2
E:root	an E element, root of the document	Structural pseudo-classes	3
E:nth-child(n)	an E element, the n-th child of its parent	Structural pseudo-classes	3
E:nth-last-child(n)	an E element, the n-th child of its parent, counting from the last one	Structural pseudo-classes	3
E:nth-of-type(n)	an E element, the n-th sibling of its type	Structural pseudo-classes	3
E:nth-last-of-type(n)	an E element, the n-th sibling of its type, counting from the last one	Structural pseudo-classes	3
E:first-child	an E element, first child of its parent	Structural pseudo-classes	2

Source : W3C

Les sélecteurs (2/3)

Sélecteur	Signification	Lien et Nom	Version css
E:last-child	an E element, last child of its parent	Structural pseudo-classes	3
E:first-of-type	an E element, first sibling of its type	Structural pseudo-classes	3
E:last-of-type	an E element, last sibling of its type	Structural pseudo-classes	3
E:only-child	an E element, only child of its parent	Structural pseudo-classes	3
E:only-of-type	an E element, only sibling of its type	Structural pseudo-classes	3
E:empty	an E element that has no children (including text nodes)	Structural pseudo-classes	3
E:link E:visited	an E element being the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited)	The link pseudo-classes	1
E:active E:hover E:focus	an E element during certain user actions	The user action pseudo-classes	1 and 2
E:target	an E element being the target of the referring URI	The target pseudo-class	3
E:lang(fr)	an element of type E in language "fr" (the document language specifies how language is determined)	The :lang() pseudo-class	2
E:enabled E:disabled	a user interface element E which is enabled or disabled	The UI element states pseudo-classes	3
E:checked	a user interface element E which is checked (for instance a radio-button or checkbox)	The UI element states pseudo-classes	3

Source : W3C

Philippe Renevier-Gonin

Web : css

Les sélecteurs (3/3)

Sélecteur	Signification	Lien et Nom	Version css
E::first-line	the first formatted line of an E element	The ::first-line pseudo-element	1
E::first-letter	the first formatted letter of an E element	The ::first-letter pseudo-element	1
E::before	generated content before an E element	The ::before pseudo-element	2
E::after	generated content after an E element	The ::after pseudo-element	2
E.warning	an E element whose class is "warning" (the document language specifies how class is determined).	Class selectors	1
E#myid	an E element with ID equal to "myid".	ID selectors	1
E:not(s)	an E element that does not match simple selector s	Negation pseudo-class	3
E F (ou E >> F)	an F element descendant of an E element	Descendant combinator	1
E > F	an F element child of an E element	Child combinator	2
E + F	an F element immediately preceded by an E element	Adjacent sibling combinator	2
E ~ F	an F element preceded by an E element	General sibling combinator	3

Source : W3C

Ceux déjà vus

- Type selector
h1 { ... } /* désigne tous les h1 du document */
- Universal selector
***** { ... } /* désigne tous les éléments */
- Class selector
E.toto { ... } /* désigne les E dont l'attribut class="toto" */
/* équivalent à E[class~="toto"] (c.f. suivant) */
- Id selector
E#ald { ... } /* désigne le seul E dont l'attribut id="ald" */

Quelques Pseudo classes

- **:hover pour l'élément survolé par un pointeur (souris)**
- Les autres « moins utiles »
 - élément:first-child correspond au premier élément (celui-ci) enfant d'un autre élément.
 - Pour une ancre (a) :link et :visited (déconseillé car modification des habitudes des utilisateurs)
 - pseudo-classes dynamiques :active , :hover et :focus
 - :hover pour l'élément survolé par un pointeur (souris)
 - :active pour l'élément actif (entre le pressed et le realeased)
 - :focus pour l'élément qui reçoit l'attention (celui-ci acceptant les événements du clavier ou d'autres formes d'entrées de texte)
 - :target : Pour savoir si un élément est la cible d'un lien, url se termine par #ald - l'élément a un id qui vaut « ald »
 - État de l'élément graphique :enabled ; :disabled ; :checked

Place dans l'arbre du document

- `:root`
- **`:nth-child()`**
Le `:nth-child($an+b$)` pseudo-class represente un element qui a $an+b-1$ frère avant lui
Mots clefs : odd / even (*impair / pair*)
`tr:nth-child(2n+1) /* ligne impaire d'un tableau */`
`tr:nth-child(odd) /* idem */`
- **`:nth-last-child()`**
`tr:nth-last-child(-n+2) /* les deux dernières lignes d'une table */`
`foo:nth-last-child(odd) /* tous les éléments "foo" de n° impair, en comptant tous les enfants, en commençant à compter par la fin */`
- **`:nth-of-type()`**
`img:nth-of-type(2n) /* une image sur deux, en commençant par la seconde */`
- **`:nth-last-of-type()`**

Quelques exemples avec nth-child

- Dans un tableau, mettre une couleur de fond une ligne sur deux (pour aider à la lecture)
- Dans un classement, par exemple de ligue 1 (en 2016)
 - Les 3 premiers, potentiels qualifiés pour la champions' league
 - Le 4^e potentiel qualifié pour l'euroleague
 - Les 3 derniers, potentiels relégués en ligue 2

ligne 1	valeur 1
ligne 2	valeur 2
ligne 3	valeur 3
ligne 4	valeur 4
ligne 5	valeur 5

1	→		NICE
2	→		MONACO
3	→		PARIS SG
4	↑		LYON
5	↓		GUINGAMP
6	↓		RENNES
17	↓		NANTES
18	↑		NANCY
19	↓		LILLE
20	→		LORIENT

Notion de « fratrie »

- Un document html = un arbre
- Un élément (sauf <html>)
 - a un parent (la balise dans laquelle il est)
 - peut avoir des « frères » (des balises incluses dans le même parent)
- Exemple
 - tous les <tr> ci-contre sont dans la même fratrie
- Ici :

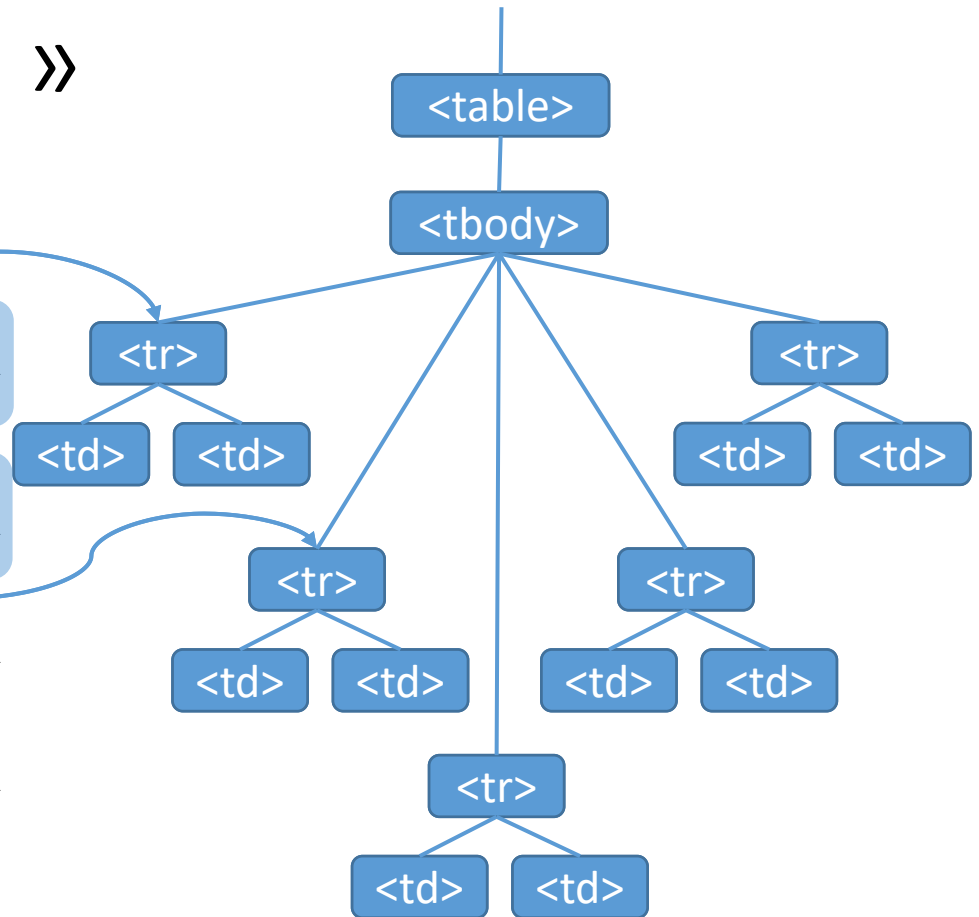

```
tr { background-color: gray; }
colorise toutes les lignes
```

ligne 1	valeur 1
ligne 2	valeur 2
ligne 3	valeur 3
ligne 4	valeur 4
ligne 5	valeur 5

```

<table>
  <tbody>
    <tr>
      <td>ligne 1</td>
      <td>valeur 1</td>
    </tr>
    <tr>
      <td>ligne 2</td>
      <td>valeur 2</td>
    </tr>
    <tr>
      <td>ligne 3</td>
      <td>valeur 3</td>
    </tr>
    <tr>
      <td>ligne 4</td>
      <td>valeur 4</td>
    </tr>
    <tr>
      <td>ligne 5</td>
      <td>valeur 5</td>
    </tr>
  </tbody>
</table>

```



➔ un sélecteur pour dire un sur 2

:nth-child : le i^e enfant

- **Pseudo-classe** qui permet de désigner des éléments en fonction de leur position dans la fratrie :

E:nth-child(a + b)

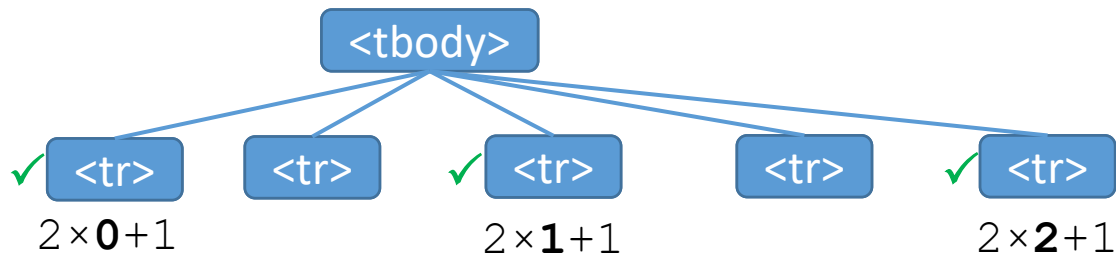
a est un nombre entier, positif ou négatif

Littéralement « n ». n est un entier variant de 0 à « $+\infty$ »

b est un nombre entier, positif ou négatif

désigne tous les éléments $\langle E \rangle$ dont la place dans leur fratrie vérifie la « formule » « $a+b$ »

- Exemple pour **tr:nth-child($2n+1$)**



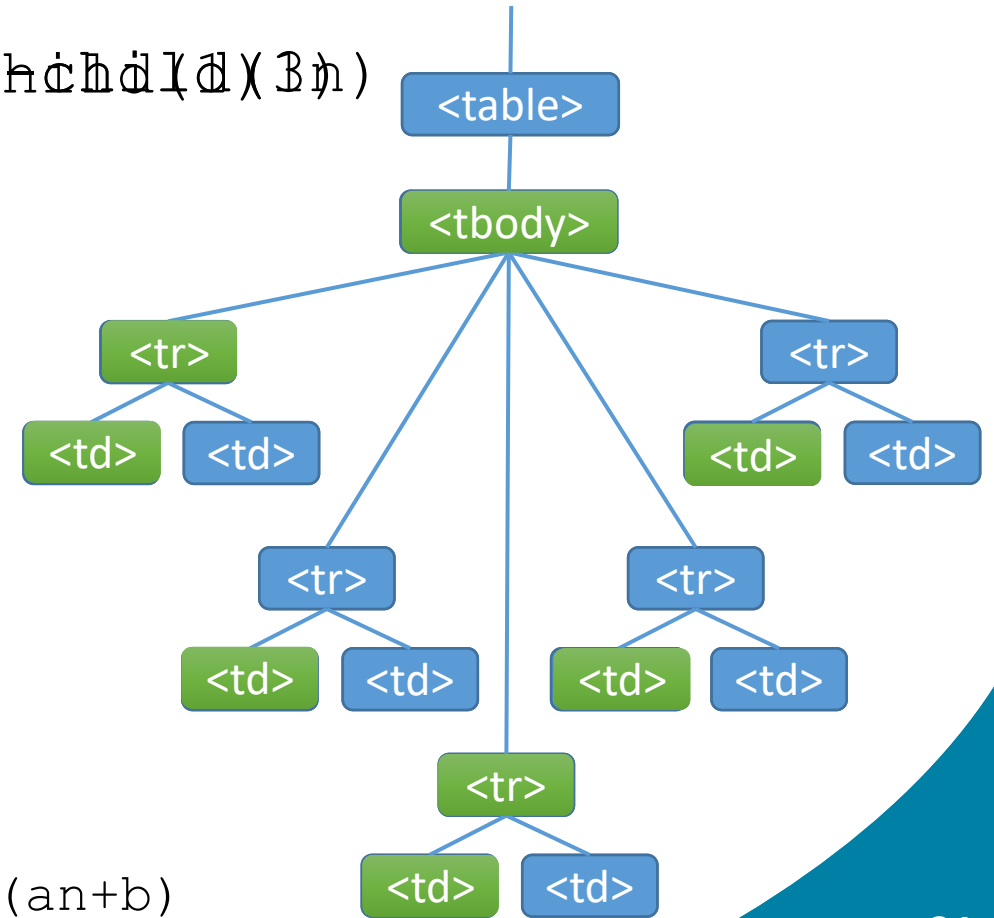
Désigne donc la 1^{re} ligne, la 3^e ligne et la 5^e ligne

ligne 1	valeur 1
ligne 2	valeur 2
ligne 3	valeur 3
ligne 4	valeur 4
ligne 5	valeur 5

:nth-child : des formes particulières

Exemple : `:nth-child(1)` (3n)

- Sans « an + » :
 - `:nth-child(1)` désigne le 1^{er} enfant ;
 - `:nth-child(4)` désigne le 4^e enfant ;
 - etc.
- Sans « + b » :
 - `:nth-child(3n)` désigne un enfant sur trois, à partir du 3^e : le 3^e, le 6^e, le 9^e, etc.
- Mot clef `even` désigne les éléments « pairs » (équivalent de `2n`)
- Mot clef `odd` désigne les éléments « impairs » (équivalent à `2n+1`)
- `:nth-child(an+b)` est équivalent à `*:nth-child(an+b)`
 - `:nth-child(1)` désigne donc tous les premiers enfants



Retour sur l'exemple du championnat (1/2)

- Désigner le 4^e : application directe
 - `tr:nth-child(4) { background-color: orange; }`

[Exemple du championnat en ligne](#)

- Les 3 premiers :
 - $an+b$ vrai pour 1, 2 et 3
 - a peut être négatif
 - $-n+3$ pour n commençant à 0 prendra les valeurs 3 ($n=0$); 2 ($n=1$) et 1 ($n=2$)
 - La position dans la fratrie est forcément ≥ 1
 - `tr:nth-child(-n+3) {background-color: green; }`

Retour sur l'exemple du championnat (2/2) : :nth-last-child

- Pour les 3 derniers, c'est comme pour les 3 premiers...
- ... mais en partant de la fin
- :nth-last-child
 - Fonctionne comme :nth-child
 - Mais on compte la fratrie à partir du dernier
 - :nth-last-child(1) : c'est le dernier élément
- Pour l'exemple : `tr:nth-last-child(-n+3) { background-color: red; }`

Pseudo classes

- Place dans l'arbre du document
 - **:first-child**
 - **:last-child**
 - **:first-of-type**
 - **:last-of-type**
 - **:only-child**
 - **:only-of-type**

`tr > td:last-of-type` : le dernier td d'une ligne (il peut y avoir des th après)

Pseudo classes

- Balise vide
:empty
pour les balises vides (<p></p>)
- Négation
:not(s)
un élément E qui ne vérifie pas un sélecteur simple s
Un sélecteur simple est un sélecteur de type (balise html) , sélecteur *, sélecteur sur les attributs, sélecteur de classe (.), sélecteur #, ou une pseudo-classe.

Pseudo éléments

- Précédé de ::
 - Pour différencier les pseudo classes des pseudo éléments
 - Préalablement de « : »
 - Compatibilité (les deux) pour :first-line, :first-letter, :before et :after
- Un seul pseudo élément par sélecteur
 - À la fin

Pseudo éléments

- **::first-line**
 - première ligne du texte
 - dépend de la taille de la fenêtre
- **::first-letter** :: première lettre du texte
- **::before** et **::after**
 - pour insérer un contenu généré avant ou après
 - `h1::before {content: counter(chapno, upper-roman) ". "}`
Ajoute une numérotation (automatique) en chiffre romain : « I. », « II. », etc.
 - Fonctionne avec content <http://www.w3.org/TR/css3-content/>
 - On peut utiliser des images de fond etc. (s'il y a un content)
- + d'autres pour les formulaires

Combinators

- Descendant combinator

E F { ... } /* désigne tous les F inclus, directement ou pas, dans un E
Notation E >> F dans le level 4 */

- Child combinators

E > F { ... } /* désigne tous les F inclus directement dans un E */

- Adjacent sibling combinator

E + F { ... } /* Un élément F dont le frère qui le précède est E */

- General sibling combinator

E ~ F { ... } /* Un élément F précédé par un élément E -> E et F ont le même parent */

Priorité pour déterminer quelle règle s'applique en cas de conflit

`<p class="toto">un paragraphe... de quelle couleur sera le texte ? </p>`

```
p { color: green; }  
*.toto { color: red; }
```

Assignation, cascade et héritage

- Le navigateur analyse le document (arbre d'éléments HTML)
- Le navigateur calcule le style pour chaque élément
 - Pour chaque propriété (de style) possible
 - En partant de la racine de l'arbre (html, puis body, ...)
 - Calcul en plusieurs étapes en se basant sur les valeurs spécifiées directement, celles qu'il faut calculer et celles qu'il faut ajuster.
- On parle de cascade car :
 - Les styles proviennent de plusieurs sources
 - Un élément peut être désigné par plusieurs sélecteurs
 - Certaines propriétés héritent des valeurs des éléments qui les contiennent (éléments parent dans l'arbre)

Valeurs spécifiées

- Les navigateurs doivent d'abord appliquer une valeur spécifiée à une propriété selon le mécanisme suivant (dans l'ordre de priorité) :
 1. Si la cascade donne une valeur, utiliser celle-ci ;
 2. Sinon, si la propriété est héritée, utiliser la valeur de l'élément parent ;
 3. Sinon, utiliser la valeur initiale de la propriété. Cette valeur initiale est définie pour chaque propriété par le navigateur.
- La racine de l'arborescence du document n'ayant pas de parent, et ne pouvant donc pas hériter de valeurs, c'est la valeur initiale qui est utilisée si nécessaire.

Cascade : identifier les styles à appliquer

- Trouver toutes les déclarations
 - Pour chaque propriété (selon média)
 - Pour chaque élément (sélecteurs)
- Trier selon l'origine
 - auteur > utilisateur > navigateur
- Trier selon spécificité des sélecteurs
 - les plus spécifiques prévalent
 - Les pseudo-éléments et les pseudo-classes sont considérés comme éléments et classes
- Trier selon ordre d'apparition
 - si deux règles ont les mêmes poids, origines et spécificités, c'est la dernière survenue qui l'emporte.
 - Les règles issues de feuilles de style importées sont considérées comme étant survenues avant chacune de celles de la feuille de style elle-même.

Spécificité d'un sélecteur

- La spécificité d'un sélecteur est déterminée comme suit :
 - dans le sélecteur, compter le nombre d'attributs Id (= a) ;
 - puis celui des autres attributs, classes et des pseudo-classes (= b) ;
 - et ensuite le nombre de « type selector » et de « pseudo element » (= c) ;
 - ignorer le sélecteur universel .
- Assembler les trois nombres a-b-c (dans un système de nombre avec une base étendue) pour obtenir la spécificité.
 - * {} /* a=0 b=0 c=0 -> spécificité = 0 */
 - LI {} /* a=0 b=0 c=1 -> spécificité = 1 */
 - UL LI {} /* a=0 b=0 c=2 -> spécificité = 2 */
 - UL LI+LI {} /* a=0 b=0 c=3 -> spécificité = 3 */
 - H1 + *[REL=up]{} /* a=0 b=1 c=1 -> spécificité = 1-1 */
 - UL OL LI.red {} /* a=0 b=1 c=3 -> spécificité = 1-3 */
 - LI.red[title] {} /* a=0 b=2 c=1 -> spécificité = 2-1 */
 - #x34y {} /* a=1 b=0 c=0 -> spécificité = 1-0-0 */
- Pour HTML, les valeurs de l'attribut "style" sont des règles de feuille de style.
 - Ces règles n'ont pas de sélecteurs
 - mais pour la cascade, elles sont plus prioritaire qu'un sélecteur d'ID (spécificité : $\alpha = 1$ a=0, b=0, c=0).
 - dans l'optique du « tri selon ordre d'apparition » (ordre d'apparition), elles surviennent après toutes les autres règles.

SNT Illustration

<body>

Titre

<p>Un paragraphe</p>

Un autre

</body>

$$* \{$$

```
color : red;
```

}

{

```
color: green;
```

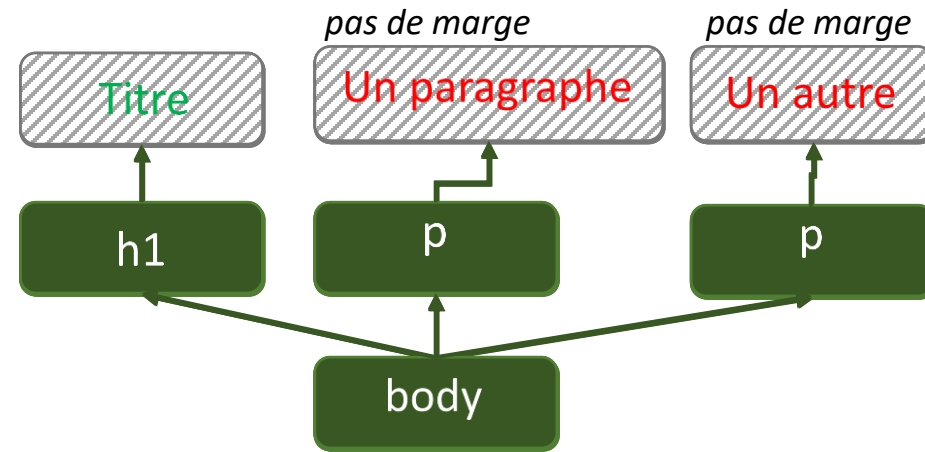
}

 $p \{$

```
margin: 0;
```

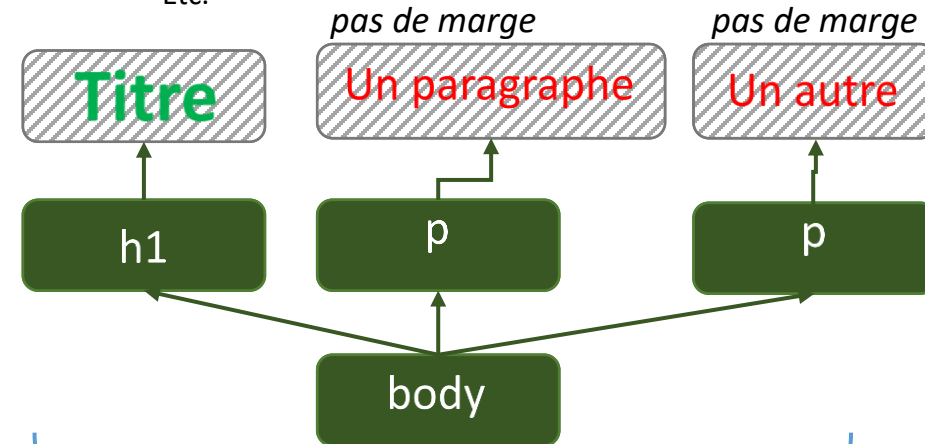
}

Valeurs par défaut



Apparence complétée par les valeurs par défaut :

- Polices de caractère (taille, épaisseur de trait, etc.)
- Marges pour h1 (car non spécifiées)
- ~~— Couleurs (car toutes spécifiées)~~
- Etc.



Titre

Un paragraphe
Un autre

<table> - HTML - CSS

Pour faire des tableaux

Pour les mettre en forme

Exemple de table

- <http://dabblet.com/gist/8410318c07bfa4084b90>
- `<table>` contient, dans l'ordre,
 - `<caption>` (optionnel)
 - `<colgroup>` (optionnel)
 - `<thead>` (optionnel)
 - `<tfoot>` (optionnel)
 - `<tbody>` (1 ou plusieurs)
 - Exception :
 - `<tbody>` peut être optionnel si les balises précédentes sont bien fermées et que le corps du tableau commence par un `<tr>`
 - `<tr>` peut donc être directement un enfant de `<table>`
 - Dans ce cas, `<tfoot>` peut venir en dernier (mais un seul `<tfoot>` en tout)
- Le contenu est fait par ligne `<tr>` (dans `<thead>`, `<tfoot>` ou `<tbody>`)
 - Nombre de lignes = nombre de `<tr>`
 - Attention aux attributs `rowspan` de `<td>` ou `<th>`
- Les cases par `<td>` ou `<th>`
 - Nombre de colonnes = nombre total de `<td>` et `<th>` par ligne
 - Attention aux attributs `colspan` de `<td>` ou `<th>`

Balise <table>

- Un seul attribut, à usage limité : border
border="" → il ne faut pas dessiner le contour
border="1" → il faut dessiner le contour
 - Le quadrillage se dessine en css par la propriété « border » sur les éléments <table> (contours extérieurs), <th> et <td> (quadrillage interne)
- En CSS, il y a 4 propriétés particulières à table
 - border-collapse : separate (les cellules ne sont pas collées) ou collapse soit inherit soit initial
 - border-spacing : pour préciser l'espacement (une longueur) entre les cellules (si border-collapse: separate) soit inherit soit initial
 - une ou deux longueurs (pour l'espacement horizontal et/ou vertical)
 - empty-cells : pour dessiner (show) ou non (hide) les cellules vides (si border-collapse: separate) soit inherit soit initial
 - table-layout : choix de l'algo pour dessiner le tableau, soit auto (par défaut, attente du chargement pour calculer la largeur en fonction du contenu) soit fixed (les largeurs des colonnes sont fixées par width sur table et/ou col/colgroup) soit inherit soit initial

Balise <caption>

- Précise la légende du tableau
- Contient du Flow exception faite d'une autre <table> (directement ou indirectement)
- CSS, une propriété particulière :
 - caption-side : top | bottom | left | right | inherit

Balises <colgroup> et <col>

- Regroupement de colonnes
 - Utilise pour dire lesquelles vont ensemble
 - Concrètement : utilisable par le css
 - Ne contient pas les éléments...
- <colgroup>
 - Contient des <col /> ou Utilise l'attribut span (nombre de « colonnes »)
- <col>
 - Balise vide (<col />)
 - Désigne une colonne (en fonction de sa place...)
 - attribut : span (nombre)
- CSS limité
 - background, border, visibility, width
 - visibility : visible | hidden (cacher mais prend de la place) | collapse (cacher mais ne prend pas de place)

Parties d'un tableau

- `<thead>` (titre sur lignes « supérieures »), `<tfoot>` (bas du tableau) et `<tbody>` (bloc de données)
- Contiennent des `<tr>` (0 ou +) ou du `<script>`

Ligne d'un tableau

- `<tr>`
- Contient des `<td>`, `<th>` (0 ou plus) et/ou du `<script>`
- En principe, possible directement sous `<table>` sous certaines conditions, mais <http://www.w3.org/TR/html5/syntax.html#element-restrictions>
➔ les navigateurs ajoutent automatiquement un `tbody` s'il n'y en a pas...

Cases (ou cellules) d'un tableau

- `<th>` : table header
 - Contient du Flow excepté du Sectioning Content, `<header>`, `<footer>` et du Heading Content
- `<td>` : table data
 - Contient du Flow
- Attributs communs
 - `rowspan` : pour indiquer sur combien de ligne s'étend la cellule
 - `colspan` : pour indiquer sur combien de colonne s'étend la cellule
 - Attention : à bien compter les td/th sur les autres lignes / colonnes...
 - Attention : pas toujours visible sans CSS...
 - <http://dabblet.com/gist/b40a4864acb7a6aaae64>
 - headers
- Attributs propres à `<th>`
 - `scope` : row | col | rowgroup (= tbody, thead ou tfoot) | colgroup | auto
 - Pour dire à quoi s'applique le titre
 - Auto : décidé par le navigateur
 - `abbr` : pour fournir une description (en principe plus courte) du titre

Mise en forme de table (1)

- <http://dabblet.com/gist/718beb16ceb98e6ecae2>
- Quadrillage sur les cellules
 - Conflit sur l'arrondi
- Ombre arrondie même s'il n'y a pas de bord
- Mettre en valeur une ligne sur 2 dans les tbody
- Parfois,
 - « border-collapse: collapse; » est remplacé par
« border-collapse: separate;
border-spacing: 0; »
 - Pour l'arrondi de cellules il faut que border-collapse soit à separate
 - Combiner caption-side avec une margin ou un padding (pas de chevauchement sur l'ombre)
 - Effets différents selon navigateur
 - Voir le côté droit de la cellule vide

Mise en forme de table (2)

- Pour désigner la cellule en haut à gauche
 - S'il y un thead et pas de script en 1er enfant
thead tr:first-of-type *:nth-child(1)
- Pour désigner la cellule en haut à droite
 - thead tr:first-of-type *:nth-last-child(1)
- Pour désigner la cellule en bas à gauche
 - tbody tr:last-of-type *:nth-child(1)
- Pour désigner la cellule en bas à droite
 - tbody tr:last-of-type *:nth-last-child (1)