

Cascading Style Sheet CSS

felipe@ift3225 Hiver 2020

Faits

- CSS1 en 1996 (W3C)
- CSS2 en 1998 (pas adoptée)
- CSS2.1 (2004-2011)
- CSS3 (depuis 2009)
 - découpé en modules qui évoluent indépendamment

HTML = contenu

CSS = présentation

Séparer les deux:

- sauve de la bande passante
- rend les pages plus robustes
- facilite la réutilisation

css3-background	CSS Backgrounds and Borders Module Level 3 ²	Candidate Rec.	Oct 2017
css3-box	CSS basic box model ²	Working Draft	Jul 2018
css-cascade-3	CSS Cascading and Inheritance Level 3 ²	Candidate Rec.	May 2016
css3-color	CSS Color Module Level 3 ²	Recommendation	Jun 2018
css3-content	CSS3 Generated and Replaced Content Module ²	Working Draft	Jun 2016
css-fonts-3	CSS Fonts Module Level 3 ²	Recommendation	Sep 2018
css3-gcpn	CSS Generated Content for Paged Media Module ²	Working Draft	May 2014
css3-layout	CSS Template Layout Module ²	Note	Mar 2016
css3-mediaqueries	Media Queries ²	Recommendation	Jun 2012
mediaqueries-4	Media Queries Level 4 ²	Candidate Rec.	Sep 2017
css3-multicol	Multi-column Layout Module Level 1 ²	Working Draft	May 2018
css3-page	CSS Paged Media Module Level 3 ²	Working Draft	Mar 2013
selectors-3	Selectors Level 3 ²	Recommendation	Nov 2018
selectors-4	Selectors Level 4 ²	Working Draft	Feb 2018
css3-ui	CSS Basic User Interface Module Level 3 (CSS3 UI) ²	Recommendation	Jun 2018

Jargon

- **Propriétés:** nom spécifique représentant un aspect `font-weight`
- **Sélecteur:** désigne des éléments du HTML `p ul`
- **Règles:** paires de propriétés / valeurs `color: red`

Commentaires en CSS: `/* ... */`

CSS et HTML

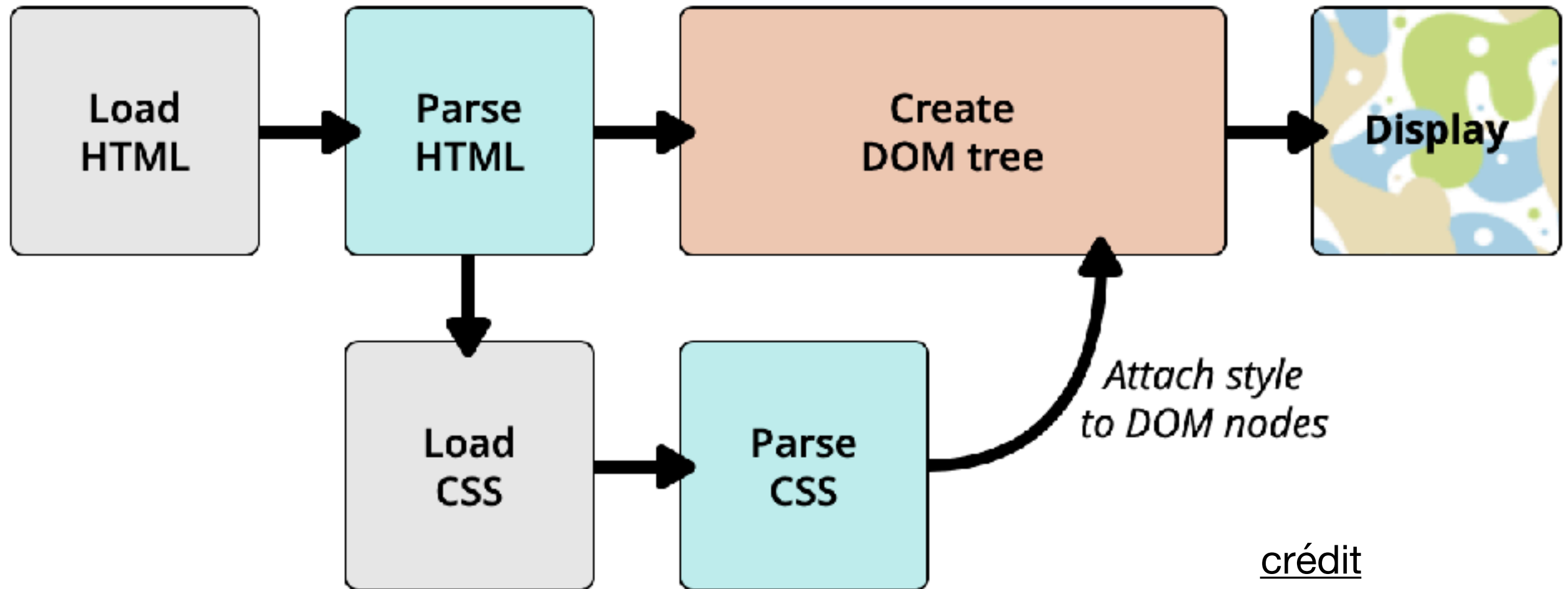
- Avant CSS: `<h1> Chapter 1. </h1>`
- En ligne: `<h1 style="color: red;"> Chapter 1. </h1>`
- Style interne

```
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
```

- Externe

```
<head>
<link href="path/to/file.css" rel="stylesheet" type="text/css">
</head>
```

Principe



CSS est tolérant

```
1 | <p> I want this text to be large, bold and blue.</p>
```

```
1 | p {  
2 |   font-weight: bold;  
3 |   colour: blue; /* incorrect spelling of the color property */  
4 |   font-size: 200%;  
5 | }
```

I want this text to be large, bold and blue.

Universal selector

Selects all elements. Optionally, it may be restricted to a specific namespace or to all namespaces.

Syntax: `*` `ns` | `*` `*` | `*`

Example: `*` will match all the elements of the document.

```
@namespace url(http://www.w3.org/1999/xhtml);
@namespace svg url(http://www.w3.org/2000/svg);
```

```
/* This matches all XHTML (default namespace) <a> elements */
a {}
```

```
/* This matches all SVG <a> elements */
svgla {}
```

```
/* This matches both XHTML and SVG <a> elements */
*|a {}
```

Type selector

Selects all elements that have the given node name.

Syntax: `elementname`

Example: `input` will match any `<input>` element.

Class selector

Selects all elements that have the given `class` attribute.

Syntax: `.classname`

Example: `.index` will match any element that has a class of "index".

ID selector

Selects an element based on the value of its `id` attribute. There should be only one element with a given ID in a document.

Syntax: `#idname`

Example: `#toc` will match the element that has the ID "toc".

Attribute selector

Selects all elements that have the given attribute.

Syntax: `[attr]` `[attr=value]` `[attr~=value]` `[attr|=value]` `[attr^=value]`
`[attr$=value]` `[attr*=value]`

Example: `[autoplay]` will match all elements that have the `autoplay` attribute set (to any value).

Sélecteurs d'attributs

[att]

Represents an element with the `att` attribute, whatever the value of the attribute.

W3C

[att=val]

Represents an element with the `att` attribute whose value is exactly "val".

[att~=val]

Represents an element with the `att` attribute whose value is a [whitespace](#)-separated list of words, one of which is exactly "val". If "val" contains whitespace, it will never represent anything (since the words are *separated* by spaces). Also if "val" is the empty string, it will never represent anything.

[att|=val]

Represents an element with the `att` attribute, its value either being exactly "val" or beginning with "val" immediately followed by "-" (U+002D). This is primarily intended to allow language subcode matches (e.g., the `hreflang` attribute on the `a` element in HTML) as described in BCP 47 ([\[BCP47\]](#)) or its successor. For `lang` (or `xml:lang`) language subcode matching, please see [the `:lang` pseudo-class](#).

Sélecteurs d'attributs

W3C

[att^=val]

Represents an element with the `att` attribute whose value begins with the prefix "val". If "val" is the empty string then the selector does not represent anything.

[att\$=val]

Represents an element with the `att` attribute whose value ends with the suffix "val". If "val" is the empty string then the selector does not represent anything.

[att*=val]

Represents an element with the `att` attribute whose value contains at least one instance of the substring "val". If "val" is the empty string then the selector does not represent anything.

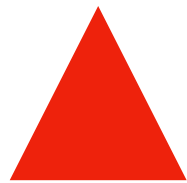
Grouping selectors

Selector list

The `,` is a grouping method, it selects all the matching nodes.

Syntax: `A, B`

Example: `div, span` will match both `` and `<div>` elements.



si une erreur se cache dans un sélecteur regroupé,
alors la règle au complet est ignorée

```
h1, h2.foo, h3 { font-family: sans-serif }
```

Descendant combinator

The (space) combinator selects nodes that are descendants of the first element.

Syntax: `A B`

Example: `div span` will match all `` elements that are inside a `<div>` element.

Child combinator

The `>` combinator selects nodes that are direct children of the first element.

Syntax: `A > B`

Example: `ul > li` will match all `` elements that are nested directly inside a `` element.

General sibling combinator

The `~` combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent.

Syntax: `A ~ B`

Example: `p ~ span` will match all `` elements that follow a `<p>`, immediately or not.

Adjacent sibling combinator

The `+` combinator selects adjacent siblings. This means that the second element directly follows the first, and both share the same parent.

Syntax: `A + B`

Example: `h2 + p` will match all `<p>` elements that directly follow an `<h2>`.

Column combinator

The `||` combinator selects nodes which belong to a column.

Syntax: `A || B`

Example: `col || td` will match all `<td>` elements that belong to the scope of the `<col>`.

Pseudo

Pseudo classes

The `:` pseudo allow the selection of elements based on state information that is not contained in the document tree.

Example: `a:visited` will match all `<a>` elements that have been visited by the user.

Pseudo elements

The `::` pseudo represent entities that are not included in HTML.

Example: `p::first-line` will match the first line of all `<p>` elements.

Pseudo sélecteurs

W3C

- The `:link` pseudo-class applies to links that have not yet been visited.
- The `:visited` pseudo-class applies once the link has been visited by the user. `a.external:visited`
- The `:hover` pseudo-class applies while the user designates an element with a pointing device, but does not necessarily activate it.
- The `:active` pseudo-class applies while an element is being activated by the user.
- The `:focus` pseudo-class applies while an element has the focus
- The `:enabled` pseudo-class represents user interface elements that are in an enabled state; such elements have a corresponding disabled state.
- Conversely, the `:disabled` pseudo-class represents user interface elements that are in a disabled state; such elements have a corresponding enabled state.
- `:checked`
- `:root`
- `:nth-child(an+b)` et beaucoup d'autres du même genre
- `:first-child`, `:last-child`, `:first-of-type`, `:only-child`, `:empty`
- `:not(X)`

`html|*:not(:link):not(:visited)`

Pseudo éléments

Propriétés non spécifiées par le marquage

::before

::after

::first-line

::first-letter

```
p::first-line { text-transform: uppercase }
```

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

Propriétés

B

Propriétés	Descriptifs	Familles
background	Propriétés de l'arrière-plan.	Background
background-attachment	Mode de défilement de l'image d'arrière-plan.	Background
background-color	Couleur de l'arrière-plan.	Background , Couleur
background-image	Image d'arrière-plan.	Background
background-position-x	Position horizontale de l'image d'arrière-plan.	Background
background-position-y	Position verticale de l'image d'arrière-plan.	Background
background-position	Position de l'image d'arrière-plan.	Background
background-repeat	Mode de répétition de l'image d'arrière-plan.	Background
behavior	Association d'un script à un élément HTML (Explorer seulement).	IE seulement
border	Caractéristiques des bordures.	Border
border-bottom	Caractéristiques de la bordure du bas.	Border
border-bottom-color	Couleur de la bordure du bas.	Border , Couleur
border-bottom-style	Style de la bordure du bas.	Border
border-bottom-width	Épaisseur de la bordure du bas.	Border
border-collapse	Mode d'encadrement des cellules adjacentes.	Table
border-color	Couleur des bordures.	Border , Couleur
border-left	Caractéristiques de la bordure de gauche.	Border
border-left-color	Couleur de la bordure de gauche.	Border , Couleur
border-left-style	Style de la bordure de gauche.	Border
border-left-width	Épaisseur de la bordure de gauche.	Border
border-right	Caractéristiques de la bordure de droite.	Border
border-right-color	Couleur de la bordure de droite.	Border , Couleur
border-right-style	Style de la bordure de droite.	Border
border-right-width	Épaisseur de la bordure de droite.	Border
border-spacing	Espacement entre deux traits de cellules adjacentes.	Table
border-style	Style des bordures.	Border
border-top	Caractéristiques de la bordure du haut.	Border
border-top-color	Couleur de la bordure du haut.	Border , Couleur
border-top-style	Style de la bordure du haut.	Border
border-top-width	Épaisseur de la bordure du haut.	Border
border-width	Épaisseur des bordures.	Border
bottom	Position basse d'un élément.	Position

C

Propriétés	Descriptifs	Familles
caption-side	Emplacement du titre des tableaux.	Table
clear	Gestion de l'habillage (en relation avec la propriété "float").	Position
clip	Définit la partie visible de l'élément.	Position
color	Couleur de l'avant plan (texte, par exemple).	Couleur , Texte
content	Ajoute du contenu dans le document (génération de contenu).	Génération de contenu
counter-increment	Définit le pas d'incrément d'un compteur (génération de contenu).	Génération de contenu
counter-reset	Fixe la valeur de départ d'un compteur (génération de contenu).	Génération de contenu
cue	Résumé des deux propriétés Cue-Before et Cue-After.	Son
cue-before	Séquence sonore à jouer avant la lecture d'un élément.	Son
cue-after	Séquence sonore à jouer après la lecture d'un élément.	Son
cursor	Forme du curseur pour la souris.	Divers

D

Propriétés	Descriptifs	Familles
direction	Sens d'écriture.	Texte
display	Façon d'afficher un élément.	Position

E

Propriétés	Descriptifs	Familles
elevation	Provenance verticale du son.	Son
empty-cells	Mode d'affichage des cellules vides.	Table

Fontes

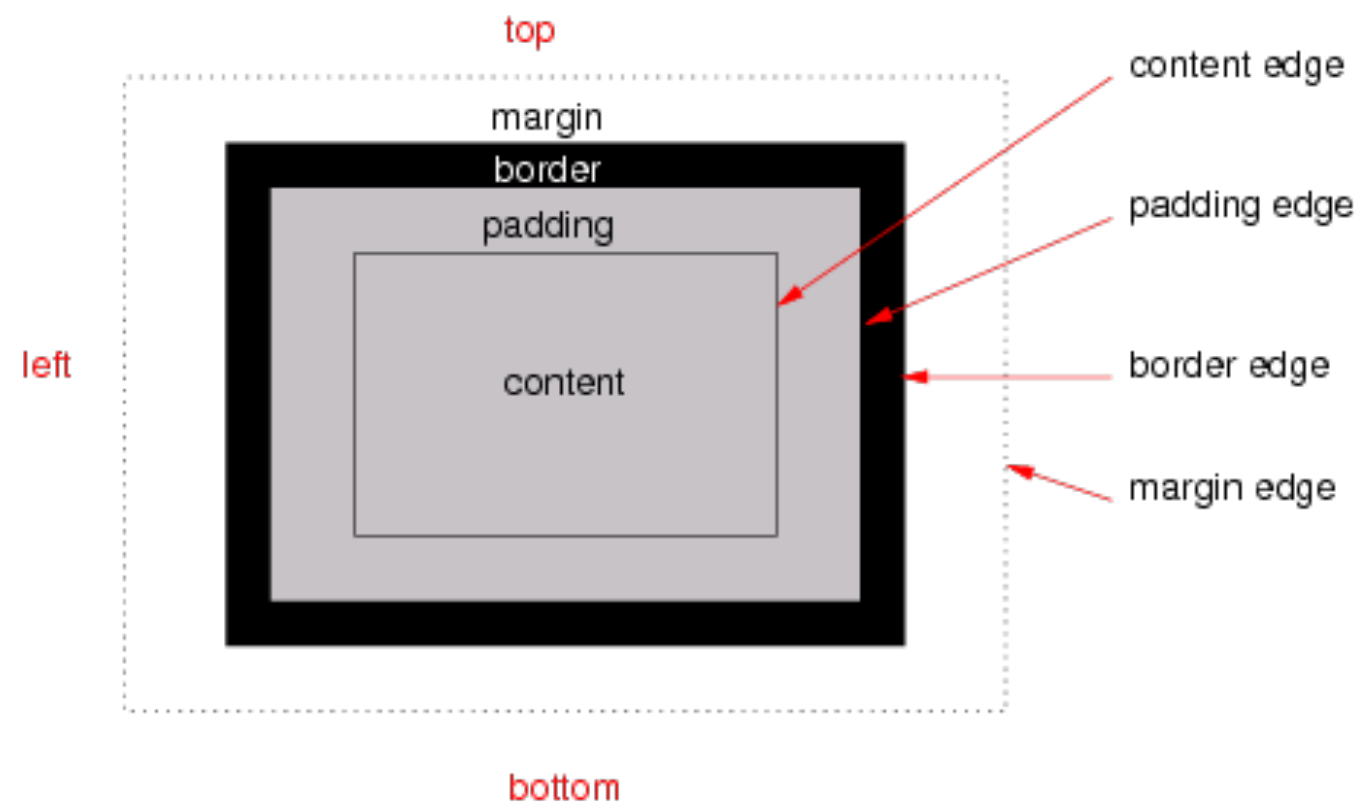
```
p {  
  font-family: "Times New Roman", Times, serif;  
}
```

souhaitée générique (défaut)

font-family	Example text
Georgia, serif	This is a heading This is a paragraph
"Palatino Linotype", "Book Antiqua", Palatino, serif	This is a heading This is a paragraph
"Times New Roman", Times, serif	This is a heading This is a paragraph

Box model

Chaque élément, string dans le document HTML est associé à une boîte qui permet son placement sur la fenêtre.



notez la date du document !

The following code demonstrates some possible **margin** declarations.

```
body { margin: 2em } /* all margins set to 2em */  
body { margin: 1em 2em } /* top & bottom = 1em, right & left = 2em */  
body { margin: 1em 2em 3em } /* top=1em, right=2em, bottom=3em, left=2em */
```

The last rule of the example above is equivalent to the example below:

```
body {  
  margin-top: 1em;  
  margin-right: 2em;  
  margin-bottom: 3em;  
  margin-left: 2em; /* copied from opposite side (right) */  
}
```

Les boites sont partout

The screenshot shows the CSS Tricks website layout. At the top, there is a navigation bar with the site logo, a search bar, and menu items for Articles, Forums, Videos, and Downloads. The main content area features an article titled "The CSS Box Model" by Chris Coyier, dated 6/1/2009. The article text explains that every element in web design is a rectangular box and includes a diagram of a box with labels for width, height, and padding. The right sidebar contains several advertisements, including one for FoxyCart, FreshBooks, and site5.

CSS TRICKS A WEB DESIGN COMMUNITY curated by [Chris Coyier](#)

Articles Forums Videos Downloads

Search...

Archives Search in: [Both](#) | [Articles](#) | [Forums](#)

The CSS Box Model

★ 6/1/2009 ★

by: CHRIS COYIER

At the risk of over-repeating myself: every element in web design is a rectangular box. This was my ah-ha moment that helped me really start to understand CSS-based web design and accomplish the layouts I wanted to accomplish. We've talked about the positioning of these boxes a bit, and about their behavior.

What we haven't talked about much is the box itself. How is the size of the box calculated exactly? Here is a diagram:

The diagram shows a yellow box with a dashed border. The text "The Box" is centered inside. Labels "WIDTH" and "HEIGHT" are placed at the bottom and right sides of the box, respectively. The word "padding" is written at the top of the box.

If you are a Firebug user, you might be used to the diagram like this, which does a nice job of showing you the numbers affecting any box on the page:

HE'S LOOKING PROFESSIONAL Online Invoicing for Freelancers

FRESHBOOKS online billing

elegant word press themes

Premium WordPress Themes That don't suck...

ARE MY SITES UP? DATA MONITORING

You Design. We Host. NO Problems. site5

Does your website? TickApps

Box-sizing



La propriété `box-sizing` a 3 valeurs:

`content-box`
`padding-box`
`Border-box`

Défaut

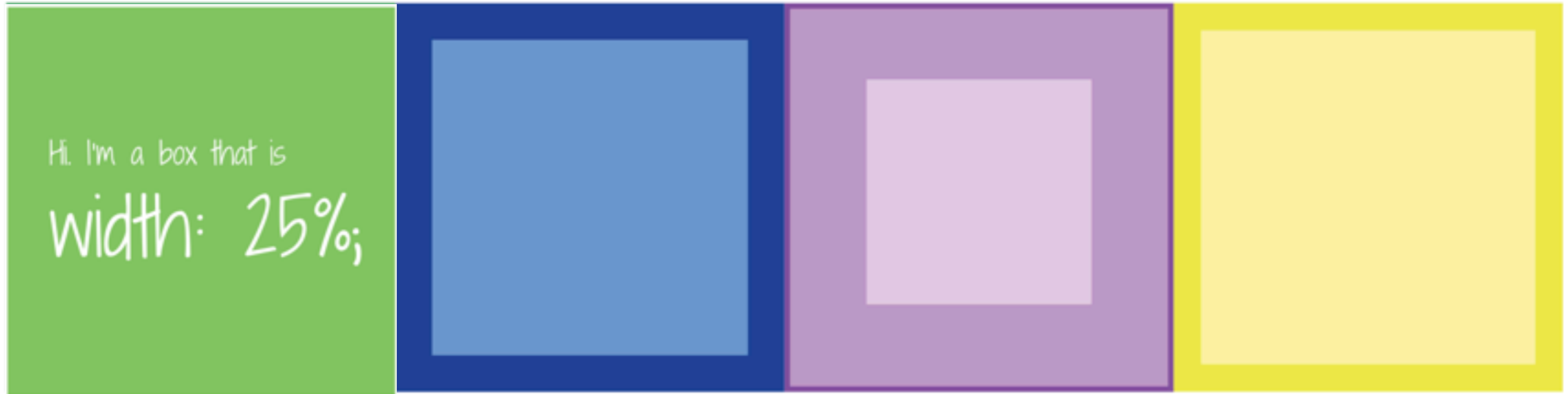
Pour des sites fluides

```
html {  
  box-sizing: border-box;  
}  
*, *:before, *:after {  
  box-sizing: inherit;  
}
```

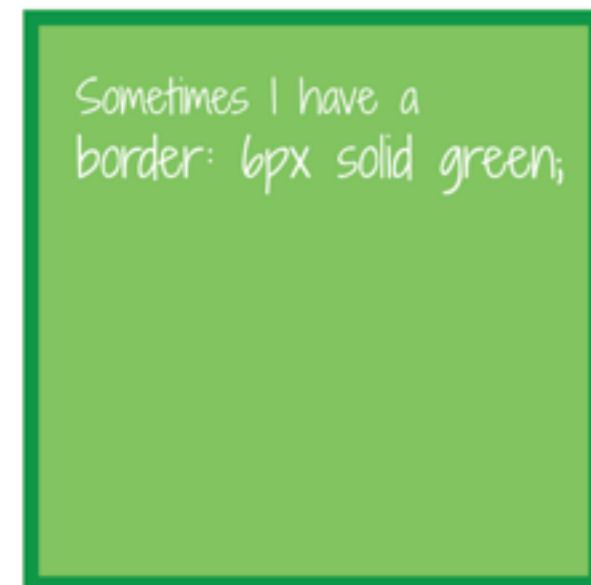
Reset

<https://css-tricks.com/box-sizing/>

box-sizing



Four of me can sit in a row no matter what border and padding we have.



With box-sizing: border-box;
my width is
still 25% (yay!)

Now my width is... uhm,
25% + 12px I guess?
I can tell you one thing,
four of me won't fit on a row.

Valeurs

- Partout où tout cela est permis

`<length>`, `<frequency>`, `<angle>`, `<time>`, `<percentage>`, `<number>`, or `<integer>`

```
.outer {  
  border: 5px solid black;  
}
```

```
.box {  
  padding: 10px;  
  width: calc(90% - 30px);  
  background-color: rebeccapurple;  
  color: white;  
}
```

```
.foo {  
  --widthA: 100px;  
  --widthB: calc(var(--widthA) / 2);  
  --widthC: calc(var(--widthB) / 2);  
  width: var(--widthC);  
}
```

La cascade

Les règles proviennent:

- de l'agent (navigateur),
- de l'auteur de la page (le plus souvent),
- de l'utilisateur (de la page)

Les règles s'appliquant à un élément sont triées:

	Origin	Importance
1	user agent	normal
2	user	normal
3	author	normal
4	animations	
5	author	!important
6	user	!important
7	user agent	!important
8	transitions	

moins important

plus important

La cascade

- À égalité, un calcul de spécificité détermine la règle gagnante selon leur sélecteurs:

1. Sélecteurs de type (h1) et pseudo-éléments (::before)
2. Sélecteurs de classe (.contenu), sélecteurs d'attributs ([name=~]) et pseudo-classes (:hover)
3. Sélecteurs ID (#monid)

<https://specifishity.com>

Le sélecteur universel, la négation (pas son contenu) ne comptent pas dans le calcul de la spécificité

- un exemple de conflit

```
#someElement p {  
  color: blue;  
}
```

le gagnant

```
<div id="someElement">  
  <p class="awesome">  
    halo  
  </p>  
</div>
```

```
p.awesome {  
  color: red;  
}
```



```
p.awesome {  
  color: red !important;  
}
```

mauvaise pratique

L'ordre compte également

À spécificité égale, l'ordre compte:

- L'ordre dans lequel les fichiers de style sont chargés
- L'ordre des règles dans un fichier

<https://css-tricks.com/precedence-css-order-css-matters/>

@media

```
@media print {  
  body { font-size: 10pt; }  
}
```

```
@media screen {  
  body { font-size: 13px; }  
}
```

```
@media screen, print {  
  body { line-height: 1.2; }  
}
```

```
@media only screen  
  and (min-width: 320px)  
  and (max-width: 480px)  
  and (resolution: 150dpi) {  
  body { line-height: 1.4; }  
}
```

screen
speech
print
all

@media

```
// Extra small devices (portrait phones, less than 576px)  
// No media query for `xs` since this is the default in Bootstrap
```

```
// Small devices (landscape phones, 576px and up)  
@media (min-width: 576px) { ... }
```

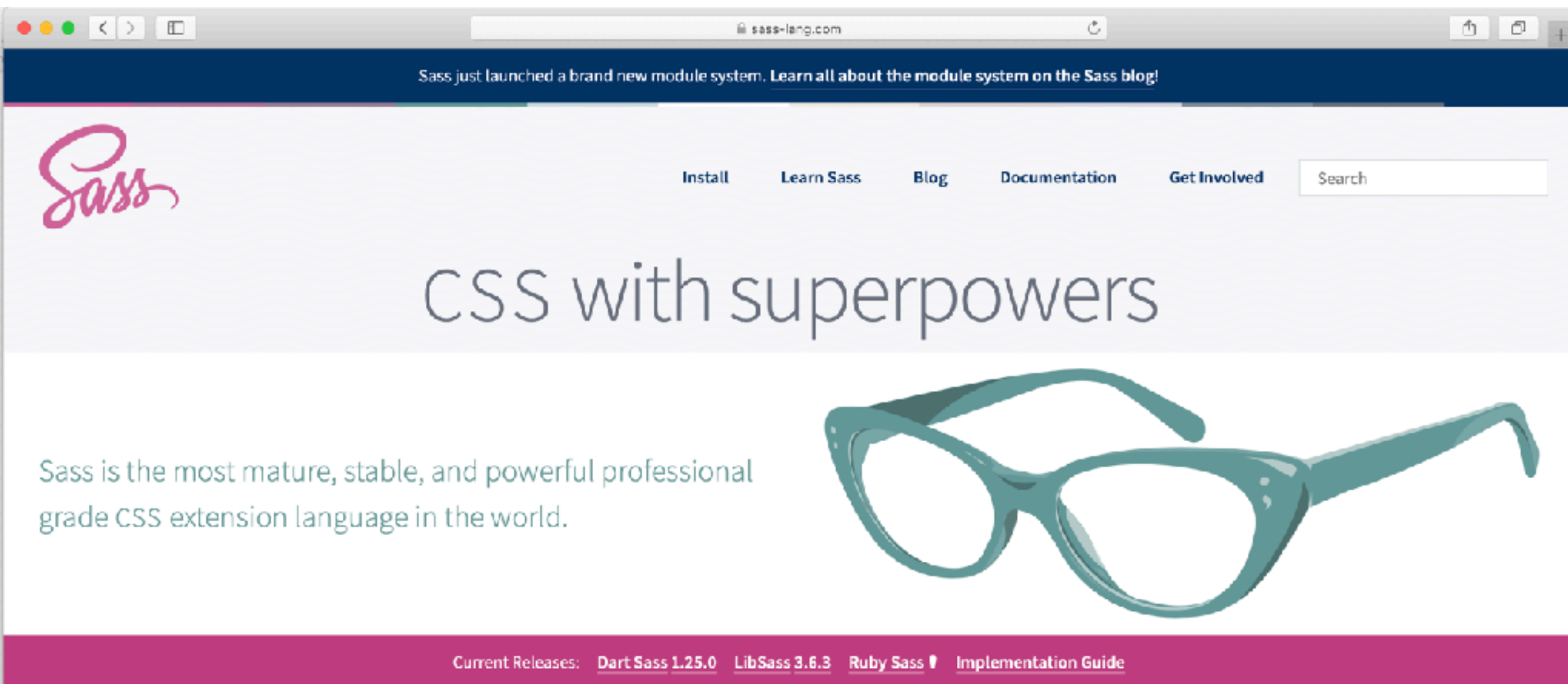
```
// Medium devices (tablets, 768px and up)  
@media (min-width: 768px) { ... }
```

```
// Large devices (desktops, 992px and up)  
@media (min-width: 992px) { ... }
```

```
// Extra large devices (large desktops, 1200px and up)  
@media (min-width: 1200px) { ... }
```

On parle souvent de **breakpoint**.

CSS est compliqué !



The screenshot shows the Sass website homepage. At the top, there's a navigation bar with the Sass logo on the left and links for 'Install', 'Learn Sass', 'Blog', 'Documentation', and 'Get Involved' on the right. A search bar is also present. Below the navigation, the main heading reads 'css with superpowers'. To the left of this heading, a paragraph states: 'Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.' To the right is a large illustration of a pair of teal-rimmed glasses. At the bottom of the page, a pink banner contains the text 'Current Releases: [Dart Sass 1.25.0](#) [LibSass 3.6.3](#) [Ruby Sass](#) [Implementation Guide](#)'.

CSS Compatible

Sass is completely compatible with all versions of CSS. We take this compatibility seriously, so that you can seamlessly use any available CSS libraries.

Feature Rich

Sass boasts more features and abilities than any other CSS extension language out there. The Sass Core Team has worked endlessly to not only keep up, but stay ahead.

Mature

Sass has been actively supported for about 13 years by its loving Core Team.

Sass

Variables

SCSS Sass

```
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

CSS

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

SCSS Sass

```
$font-stack: Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

SASS: Syntactically Awesome

Sass

Imbrication

SCSS Sass

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

CSS

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

SCSS Sass

```
nav  
  ul  
    margin: 0  
    padding: 0  
    list-style: none  
  
  li  
    display: inline-block
```

Sass

Opérateurs

SCSS **Sass**

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 600px / 960px * 100%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 300px / 960px * 100%;  
}
```

CSS

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 62.5%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 31.25%;  
}
```

Sass

Inbrication

SCSS **Sass**

```
// _base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

```
// styles.scss
@use 'base';

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

CSS

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}

.inverse {
  background-color: #333;
  color: white;
}
```

Sass

Mixins

SCSS **Sass**

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}  
.box { @include transform(rotate(30deg)); }
```

CSS

```
.box {  
  -webkit-transform: rotate(30deg);  
  -ms-transform: rotate(30deg);  
  transform: rotate(30deg);  
}
```

Définit ici une macro **transform** que l'on peut par la suite utiliser pour éviter la duplication de code css

Sass

Héritage

SCSS Sass ⇒ CSS

```
/* This CSS will print because %message-shared is extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

// This CSS won't print because %equal-heights is never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}

.message {
  @extend %message-shared;
}

.success {
  @extend %message-shared;
  border-color: green;
}

.error {
  @extend %message-shared;
  border-color: red;
}

.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

SCSS Sass ⇒ CSS

```
/* This CSS will print because %message-shared is extended. */
.message, .success, .error, .warning {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}

.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

Sass

Préprocesseur

Dans votre terminal:

```
sass --watch input.scss output.css
```

pour une transformation à la demande

```
sass --watch app/sass:public/stylesheets
```

transformation lors de changements dans les s(a|c)ss