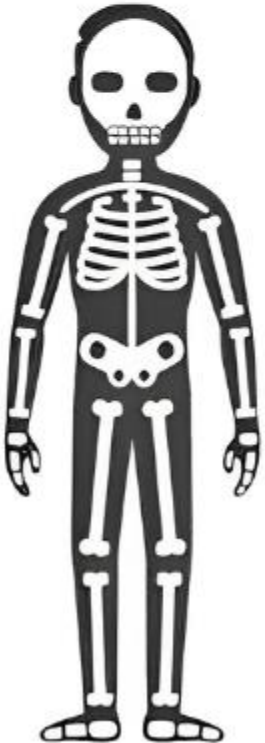


HTML



HTML the Skeleton



CSS



CSS the Skin



Cours CSS

BC DEV ITIS

Evry-Courcouronnes

GLODIE TSHIMINI

Consultant Formateur et Développeur depuis 2017

- Avant-hier, développeur chez **INRAE** à Paris
- Hier, **Tech Lead** dans une agence digitale à Saint Raphaël
- Aujourd'hui, **consultant indépendant** dans la formation et le développement d'applications Web
- Certifié Professional Scrum Developer (PSD I)
- student@tshimini.fr

Avant de commencer

Les ressources

- ▶ Cours, exercices et corrections disponibles en ligne depuis ce lien GitHub
- ▶ Les démonstrations de code et les corrections des exercices seront envoyées au fur à mesure de l'avancement du cours sur GitHub

Validation des acquis

- ▶ Exercices
- ▶ QCM/Quiz
- ▶ Devoirs maisons

Avant de commencer

Objectifs pédagogiques

- Comprendre la structure d'un document CSS
- Ajouter du style à une page Web
- Effectuer des mises en page simple d'une page Web

Programme

- Sélecteurs
- Propriétés
- Le boxing model
- Mise en page avec le Flexbox
- Mise en page avec le CSS Grid
- ▶ Durée de la formation : 3H30

Plan du cours

- I. Sélecteurs et propriétés
- II. Boxing Model
- III. Mise en page avec le CSS Flexbox
- IV. Mise en page avec le CSS Grid



Commençons par ce que vous connaissez
déjà :

Que connaissez-vous du CSS ?

I. Sélecteurs et propriétés

Définitions

- ▶ CSS pour Cascading Style Sheet
- ▶ Vient compléter le HTML (contenu brut qui sert de squelette d'une page web) pour ajouter du style (« décoration »)
- ▶ Par ajouter du style, on entend, l'ajout des couleurs, polices, faire la mise en page, animation, etc.

Comment utiliser le CSS avec le HTML

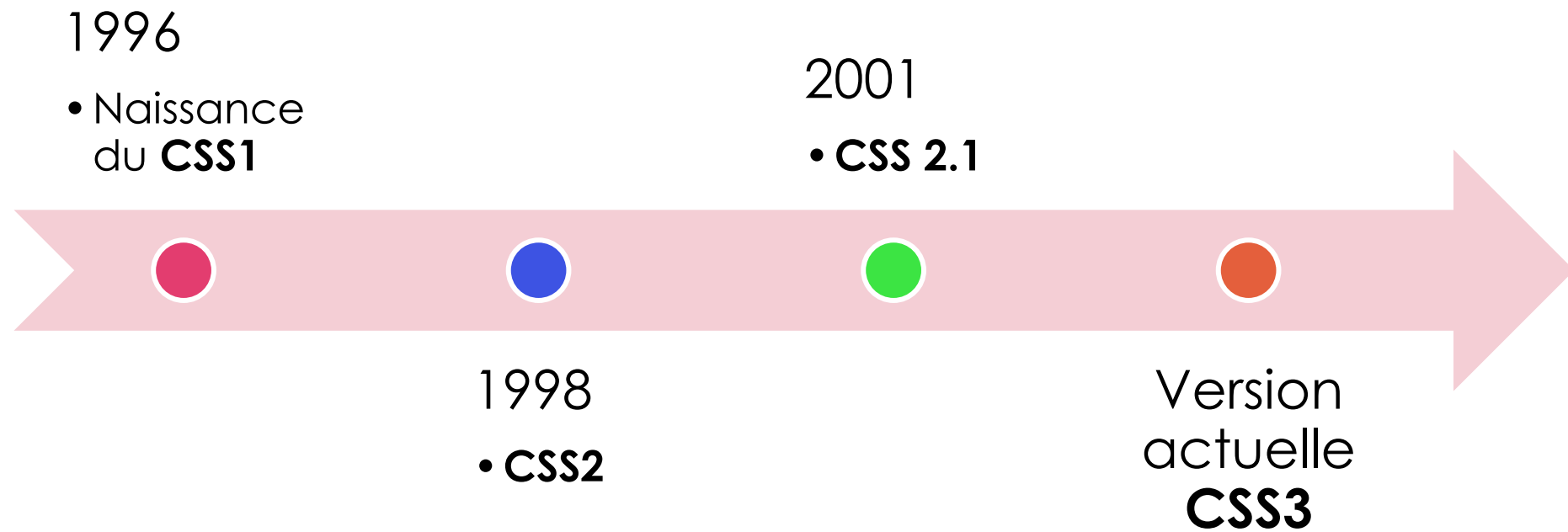
► 3 méthodes :

1. Dans un fichier externe main.css via une liaison au sein du HTML par l'intermédiaire de la balise enfant **link** du **head** :
`<head> <link rel="stylesheet" href="main.css"/>`
`</head>`
2. Ou directement dans le document HTML avec l'utilisation de la balise `<style></style>`
3. Ou directement au sein d'une balise HTML avec l'utilisation de **l'attribut style** .

► Nous utiliserons uniquement la première méthode

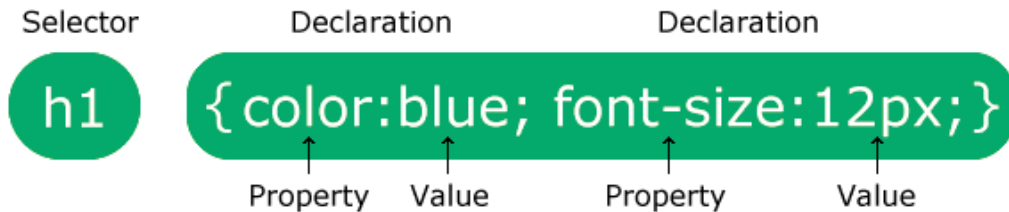
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="main.css">
  <title>Document</title>
  <style>
    p {
      color: orange;
    }
  </style>
</head>
<body>
  <h1 style="color: red; font-weight:bold;">Title</h1>
  <p>Paragraph</p>
</body>
</html>
```

Un peu d'histoire



Usage du CSS avec sa syntaxe

[Source image W3Schools](#)



1. Sélecteur : élément HTML ciblé
2. Propriété : le style souhaité
3. Valeur : la valeur associée à la propriété souhaitée











► */* Commentaires en CSS */*

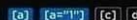
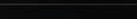

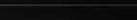

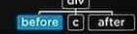



A vous de jouer

[Jeux CSS Dinner pour apprendre à utiliser les sélecteurs](#)

Synthèse sélecteurs

All css selectors			
CSS {selectors: cheat-sheet}			
Basic			
Name	CSS	Description	Results
Universal Selector	*	Select all elements	
Type Selector	div	Select elements of that type	
Class Selector	.c	Select elements with that class	
Id Selector	#i	Select elements with that id	
Combination			
Name	CSS	Description	Results
Descendant Selector	div a	Select elements that are descendants of the first element	
Direct Child Selector	div > a	Select elements that are direct children of the first element	
General Sibling Selector	div ~ a	Select elements that are siblings of the first element and come after the first element	
Adjacent Sibling Selector	div + a	Select elements that are siblings of the first element and come directly after the first element	
Or Selector	div, a	Select elements that match any selector in the list	
And Selector	div.c	Select elements that match all the selector combinations	

Attribute			
Name	CSS	Description	Results
Has Attribute	[a]	Select elements that have that attribute	
Exact Attribute	[a="1"]	Select elements that have that attribute with exactly that value	
Begins With Attribute	[a^="1"]	Select elements that have that attribute which start with that value	
Ends With Attribute	[a\$="1"]	Select elements that have that attribute which end with that value	
Substring Attribute	[a*="1"]	Select elements that have that attribute which contain that value anywhere	
Pseudo Element			
Name	CSS	Description	Results
Before Selector	div::before	Creates an empty element directly before the children of selected element	
After Selector	div::after	Creates an empty element directly after the children of selected element	
Pseudo Class State			
Name	CSS	Description	
Hover Selector	button:hover	Select elements that are hovered by the mouse	
Focus Selector	button:focus	Select elements that are focused	
Required Selector	input:required	Select inputs that are required	
Checked Selector	input:checked	Select checkboxes/radio buttons that are checked	
Disabled Selector	input:disabled	Select inputs that are disabled	

Pseudo-classes

- ▶ Ajouter un style spécifique en fonction de l'état dans lequel se trouve l'élément ciblé ou en tenant compte des autres éléments présents dans le document HTML
- [Documentation MDN pseudo-classes](#)
- [Documentation W3Schools sélecteurs](#)
- [Test sélecteurs W3Schools](#)

[Source image arredev](#)

Pseudo Class Position/Other			
Name	CSS	Description	Results
First Child Selector	<code>a:first-child</code>	Select elements that are the first child inside a container Select anchors that are the first child	<div>div</div> <div>a b a</div>
Last Child Selector	<code>a:last-child</code>	Select elements that are the last child inside a container Select anchors that are the last child	<div>div</div> <div>a b a</div>
Nth Child Selector	<code>a:nth-child(2n)</code>	Select elements that are the nth child inside a container based on the formula Select anchors that are even-numbered children	<div>div</div> <div>a a b a</div>
Nth Last Child Selector	<code>a:nth-last-child(3)</code>	Select elements that are the nth child inside a container based on the formula counting from the end Select anchors that are the third to last child	<div>div</div> <div>a b a</div>
Only Child Selector	<code>a:only-child</code>	Select elements that are the only child inside a container Select anchors that are the only child	<div>div</div> <div>a b a</div>
First Of Type Selector	<code>a:first-of-type</code>	Select elements that are the first of a type inside a container Select the first anchor in a container	<div>div</div> <div>b a b</div>
Last Of Type Selector	<code>a:last-of-type</code>	Select elements that are the last of a type inside a container Select the last anchor in a container	<div>div</div> <div>b a b</div>
Nth Of Type Selector	<code>a:nth-of-type(2n)</code>	Select elements that are the nth of a type inside a container based on the formula Select every second anchor	<div>div</div> <div>b a b a</div>
Nth Last Of Type Selector	<code>a:nth-last-of-type(2)</code>	Select elements that are the nth of a type inside a container based on the formula counting from the end Select the second to last anchor	<div>div</div> <div>b a b</div>
Only Of Type Selector	<code>a:only-of-type</code>	Select elements that are the only of a type inside a container Select anchors that are the only anchor in a container	<div>div</div> <div>b a b</div>
Not Selector	<code>a:not(.c)</code>	Select all elements that do not match the selector inside the not selector Select all anchors tags that do not have the c class	<div>div</div> <div>b a c a d</div>

Les propriétés et valeurs : sélecteur { propriété : valeur ;}

Couleurs : couleur
du texte , fond, etc.

`color : red;`

`background-
color : #808080;`

Police de caractère

`font-family:
Impact,
Verdana, "Arial
Black";`

Taille du texte

`font-size : 15px;`

Styles de police :
italique, gras, etc.

`font-style :
italic;`

`font-weight :
bold;`

Alignement du
texte

`text-align:
justify;`

Les propriétés et valeurs : sélecteur { propriété : valeur ;}

Marges extérieures : espace entre les éléments en haut, droite, gauche et bas

Marges intérieures : espace en haut, droite, bas et gauche au sein de l'élément

`margin : 5px ;`

`margin-top : 5px ;`

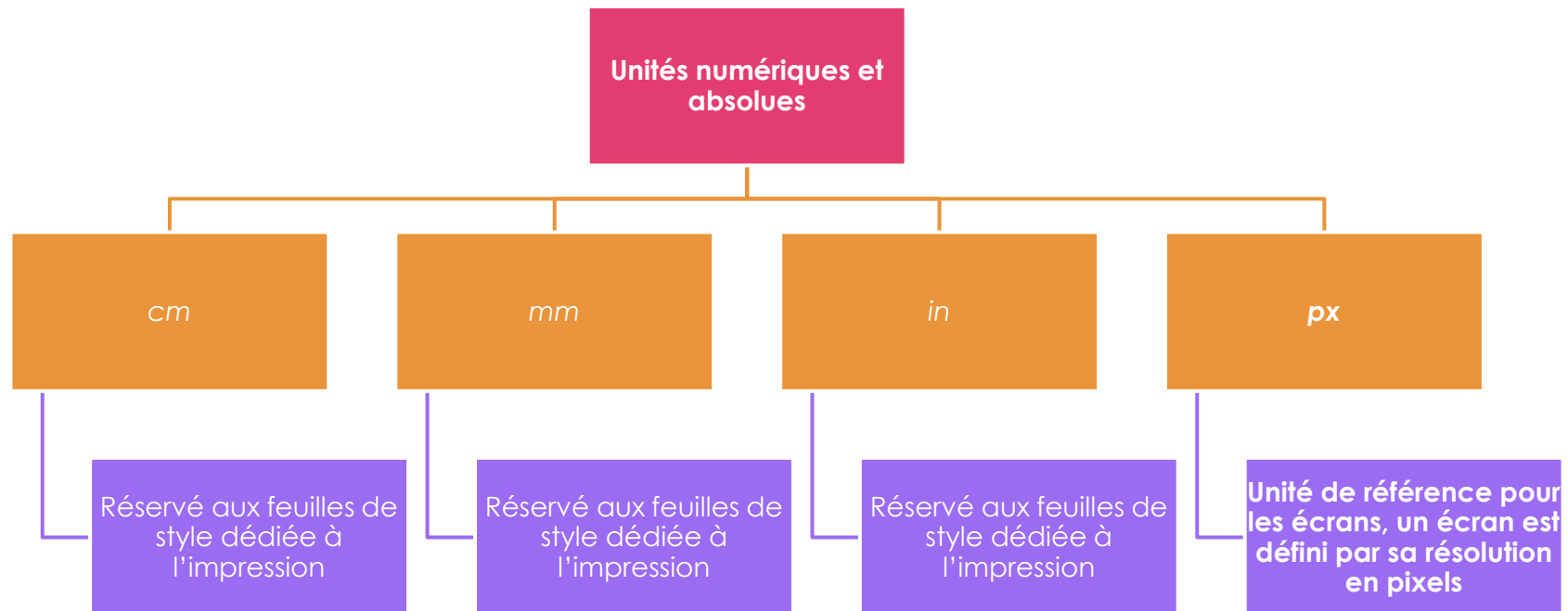
`margin-bottom : 30px ;`

`padding-left : 10px ;`

`padding : 2px ;`

`padding-right : 8px ;`

Les unités numériques et absolues



Les unités numériques et relatives

Unités numériques et relatives

%

Dimension est proportionnelle à la dimension du conteneur (parent) à utiliser principalement pour définir une largeur (*width*)

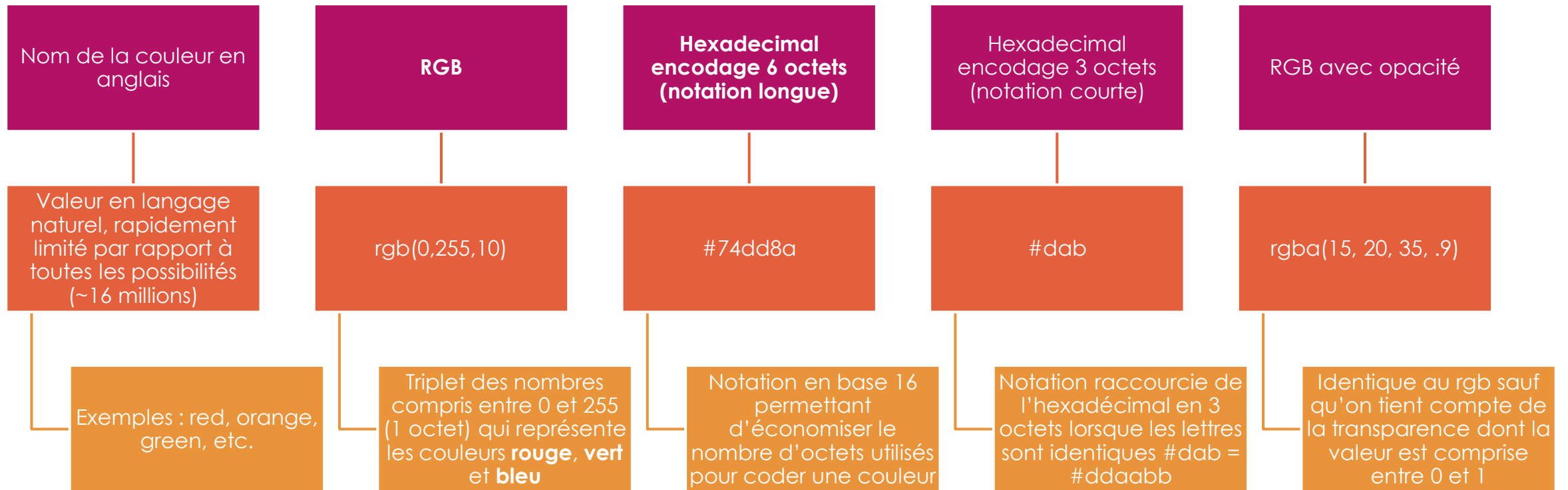
em

Relative par rapport à la taille de la police de l'élément ou de l'élément parent

rem

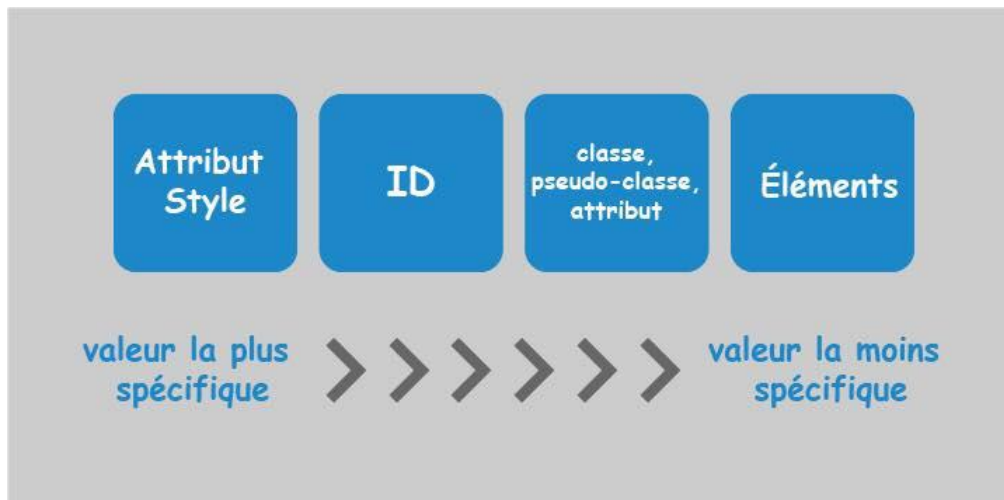
Relative par rapport à la taille de la police de l'élément racine html ou *body*

Les couleurs : les valeurs



Spécificités et priorités

[Source image gauche w3docs](#)



[Source image droite groupe Sii](#)

- Lorsque plusieurs règles (propriétés et valeurs) sont appliquées à un élément avec des sélecteurs différents qui ciblent ce même élément, le navigateur applique en premier les règles les plus spécifiques.

div.class1 > div.class2.class3.class4 .class5:hover			
0 Inline styles	0 IDs	6 Classes, attributes and pseudo-classes	2 Elements and pseudo-elements
#id1:hover			
0 Inline styles	1 IDs	1 Classes, attributes and pseudo-classes	0 Elements and pseudo-elements

II. Boxing Model

BOX MODEL : 3 grandes catégories d'élément

1. Block

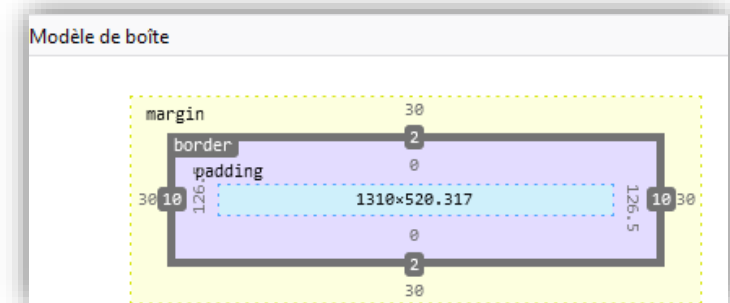
- Occupe toute la largeur disponible
- Élément suivant se positionne en bas (retour à la ligne)
- Propriétés **width** & **height** modifiables
- Exemples des éléments de type *block* : les titres **h1** à **h6**, les paragraphes **p**, **div**, **header**, **ul**, **ol**, **table**, etc.

2. Inline

- Occupe uniquement la place nécessaire qu'il lui faut
- Élément suivant se positionne à la suite
- Propriétés **width** et **height** non modifiables
- Exemples des éléments de type *inline* : **span**, **a**, etc.

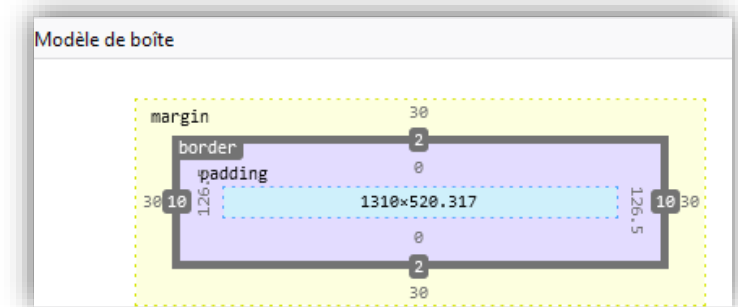
3. Inline-block

- Se comporte comme un *inline* avec les propriétés **width** & **height** modifiables
- Exemples : **img**, **input**, **textarea**



BOX SIZING : modèle de boîte

- Définit la façon dont la hauteur et la largeur totale d'un élément sont calculées à l'aide de la propriété **box-sizing**
- Valeur par défaut **content-box**
 - A la largeur (*width*) et à la hauteur (*height*) de base de l'élément, viennent s'ajouter les marges intérieures (*padding*), les marges extérieures (*margin*) et la bordure (*border*)
- Valeur **border-box**
 - Prend en compte le *padding*, *margin* et *border* dans le calcul du *width* et *height*
 - Valeurs du *width* et *height* incluent le *margin*, *padding* et *border*



III. Mise en page CSS Flexbox

A vous de jouer

[Jeux CSS Flexbox Froggy pour apprendre à faire une mise en page simple](#)

SYNTHÈSE FLEXBOX

- Boîtes flexibles
- Un **conteneur** (l'élément parent)
 - Rendu flexible grâce à la la déclaration **display : flex**
 - Les éléments vont se positionner de manière flexible à l'intérieur selon une direction
- 2 axes
 - **Axe principal** ou main axis
 - **Axe secondaire** ou cross axis ou axe transversal

Flexbox Cheat Sheet

@simonpaix

Parent properties

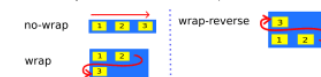
display: enables flex context for all direct children.

```
.container {  
  display: flex; // or inline-flex  
}
```

flex-direction: sets the main-axis.



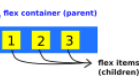
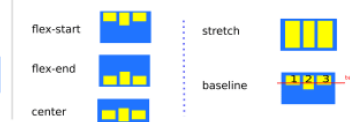
flex-wrap: allows the items to wrap as needed.



justify-content: defines alignment along the main axis.

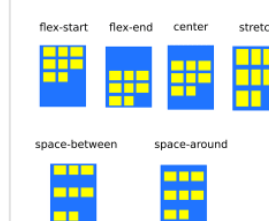


align-items: defines alignment along the cross axis.



LearnPine

align-content: aligns multiple lines, like justify-content does with individual items.



Children properties

order: changes the order of flex items.

```
.item {  
  order: 3 // the default is 0  
}
```

flex-grow: allows item to grow using remaining space.

```
.item-1 { flex-grow: 0 } .item-1 { flex-grow: 1 }  
//default
```

Tip: If all items have flex-grow: 1, the remaining space is distributed equally.

flex-shrink: defines the ability for a flex item to shrink.

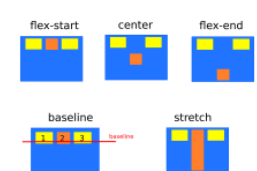
```
.one { flex-shrink: 1; }  
.two { flex-shrink: 2; }  
.three { flex-shrink: 3; }  
.four { flex-shrink: 4; }
```

Tip: Defaults to 1. The highest the value the more it shrinks compared to siblings.

flex-basis: sets the default size of a flex item. It accepts:

- specific values : pixels, rm, %
- auto : defaults to width or height property
- content : automatic sizing, based on its content
- global values : inherit, initial, unset

align-self: overrides default alignment (or the one specified by align-items) for a specific item.



Flexbox Cheat Sheet



@simonpaix

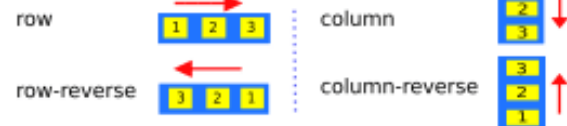


Parent properties

display: enables flex context for all direct children.

```
.container{  
  display: flex; // or inline-flex  
}
```

flex-direction: sets the main-axis.



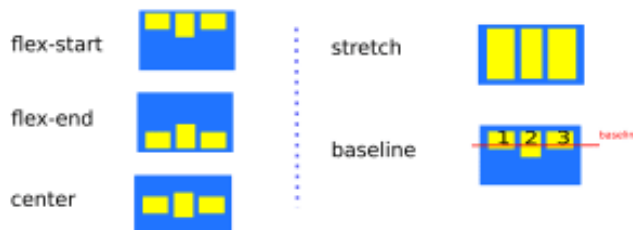
flex-wrap: allows the items to wrap as needed.



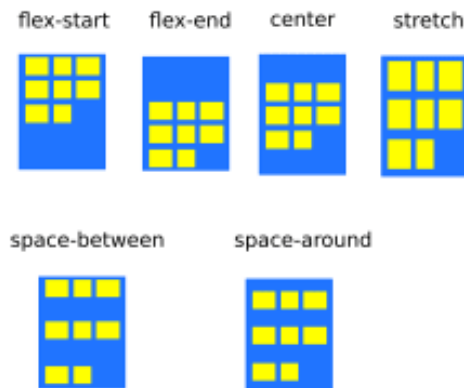
justify-content: defines alignment along the main axis.



align-items: defines alignment along the cross axis.



align-content: aligns multiple lines, like justify-content does with individual items.



Children properties

order: changes the order of flex items.

```
.item {  
  order: 3 // the default is 0  
}
```

flex-grow: allows item to grow using remaining space.

```
.item-1 { flex-grow: 0; } //default  
.item-1 { flex-grow: 1; }
```



Tip: If all items have flex-grow: 1, the remaining space is distributed equally.

flex-shrink: defines the ability for a flex item to shrink.

```
.one { flex-shrink: 1; }  
.two { flex-shrink: 2; }  
.three { flex-shrink: 3; }  
.four { flex-shrink: 4; }
```

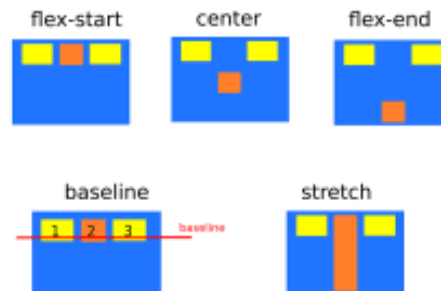


Tip: Defaults to 1. The highest the value the more it shrinks compared to siblings.

flex-basis: sets the default size of a flex item. It accepts:

- specific values : pixels, rm, %
- auto : defaults to width or height property
- content : automatic sizing, based on its content
- global values : inherit, initial, unset

align-self: overrides default alignment (or the one specified by align-items) for a specific item.



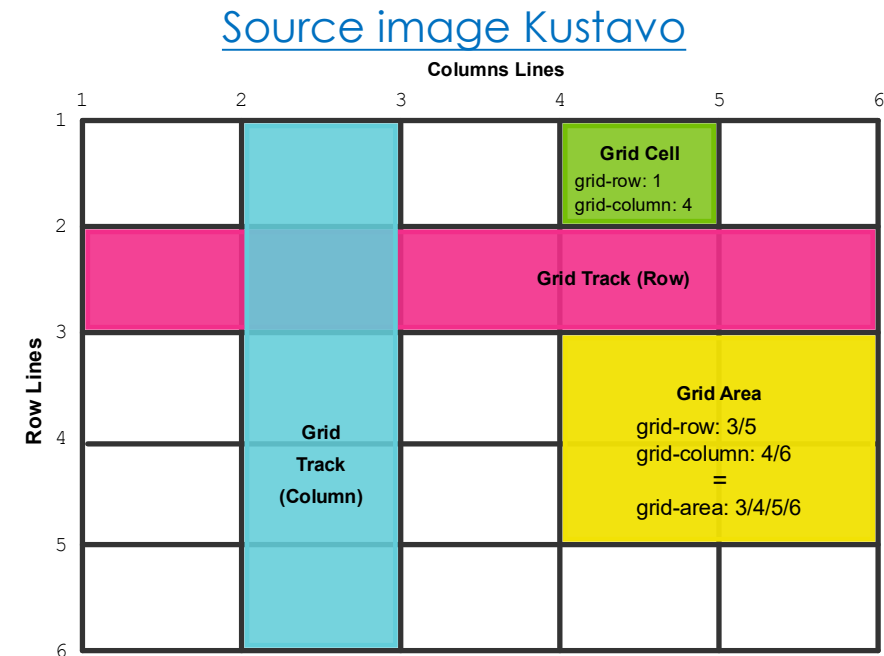
IV. Mise en page CSS Grid

A vous de jouer

[Jeux CSS Grid Garden pour apprendre à faire une mise en page complexe](#)

SYNTHÈSE CSS GRID

- Découpage en Grille
- Adapté pour les mises en pages complexes
- Un **conteneur** (l'élément parent)
 - ▶ Transformer en grille grâce à la déclaration **display : grid**
 - ▶ Les éléments vont se positionner selon 2 dimensions (colonne et ligne).
- Unité **fr unit** (fraction unit) correspond à une fraction de l'espace disponible dans le conteneur de la grille ([Source Developer Mozilla](#))



Grid Cheat Sheet



@simonpaix

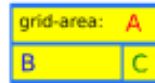
Parent properties

display: enables grid context for all direct children.
`.container{ display: grid; } // or inline-grid`

grid-template: defines the rows and columns of the grid. Set track size values and line-names(optional).

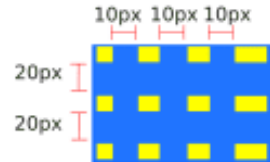
`grid-template-columns: 10px 30px auto 20%;`
`grid-template-rows: repeat(3, 20px);`

`grid-template-areas:`
`"A A A A"`
`"B B B C"`



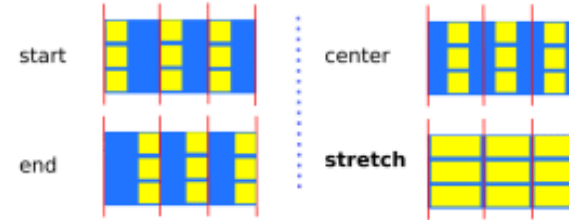
grid-gap: sets the size of the grid lines, the gutters between columns and between rows.

`column-gap: 10px;`
`row-gap: 20px;`

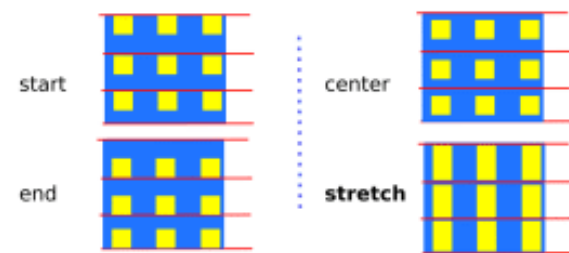


tip: default values are in bold

justify-items: defines alignment along the row axis.



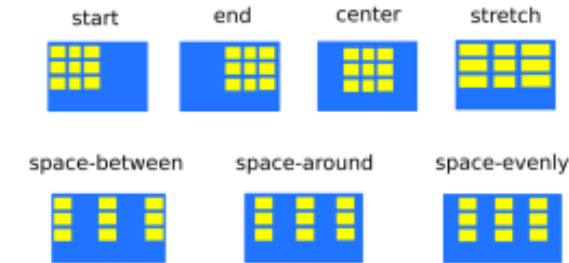
align-items: defines alignment along the column axis.



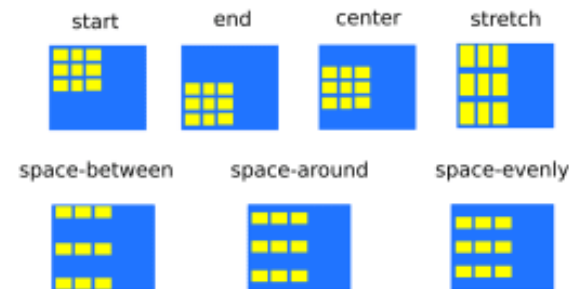
grid-auto-flow: defines how to automatically place grid items that aren't explicitly placed.



justify-content: justifies all grid content on row axis if total grid size is smaller than container.

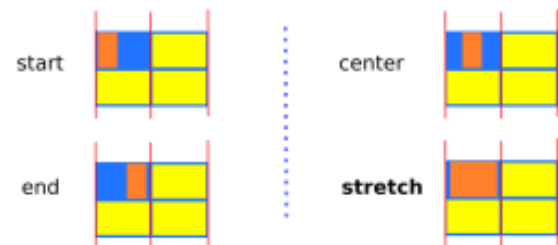


align-content: justifies all grid content on column axis if total grid size is smaller than container.

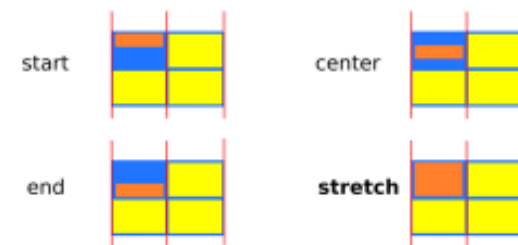


Children properties

justify-self: aligns an item inside a single cell along the row axis.



align-self: aligns an item inside a single cell along the column axis.



grid-column: determines the item's location based on a start and an end column lines (or a span).

`grid-column-start: 2;`
`grid-column-end: 4;`
`// or grid-column: 2 / 4`



grid-row: same but for the row location.

`grid-row-start: 1;`
`grid-row-end: 3;`
`// or grid-row: 1 / span 2`



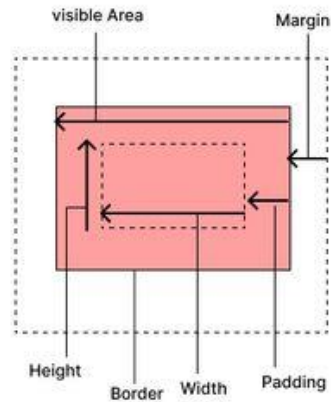
CSS Cheat Sheet



Selectors

*	All elements
div	<div>
div*	all elements with <div>
div span	 with <div>
div, span	<div> and
div > span	 with parent <div>
div+span	 precede by <div>
.class	Elements of class "class"
div.class	<div> of class "class"
#itemid	Element with id "itemid"
div#itemid	<div> with id "itemid"
a[attr]	<a> with attribute "attr"
a[attr='x']	<a> when "attr" is "x"
a[attr~='x']	<a> when class is a list containing
a[lang='en']	All elements

Box Model



Boxes

margin	border-color
margin-top	border-top-color
margin-right	border-right-color
margin-bottom	border-bottom-color
margin-left	border-left-color
padding	border-style
padding-top	border-top-style
padding-right	border-right-style
padding-bottom	border-bottom-style
padding-left	border-left-style
border	border-width
border-top	border-top-width
border-bottom	border-bottom-width
border-right	border-right-width
border-left	border-left-width

Pseudo-Selectors and -Classes

:first-child	First child element
:first-line	First line of element
:first-letter	First letter of element
:hover	Element with mouse over
:active	Active element
:focus	Element with focus
:link	Unvisited links
:visited	Visited links
:lang(var)	Element with language "var"
:before	Before element
:after	After element

Size and Colours

0	border-color
Relative Size	
em	1em equal to font size of parent (same as 100%)
ex	Height of lower case "x"
%	Percentage
Absolute Sizes	
PX	Pixels
cm	Centimeters
mm	Millimeters
in	Inches
pt	1pt=1/72 in
pc	1pc=12pt
Colours	
#789abc	RGB Hex Notation
#acf	Equates to "#aacff"
rgb(0,25,50)	Value of each of red, green, and blue, 0 to 255, may be swapped for percentages.

Positioning

display	clear
Position	z-index
top	direction
right	unicode-bidi
bottom	overflow
left	clip
float	visiblty

Dimensions

width	min-height
min-width	max-height
max-width	vertical-align
height	

Color / Background

color	background-repeat
background	background-image
background-color	background-position
background-attachment	

Text

text-indent	word-spacing
text-align	text-transform
text-decoration	white-space
text-shadow	line-height
letter-spacing	

Fonts

font	font-weight
font-family	font-stretch
font-style	font-size
font-variant	font-size-adjust

Tables

caption-side	border-spacing
table-layout	empty-cells
border-collapse	speak-header

Paging

size	page-break-inside
marks	page
page-break-before	orphans
page-break-after	widows

Interface

cursor	outline-style
outline	outline-color
outline-width	

Aural

volume	elevation
speak	speech-rate
pause	voice-family
pause-before	pitch
pause-after	pitch-range
cue	stress
cue-before	richness
cue-after	speak-punctuation
alay-during	speak-numeral
azimuth	

Miscellaneous

content	list-style-type
quotes	list-style-image
counter-reset	list-style-position
counter-increment	marker-offset
list-style	

[Source images APSignals](#)



FIN

MERCI POUR VOTRE ATTENTION

GLODIE TSHIMINI – ITIS 2025/2026