

# RA2: HTML I CSS

# EINES QUE UTILITZAREM

1. Instal·leu el Visual Studio Code <https://code.visualstudio.com/>
2. Utilitzarem el Google Chrome com a navegador
3. Farem servir aquestes extensions:

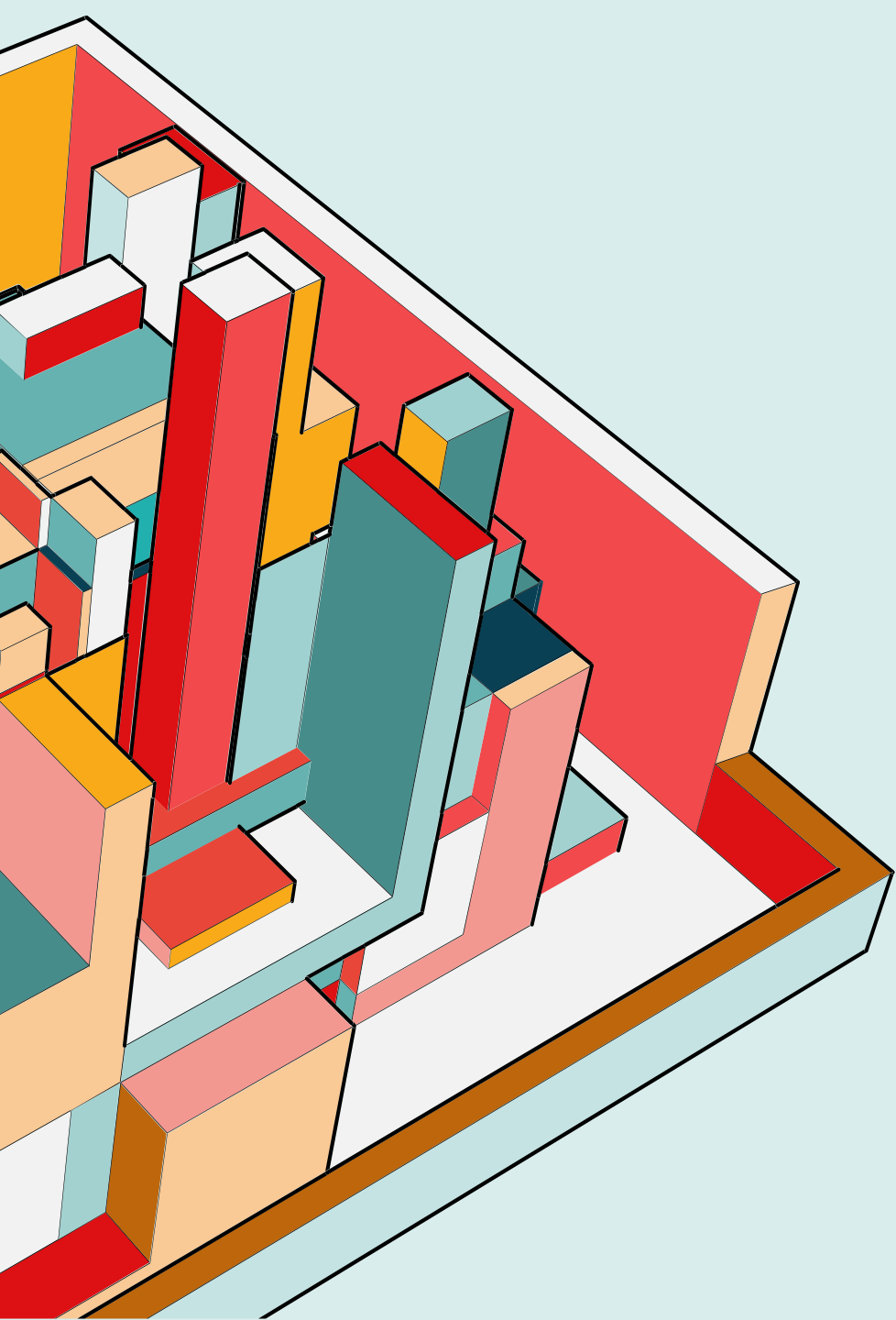
[Live Preview](#) : Permet visualitzar els canvis en temps real dels fitxers HTML, CSS i JavaScript directament dins l'evscose-icons

[Prettier](#): és una eina de formateig de codi automàtica que ajuda a mantenir el codi consistent i llegible. La seva funció principal és aplicar un estil de codificació uniforme segons unes regles definides prèviament

[vscode-icons](#): serveix per afegir icones visuals als arxius i carpetes dins de l'explorador de fitxers de VSCode que ajuden a identificar ràpidament els diferents tipus d'arxius.

# RA2 – HTML I CSS



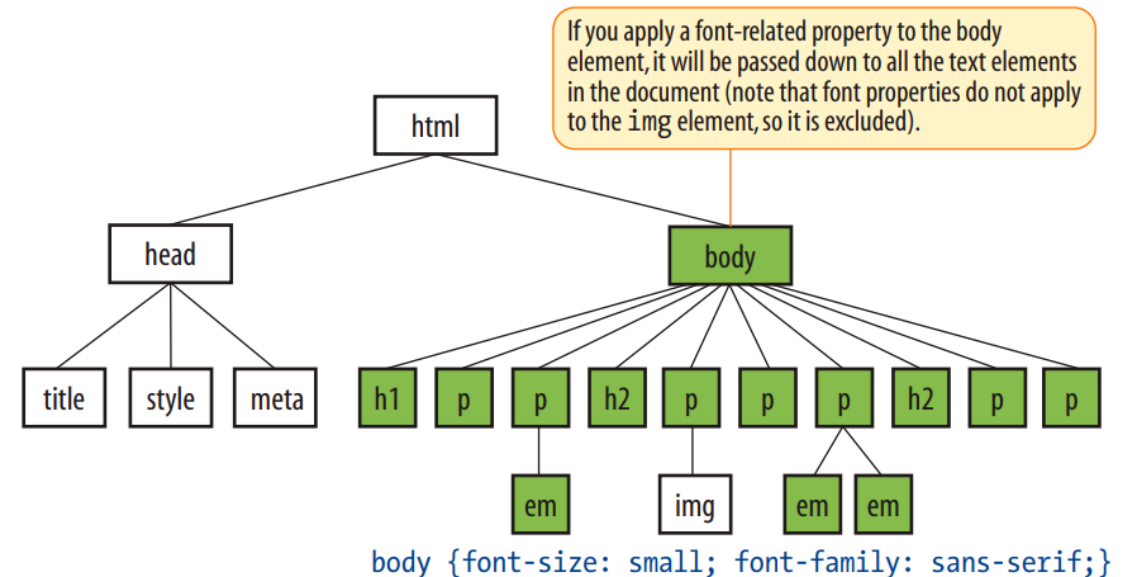
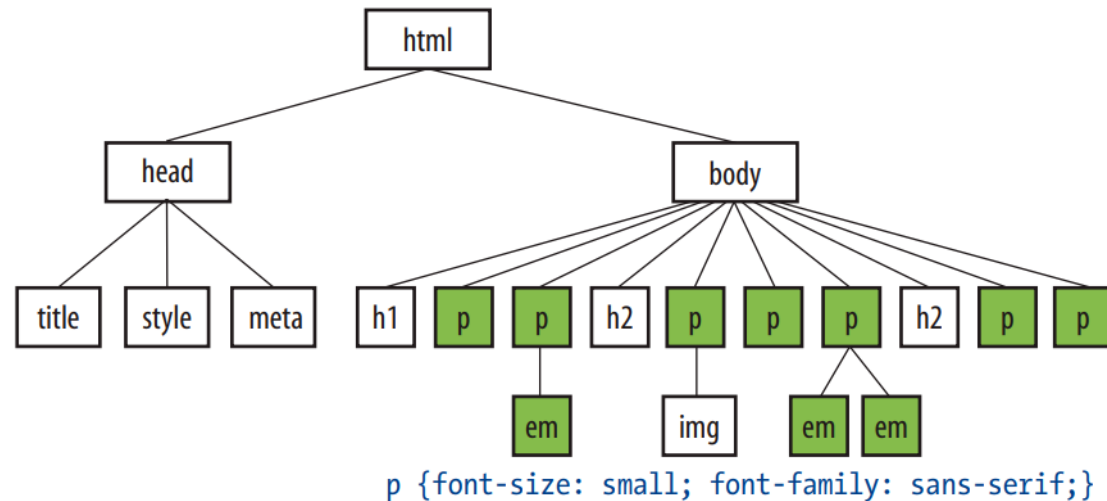


**CSS**

# 1. INTRODUCCIÓ

**CSS (Cascading Style Sheets)** és un llenguatge utilitzat per descriure l'aparença i el format d'un document escrit en HTML o XML. Això inclou aspectes com: Colors de fons i de text, tipus i grandàries de lletra, disposició d'elements a la pàgina, marge, espais i alineació

## CASCADING



## STYLE SHEET

<https://appbrewery.github.io/just-add-css/>

## 2. SINTAXI CSS

declaration

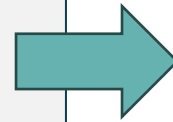
selector { property: value; }

declaration block

selector {  
property1: value1;  
property2: value2;  
property3: value3;  
}

### Exemple 1:

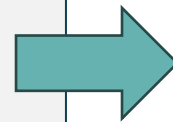
```
html {background:blue;}
```



Efecte: canvia el color de fons a blau de tot el document html

### Exemple 2:

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



Efecte: canvia el color del text de la capçalera h1 a vermell

### 3. TRES FORMES D'APLICAR CODI CSS

#### Inline

Com a atribut *style* d'una etiqueta

```
<etiqueta style="codi_css"/>
```

#### Internal

Com a codi CSS dins d'un element `<style>` a la capçalera

```
<head>  
  <style>codi_css </style>  
</head>
```

#### External

Com a arxiu a part *styles.css* que conté el codi CSS referenciat des d'un element *link* a la capçalera

```
<link href="styles.css"/>
```

#### Exemple:

*Arxiu index.html*

```
<html style="background:blue">  
  
</html>
```

#### Exemple:

*Arxiu index.html*

```
<head>  
  <style> html{  
    background:red;  
  }  
</style>  
</head>
```

#### Exemple:

*Arxiu index.html*

```
<link      rel="stylesheet"  
          href="./styles.css"  
>
```

*Arxiu styles.css*

```
html{  
    background:green;  
}
```



# SELECTORS





## 4. SELECTORS CSS (I)

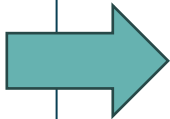
Els selectors seleccionen els elements als que s'apliquen les regles CSS que van entre els { }. N'hi ha de 5 tipus:

### Selectors d'element:

Indiquen l'element (<a>, <p>, <h1>...) al que s'apliquen les regles CSS. Aquestes regles s'aplicaran a tots els elements del mateix tipus seleccionat.

#### Exemple:

```
h1 {  
  color:blue  
}
```



Efecte: totes les capçaleres h1 existents seran de color blau

## 4. SELECTORS CSS (II)

### Selectors de classe (.):

Als elements es crea un atribut anomenat *class* que té de valor el *nom\_de\_la\_classe*. Aquest *nom\_de\_la\_classe* es pot aplicar a diferents elements.

A l'arxiu CSS s'utilitza el *.nom\_de\_la\_classe* precedit de *.* per a seleccionar tots els elements que tinguin aquest nom.

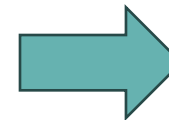
#### Exemple:

##### Codi HTML

```
<h2 class="cap_verd"> Capçalera de nivell 2 </h2>
```

##### Codi CSS

```
.cap_verd {
    color:green
}
```



**Efecte:** tots els elements de la classe cap\_verd seran text de color verd

## 4. SELECTORS CSS (III)

### Selectors de ID (#):

Als elements es crea un atribut anomenat *id* que té de valor el *nom\_del\_id*. Aquest *nom\_del\_id* es pot aplicar a un únic element.

A l'arxiu CSS s'utilitza el *.nom\_del\_id* precedit de # per a seleccionar l'element que tingui aquest nom.

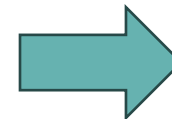
#### Exemple:

Codi HTML

```
<h2 id="main"> Capçalera de nivell 2 </h2>
```

Codi CSS

```
#main {
    color:red
}
```



**Efecte:** l'element amb el ID main tindrà text de color vermell

## 4. SELECTORS CSS (III)

### Selectors per atributs:

Selecciona només els elements (<a>, <p>, <h1>...) que tenen un cert valor d'atribut

A l'arxiu CSS s'utilitza aquesta sintaxis

`element[atribut="valor"]{propietat:valor}`

per a seleccionar l'element que tingui aquest nom.

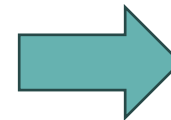
#### Exemple:

##### Codi HTML

```
<h2 draggable="true"> Capçalera de nivell 2 </h2>
```

##### Codi CSS

```
h2 [draggable="true"]
{
    color:red
}
```



**Efecte:** l'element h2 amb la propietat draggable=true tindrà text de color vermell

## 4. SELECTORS CSS (III)

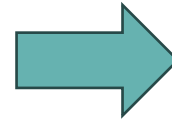
### Selectors universal (\*):

Selecciona tots els elements

**Exemple:**

**Codi CSS**

```
* {  
  color:red  
}
```



**Efecte:** tots els elements tindran text de color vermell

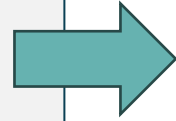
# COLORS



## 5. COLORS EN CSS

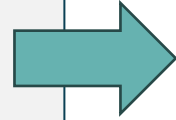
### Propietats *background* i *color*

```
html {background:blue;  
}
```



Efecte: canvia el color de fons a blau de tot el document html

```
h1 {background:red;  
color: white;}
```



Efecte: canvia el color del fons de la capçalera h1 a vermell i el color del text a blanc



## 5. COLORS EN CSS

Els colors es poden anomenar mitjançant:

- El nom del color: red, yellow, blue, chocolate, beige...

### Els noms dels colors

<https://developer.mozilla.org/en-US/docs/Web/CSS/named-color>

- El seu valor hexadecimal: #rrggbb, on
  - rr: valor de vermell (de 0 a 255) en hexadecimal
  - gg: valor de verd (de 0 a 255) en hexadecimal
  - bb: valor de blau (de 0 a 255) en hexadecimal

### Calculadora de colors

<https://www.csfieldguide.org.nz/en/interactives/rgb-mixer/>

### Paletes de colors

<https://colorhunt.co>

# PROPIETATS DE LA FONT



## 6. PROPIETATS DE LES FONTS: FONT-FAMILY

### font-family

Indica el tipus de lletra que utilitzem

### font-family

**Valors:** *una o més fonts o una família de font genèrica, separades per comes “,”*

**Font per defecte:** *depèn del navegador*

*Aplica a: tots els elements*

**S’hereda?:** *Sí*

### Exemples

```
body { font-family: Arial; }
```

```
tt { font-family: Courier, monospace; }
```

```
p { font-family: “Duru Sans”, Verdana, sans-serif; }
```

## 6. PROPIETATS DE LES FONTS: FONT-FAMILY

### font-family

#### Regles sintàctiques

- Tots els noms de tipus de lletra, a excepció de les famílies de tipus de lletra genèriques, han d'anar en majúscula. Per exemple, utilitzeu "Arial" en lloc de "arial".
- Utilitzeu comes per separar diversos noms de font
- Tingueu en compte que els noms de tipus de lletra que contenen un espai de caràcters (com ara Duru Sans) ha d'aparèixer entre cometes.
- És una **bona pràctica** utilitzar sempre dos noms de font-family: primer un de particular seguit d'un de genèric.

# 6. PROPIETATS DE LES FONTS: FONT-FAMILY

## font-family

### Famílies de font genèriques

Son famílies de fonts. Quan especifiquem una família de font, el navegador agafa una font que pertanyi a la família i que estigui disponible.

Les famílies de fonts genèriques, no es requereix que estiguin en majúscules.

#### Serif



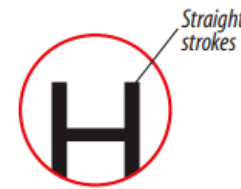
Hello  
Times

Hello  
Georgia

Hello  
Times New Roman

Hello  
Lucida (Mac)

#### Sans-serif



Hello  
Veranda

Hello  
Trebuchet MS

Hello  
Arial

**Hello**  
Arial Black

#### Monospace

W i  
Monospace font  
(equal widths)

W i  
Proportional font  
(different widths)

Hello  
Courier

Hello  
Courier New

Hello  
Andale Mono

#### Cursive

Hello  
Apple Chancery

Hello  
Comic Sans

Hello  
Snell

#### Fantasy

**Hello**  
Imapct

**HELLO**  
Stencil

**HELLO**  
Mojo

## 6. PROPIETATS DE LES FONTS: FONT-FAMILY

### font-family

#### Fonts de tercers

Hi ha repositoris de fonts des d'on seleccionar fonts per a la nostra pàgina web i garantir que el navegador disposarà d'aquella font.

Utilitzarem: **fonts.google.com**

## 6. PROPIETATS DE LES FONTS: FONT-SIZE

### font-size

Indica el tamany de lletra que utilitzem

### font-size

**Valors:** *unitat de mesura | percentatge | xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger*

**Per defecte:** *medium (16 px)*

**Aplica a:** *tots els elements*

**S'hereda?:** *Sí*



## 6. PROPIETATS DE LES FONTS : FONT-SIZE

### font-size

#### Tamanyys absoluts

Valors	nomenclatura	mm	exemple
pixels	px	0,26mm	20px
points	pt	0,35mm	20pt
paraules clau	xx-small, x-small, small, medium, large, x-large, xx-large, xxx-large		

#### Tamanyys relatiu

Valors	nomenclatura	exemple	significat
Ems	em	2em	2 vegades (200%) el tamany de l'element pare
Rems	rem	2rem	2 vegades (200%) el tamany de l'element arrel <html>
percentatge	%	80%	0,8 vegades el tamany de l'element pare
smaller / larger	smaller / larger		més petit o més gran que l'element pare.

## 6. PROPIETATS DE LES FONTS: FONT-WEIGHT

### font-weight

Indica l'amplada (negreta) de la lletra que utilitzem

### font-weight

**Valors:** *normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 |*

**Per defecte:** *normal*

**Aplica a:** *tots els elements*

**S'hereda?:** *Sí*

## 6. PROPIETATS DE LES FONTS: FONT-WEIGHT

### font-weight

#### Amplades absolutes

Valors	nomenclatura	exemple
normal	normal	normal
bold	bold	bold
valor del 100 al 900	valor del 100 al 900	150

#### Amplades relatives

Valors	nomenclatura	significat
lighter	lighter	100 punts més estreta respecte al pare
bolder	bolder	100 punts més ampla respecte al pare

## 6. PROPIETATS DE LES FONTS: FONT-STYLE

### font-style

La propietat d'estil de font afecta la postura del text, és a dir, si elles formes de les lletres són verticals (normals) o inclinades (cursiva i obliqua).

### font-style

**Valors:** *normal | italic | oblique*

**Per defecte:** *normal*

**Aplica a:** *tots els elements*

**S'hereda?:** *Sí*

## 6. PROPIETATS DE LES FONTS: FONT

### UNA DRECERA: FONT

El valor de la propietat font és una llista de valors, separats per espais de caràcters. En aquesta propietat, l'ordre dels valors són importants.

```
{ font: style weight variant size/line-height font-family }
```

Els valors en vermell son obligatoris. La resta són opcionals i si no es posen es prenen els seus valors per defecte.

## 6. PROPIETATS DE LES FONTS: FONT

### UNA DRECERA: FONT

Especificar diverses propietats de tipus de lletra per a cada element de text pot tornar-se repetitiu i llarg. Existeix la propietat **font** que compila totes les propietats relacionades amb el tipus de lletra en una regla.

#### font

**Valors:** *font-style font-weight font-variant font-size/line-height font-family*

**Per defecte:** *depèn dels valors defecte dels elements de la llista*

**Aplica a:** *tots els elements*

**S'hereda?:** *Sí*

## 6. PROPIETATS DEL TEXT

### text-align

Valors	nomenclatura
left	Alineat a l'esquerra
right	Alineat a la dreta
center	Alineat al centre
start	Igual que <i>left</i> si l'escriptura és d'esquerra a dreta i dreta si l'escriptura és de dreta a esquerra
end	Igual que <i>right</i> si l'escriptura és d'esquerra a dreta i esquerra si l'escriptura és de dreta a esquerra
justify	Justificat



# **PRIORITAT DE LES REGLES CSS**



# 1. PRIORITAT DE LES REGLES CSS

Quan més d'una regla CSS aplica a un element, cal determinar quina d'elles és la que s'aplica a l'element. Se segueix un model en cascada, on les regles de dalt la cascada son **menys** prioritàries i les de sota la cascada son **mes** prioritàries.

**Regla de la posició:** per un mateix element, aplica la regla que es llegeix en darrera posició

```
li {color: ■red;
    color: ■blue;
}
```

**Regla especificitat:** per un mateix element, aplica la regla del selector més específic

```
li {color: ■blue;}
.classe {color: ■red;}
li [draggable=true] {color: ■purple;}
#ident {color: ■yellow;}
```

**Regla aplicació estil:** per un mateix element, aplica primer la regla inline, després la interna i al final l'externa.

```
<link rel="stylesheet" href="./styles.css"/>
<style> </style>
<h1 style="">Hola</h1>
```

**Regla importància:** puc dobar prioritat a un atribut amb **!important**

```
li {color: ■red !important;
    color: ■blue;
}
```

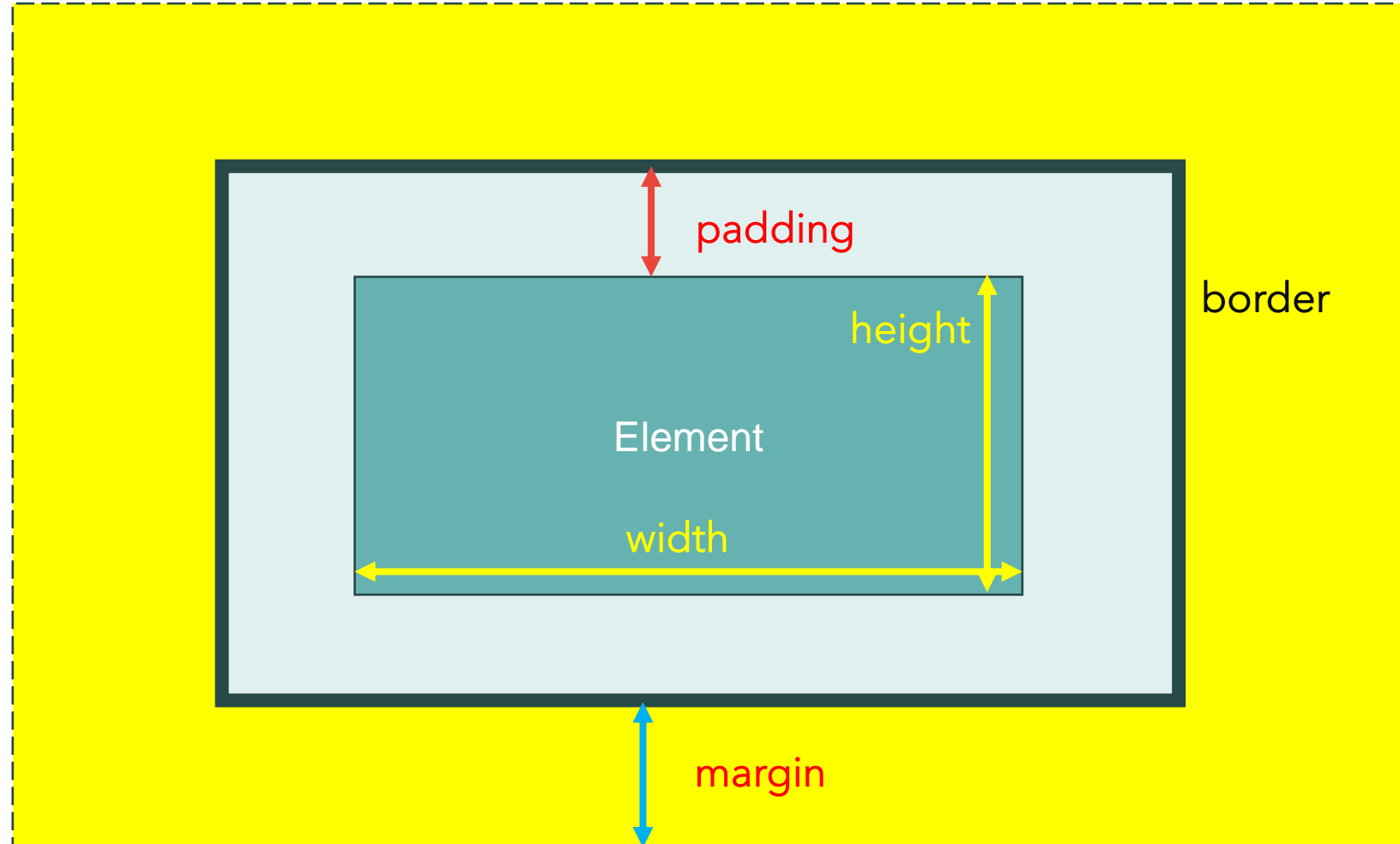
- prioritari

+ prioritari

# THE BOX MODEL



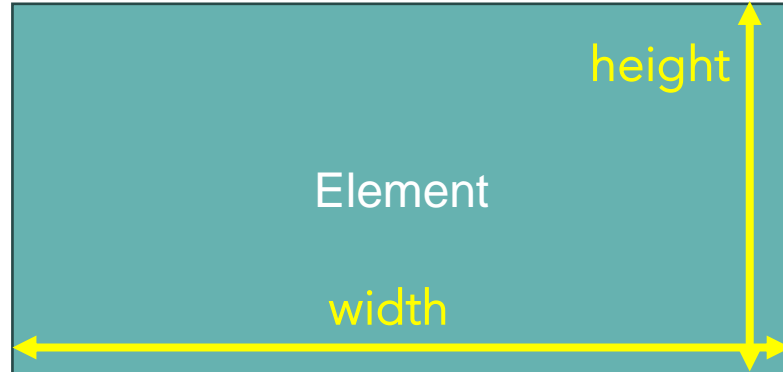
# 1. THE BOX MODEL



Aquesta àrea és invisible. Delimita l'espai amb altres elements.

# 1. THE BOX MODEL

## height i width



En HTML, els elements estan dins d'una “caixa” o BOX en anglès.

Puc definir **l'alçada i l'amplada** d'aquesta caixa amb les propietats:

### Valors absoluts

**height:** 300px (no s'acostuma a utilitzar)

**width:** 300px

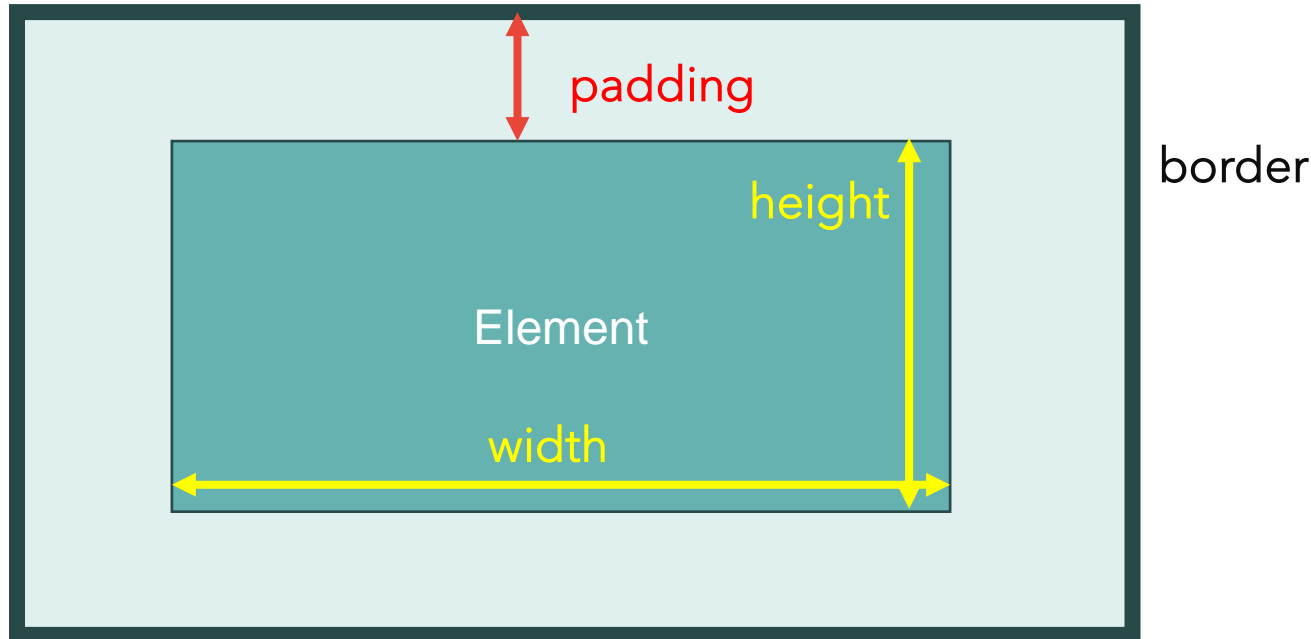
### Valors relatius

**width:** 100% (ocupa tota l'amplada de la pantalla)

**width:** "x" % (ocupa un percentatge x de tota l'amplada de la pantalla)

# 1. THE BOX MODEL

## padding



El **padding** ens indica la distància que hi ha entre un element i el seu **border**.

**S'utilitza quan l'element té text i es vol separar del border,**

Puc definir la distància del padding amb la propietat **padding**:

**padding: 20px**

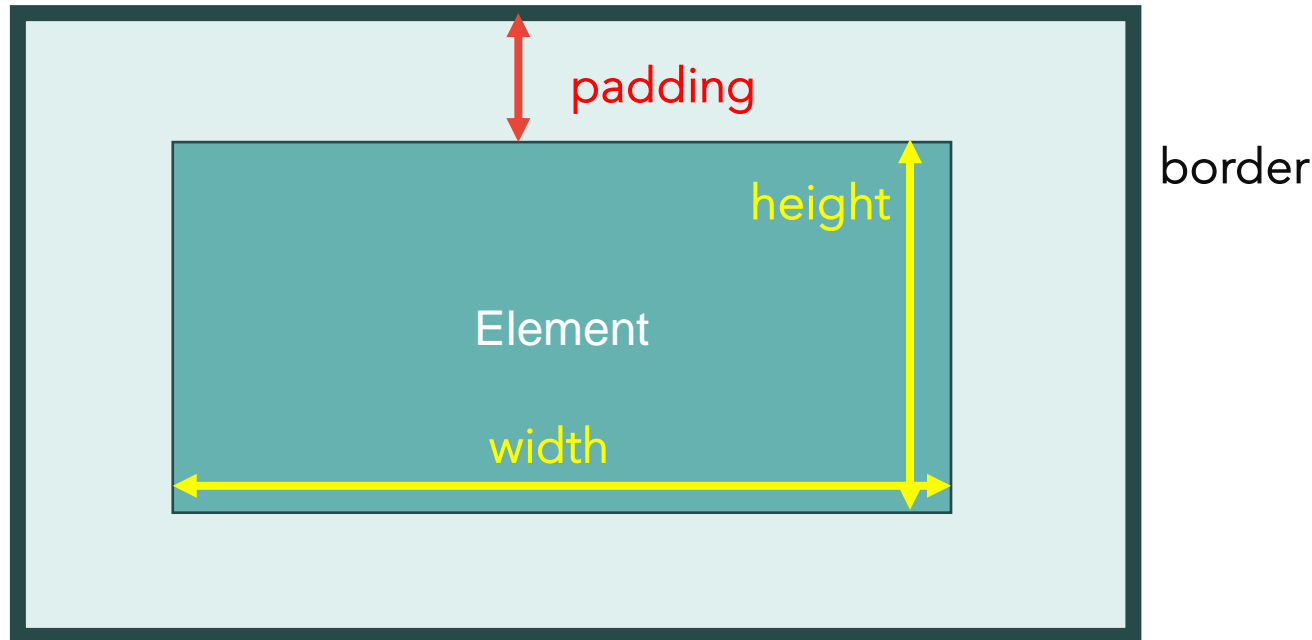
Existeixen les propietats **padding-left**, **padding-right**, **padding-top** i **padding-bottom** per especificar el padding de cada costat per separat si ho necessito.

Afegir un **padding** no modifica els valors **height** i **width** de l'element.



# 1. THE BOX MODEL

## border



El **border** és l'extrem de l'element, la seva vora, que es pot "separar" amb el padding.

**Es defineix amb 3 paràmetres:**

**border:** 20px solid black;

**border:** 5px dashed blue;

[tamany, estil, color]

Existeixen les propietats **border-left**, **border-right**, **border-top** i **border-bottom** per especificar el **border** de cada costat per separat si ho necessito.

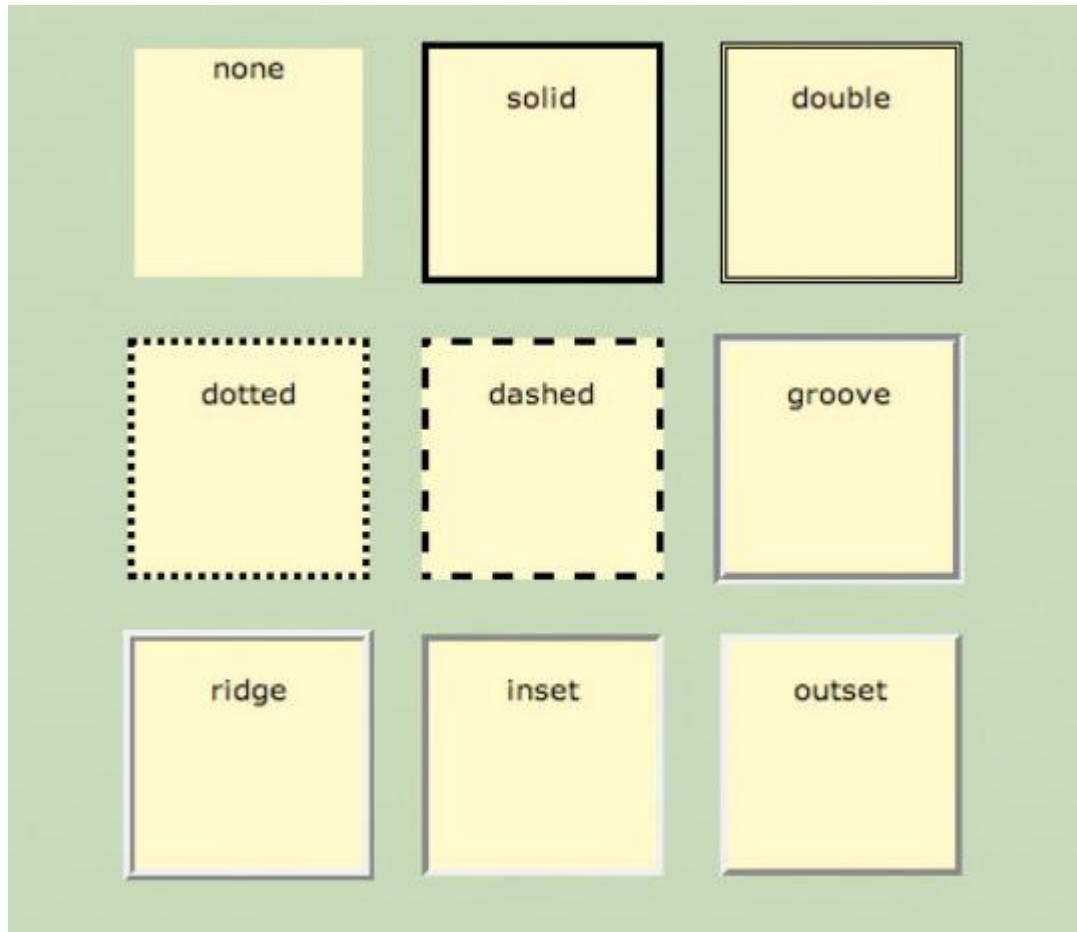
La propietat **border-width** permet definir de cop els valors de cada border en el sentit top > right > bottom > left.

**Exemple:** **border-width:** 5px 10px 15px 20px;



# 1. THE BOX MODEL

## Estils possibles per a *border*



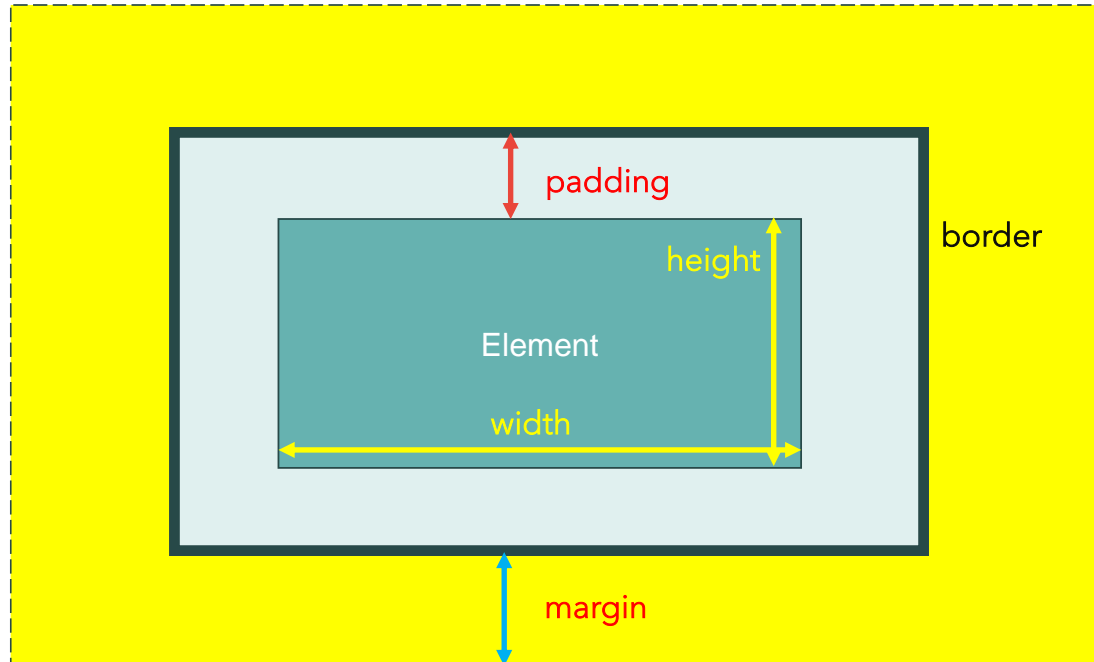
## Border-radius

**Proveu què passa amb:**

```
Border-radius: 20px;  
border-top-left-radius: 1 em;  
border-top-right-radius: 2 em;  
border-bottom-right-radius: 1 em;  
border-bottom-left-radius: 2 em;
```

# 1. THE BOX MODEL

## margin



El **margin** és el marge que s'afegeix fora del **border** per separar un element d'un altre.

Puc definir la distància del margin amb la propietat **margin**:

**margin: 20px**

Existeixen les propietats **margin-left**, **margin-right**, **margin-top** i **margin-bottom** per especificar el **margin** de cada costat per separat si ho necessito.

# 1. THE BOX MODEL

## Assignació de valors amb una propietat

Puc assignar els valors de les propietats **border-width**, **padding** i **margin** amb aquests criteris:

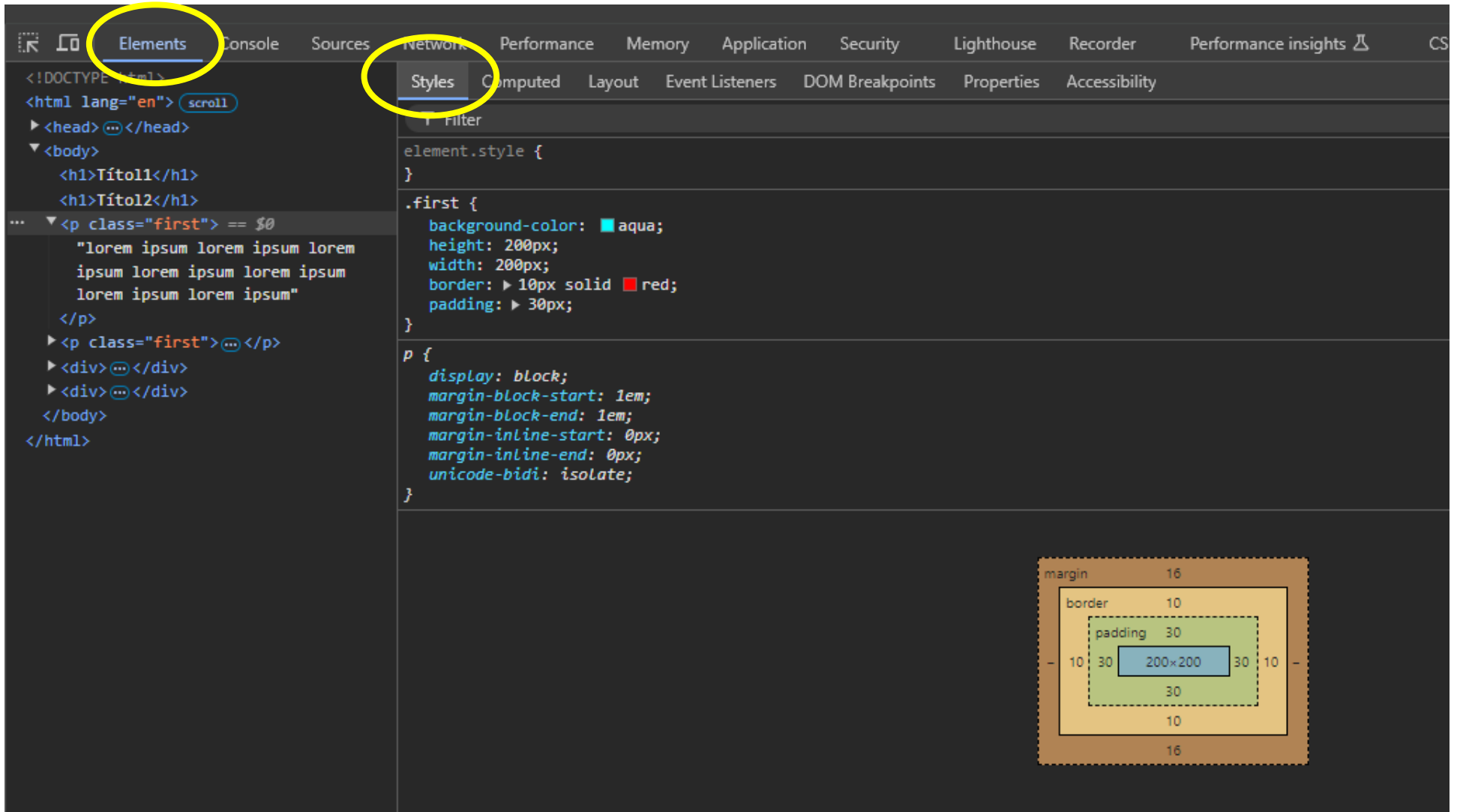
propietat: 10px; (*aplica als 4 costats*)

propietat: 0px 10px 20px 30px; (*0px top 10px right 20px bottom 30px left*)

propietat: 0px 10px; (*0px top i bottom i 10px right left*)

# 1. THE BOX MODEL

## Inspecció amb Google Chrome



The screenshot shows the Google Chrome DevTools interface. The **Elements** panel on the left displays the HTML structure, with a paragraph element selected: `<p class="first"> lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum </p>`. The **Styles** panel on the right shows the default styles for the `.first` class and the `p` element. The `.first` class styles are: `background-color: aqua;`, `height: 200px;`, `width: 200px;`, `border: 10px solid red;`, and `padding: 30px;`. The `p` element styles are: `display: block;`, `margin-block-start: 1em;`, `margin-block-end: 1em;`, `margin-inline-start: 0px;`, `margin-inline-end: 0px;`, and `unicode-bidi: isolate;`. A diagram of the Box Model is shown at the bottom right, illustrating the relationship between the content, padding, border, and margin.

Diagram illustrating the Box Model for the selected element:

- margin:** 16 (outermost layer)
- border:** 10 (middle layer)
- padding:** 30 (innermost layer)
- Content:** 200x200 (innermost box)

The diagram shows the total width and height of the element, including the margin, border, and padding, and the dimensions of the content area.

## 2. L'ELEMENT DIV (CONTENT DIVISION ELEMENT)

`<div> Contingut </div>`

És l'element que agrupa altres elements dins d'una caixa (BOX) invisible.

```
<div>
  <h2>hello world</h2>
  
</div>
```

### EXTENSIÓ PESTICIDE de CHROME

Com que és difícil fer debug sobre els elements div, el Chrome disposa de l'extensió PESTICIDE que en le mode *Inspecció* de Chrome dibuixa les caixes del box model a la pantalla on es mostra la pàgina web.

# PROPIETAT DISPLAY

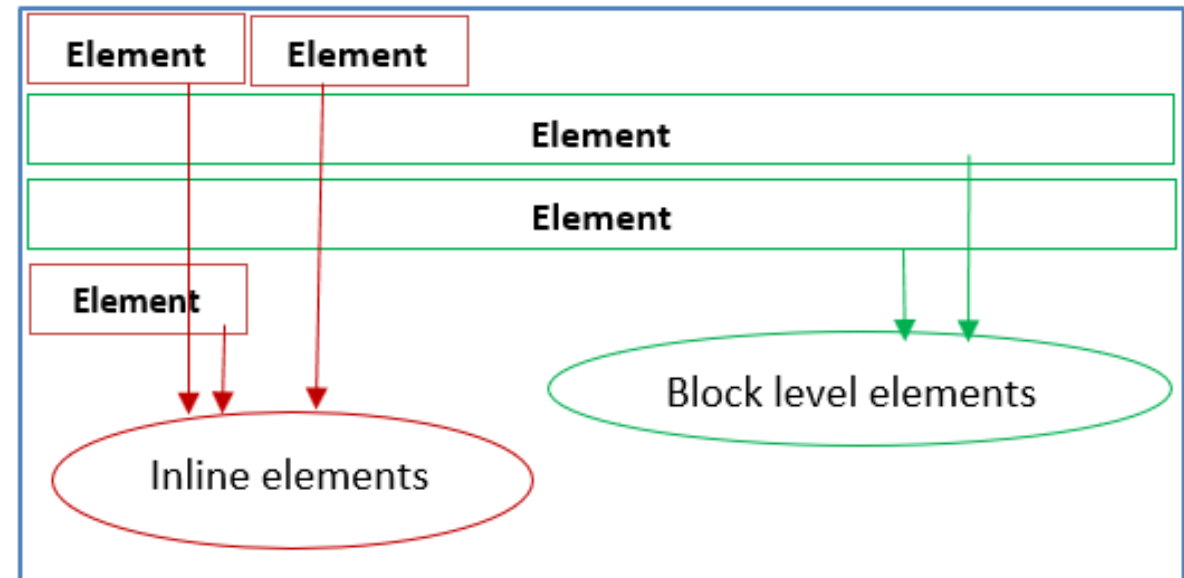


# 1. LA PROPIETAT DISPLAY

## Categories d'elements HTML:

Hi ha dos tipus de categories d'elements HTML.

- Un són els elements de **nivell de bloc**: **Block elements**. Cada element de nivell de bloc comença una línia nova i ocupa tota l'amplada. Per defecte, els següents elements son de tipus Block: <div>, <p>, <h>, <ul>, <ol>, <li>
- Uns altra són els elements **en línia**: **Inline elements**. Els elements en línia només ocupen l'espai necessari de la línia. Per defecte, els següents elements son de tipus Inline: <span>, <img />, <a>, <strong>, <em>)



# 1. LA PROPIETAT DISPLAY

La propietat display en CSS determina com un element es comporta visualment a la pàgina i com es disposa dins del seu contenidor.

Pot prendre diferents valors, però ara en veurem 4:

**display: block:** L'element es mostra com un bloc i ocupa tota l'amplada disponible. Comença en una nova línia.

- Puc definir la seva amplada (width) i alçada (height).
- Puc definir margin i padding.

**Display: Inline:** L'element es mostra en línia sense trencar la línia actual. Només ocupa l'amplada necessària pel seu contingut.

- **NO** puc definir la seva amplada (width) i alçada (height)
- **NO** puc definir margin i padding.

display: block

display: block	abarca 100% del width
display: block	abarca 100% del width
display: block	abarca 100% del width

display: inline

display: inline	display: inline	display: inline	display: inline
display: inline	display: inline	display: inline	



# 1. LA PROPIETAT DISPLAY

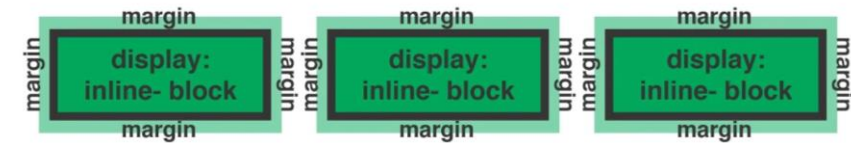
**display: inline-block:** És un comportament mixt de inline i block.

Cada element es comporta com un block pel que fa a que:

- Puc definir la seva amplada (width) i alçada (height).
- Puc definir margin i padding.

Cada element es comporta com un inline pel que fa a la ubicació contigua en una mateixa línia.

**display: inline-block**



**Display: none:** L'element no es mostra (ni ocupa espai). És útil per ocultar elements completament.

## 2. L'ELEMENT SPAN

`<span>` *Contingut* `</span>`

És un element en línia (**inline**), cosa que significa que no trenca el flux del text ni comença en una nova línia, a diferència dels elements de bloc com `<div>`.

```
<p>Aquesta és una text.</p>
```

### UTILITAT

- Per estilitzar part d'un text amb CSS.
- Per aplicar atributs personalitzats.

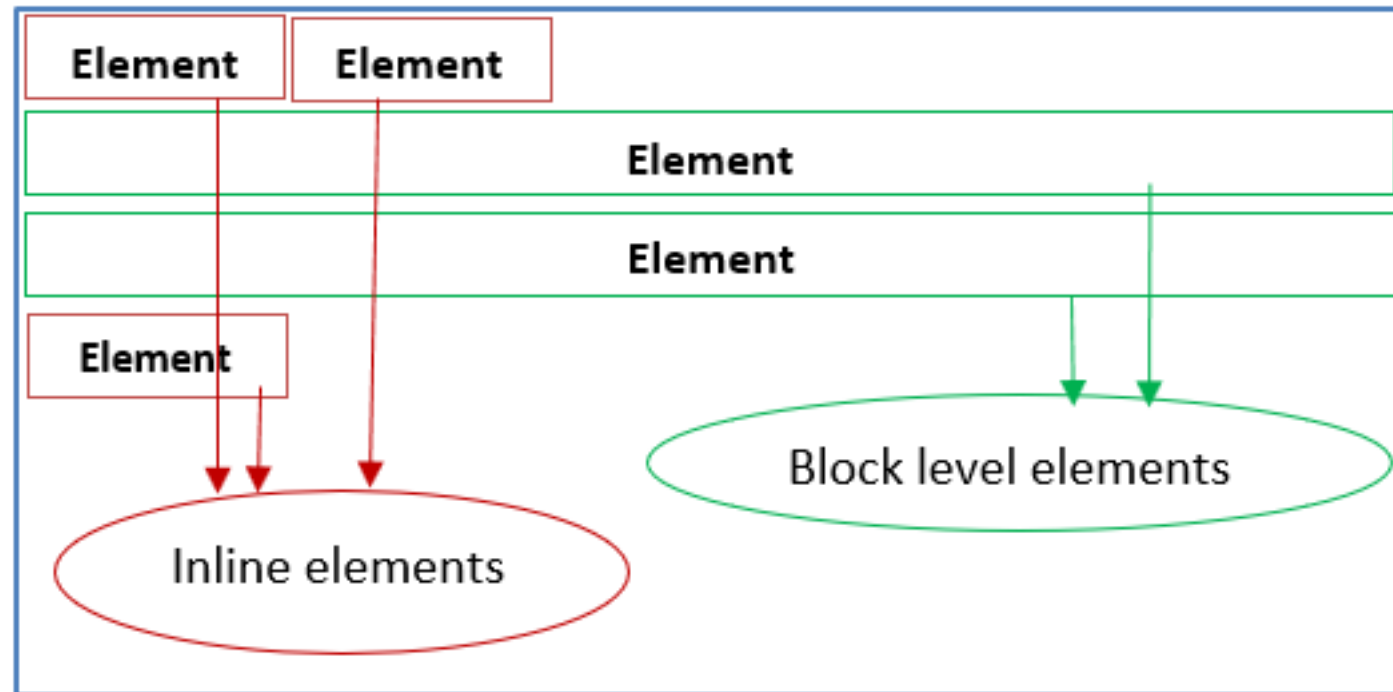
# PROPIETAT POSICIÓ (D'ELEMENTS)



# 1. POSICIÓ D'ELEMENTS

## Categories d'elements HTML:

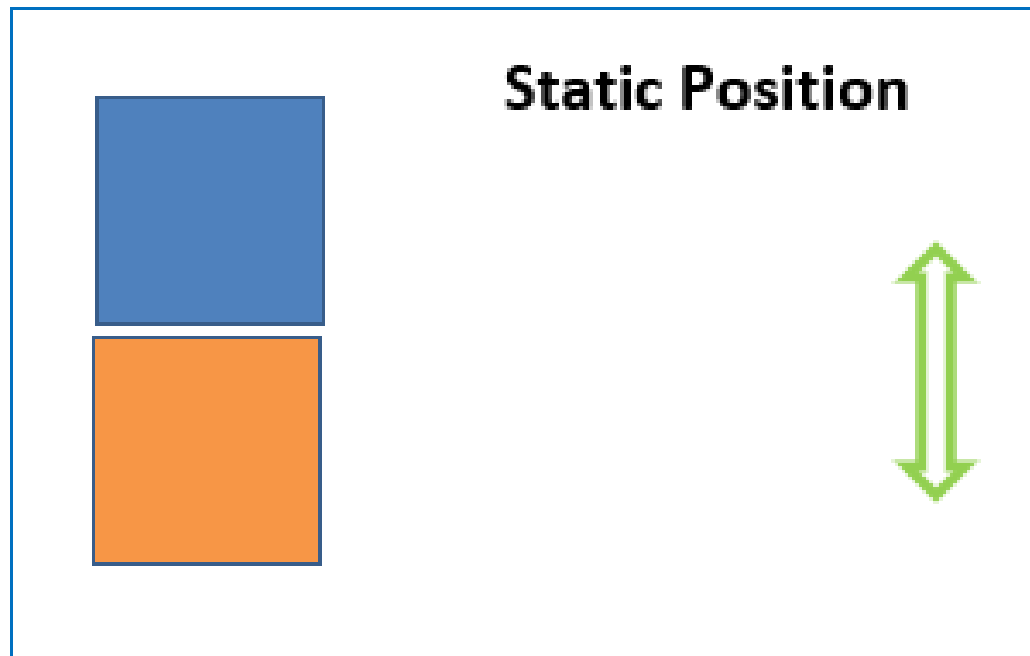
Les propietats de posició es poden aplicar als elements **Inline** i **Bloc**.



# 1. POSICIÓ D'ELEMENTS

Tenim 4 mètodes per a posicional elements a les nostres pàgines web:

**STATIC:** Aquesta posició col·loca l'element en funció del flux de documents normal. Aquí les propietats de posició (left, right, top, bottom) no són aplicables. **Aquesta és la posició de l'element per defecte.**



```
.static-box {  
    position: static; /* Opcional perquè és  
el valor per defecte */  
    background-color: lightblue;  
    padding: 10px;  
    margin: 10px;  
    margin: 10px;  
    top: 20px; /* No aplica*/  
    left: 20px; /* No aplica*/  
}
```

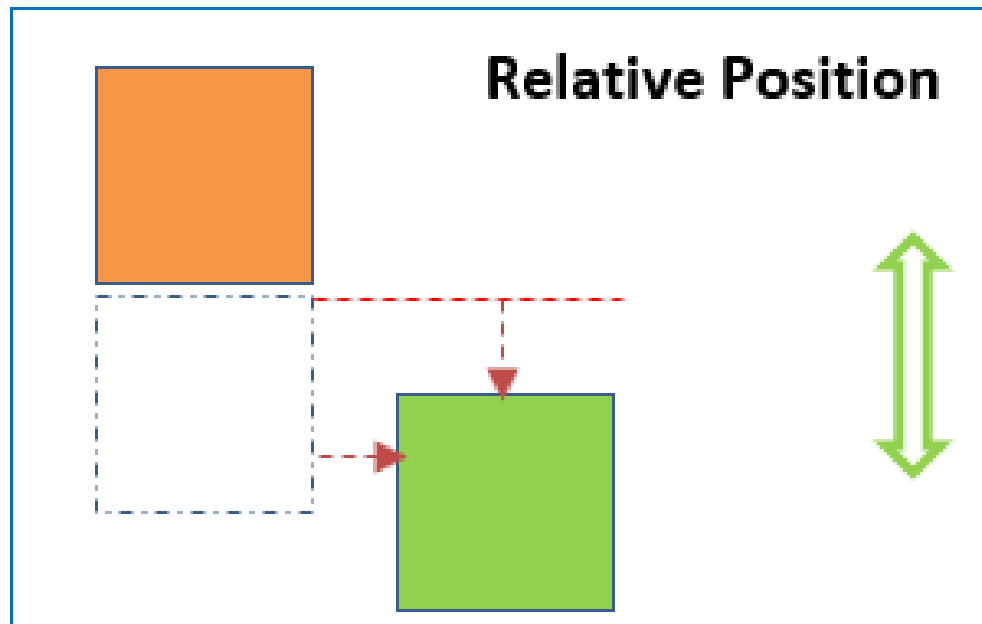
# 1. POSICIÓ D'ELEMENTS

**RELATIVE:** Aquesta posició situa l'element **en relació a la seva posició de flux normal**.

Utilitzant les propietats Left, Right, Top, Bottom, els elements de posició relativa es poden desplaçar de la seva posició de flux normal.

**Què és el flux normal?** La representació d'elements normalment un darrere l'altre es considera flux normal.

Els elements de posició relativa encara estan en el flux normal del document.

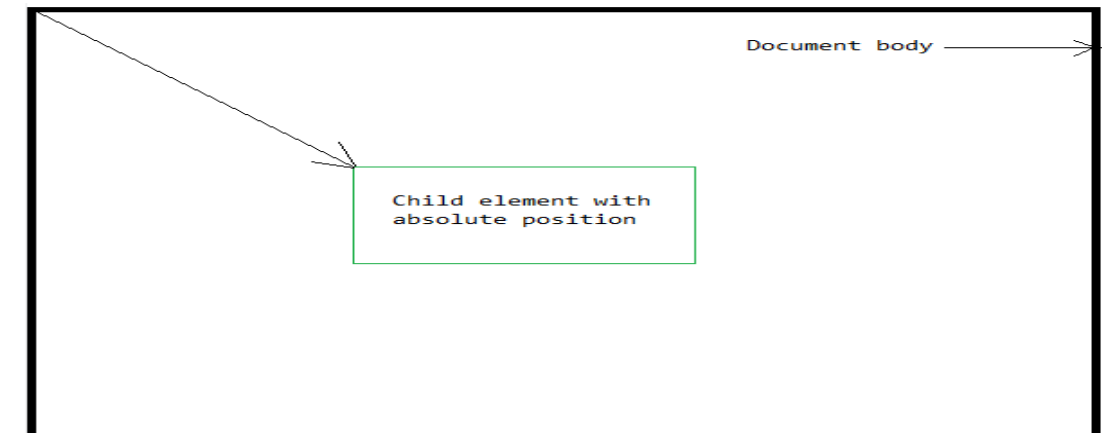
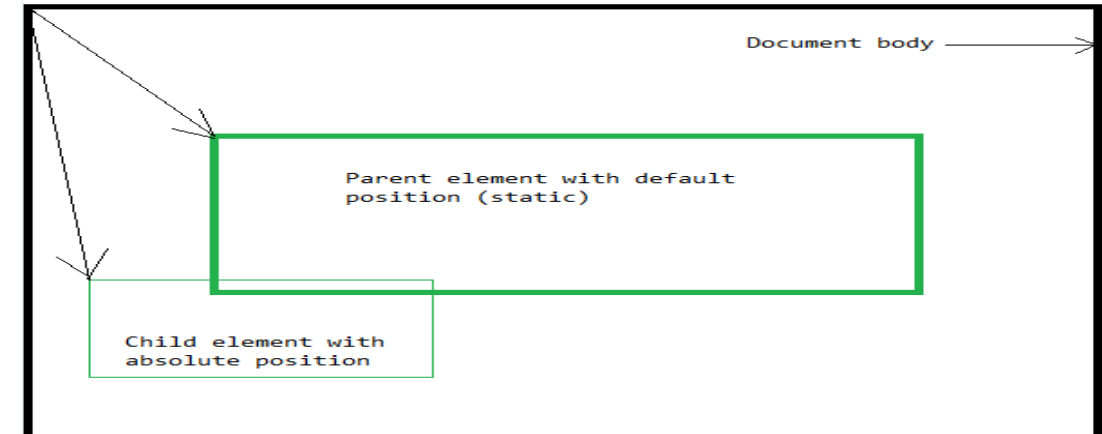
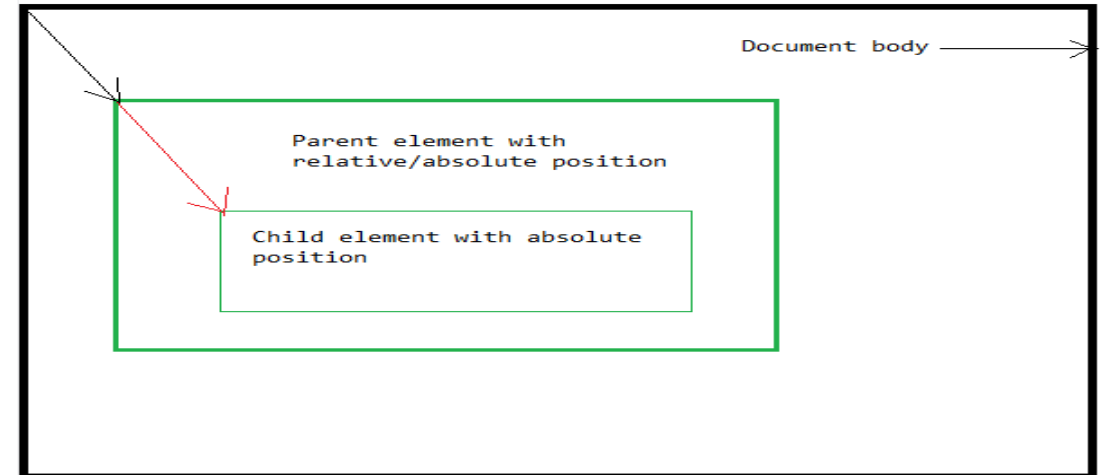


```
.relative-box {  
  position: relative;  
  background-color: lightgreen;  
  padding: 10px;  
  top: 20px; /* Aquesta propietat s'aplica  
perquè no és static */  
  left: 20px; /* Aquesta propietat s'aplica  
perquè no és static */  
}
```

# 1. POSICIÓ D'ELEMENTS

## ABSOLUTE:

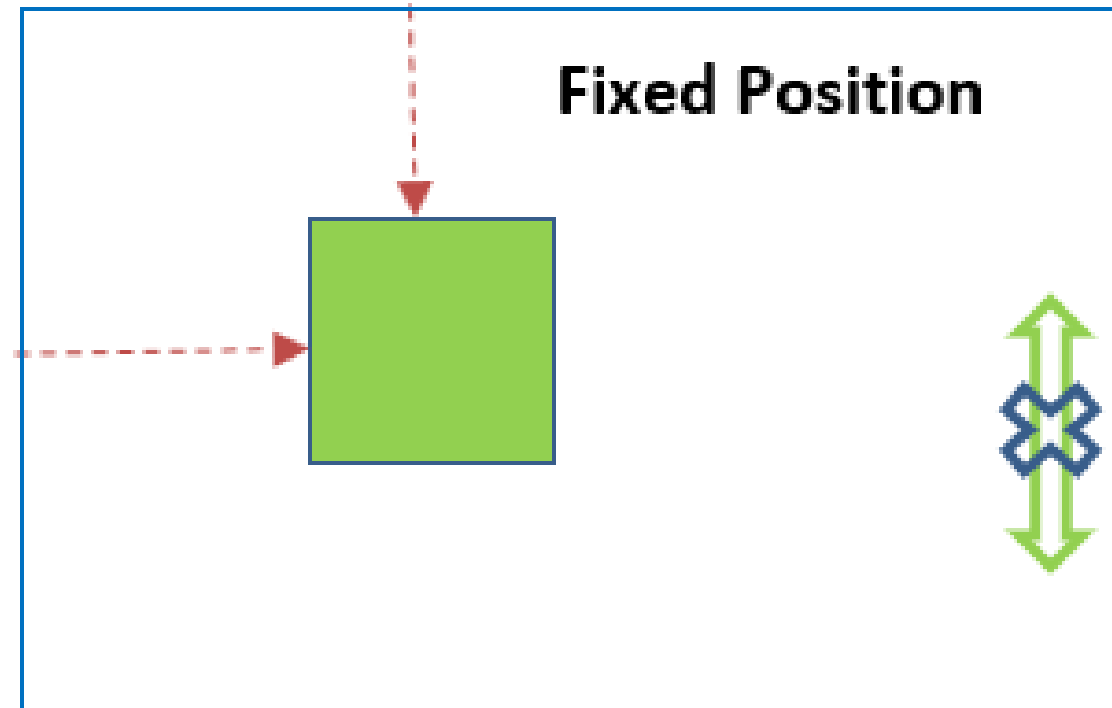
- a) L'element de posició absoluta es posiciona **en relació amb el seu element pare** quan la posició de l'element pare és relativa o absoluta.
- b) L'element de posició absoluta es posiciona **en relació amb l'element <body>** quan la posició de l'element pare és static.
- c) L'element de posició absoluta es posiciona **en relació amb l'element <body>** quan no hi ha element pare.



# 1. POSICIÓ D'ELEMENTS

## FIXED:

Ubica l'element en un lloc fix respecte a la finestra gràfica. El desplaçament de la pàgina no afectarà aquesta posició.





# PROPIETAT FLOAT



# 1. PROPIETAT FLOAT

**FIXED:**  
Ubica l'element en un lloc fix respecte a la finestra gràfica. El desplaçament de la pàgina no afectarà aquesta posició.



# 1. PROPIETAT FLOAT

La propietat float és una eina de disseny que permet posicionar elements dins d'una pàgina web i fer que altres continguts flueixin al voltant d'ells. És útil per crear layouts i estructurar continguts de manera visual.

## Valors de la propietat **float**

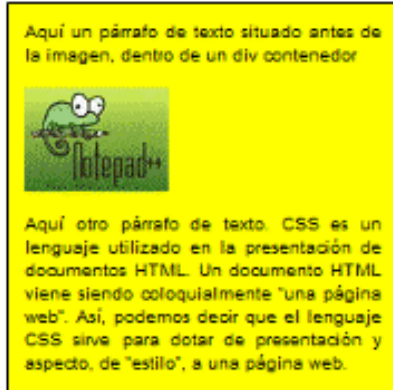
- **left:** L'element es posiciona al costat esquerre del contenidor, i altres continguts flueixen a la seva dreta.
- **right:** L'element es posiciona al costat dret del contenidor, i altres continguts flueixen a la seva esquerra.
- **none (per defecte):** L'element no flota; segueix el flux normal del document.
- **inherit:** L'element hereta el valor de float del seu contenidor pare.

Quan s'aplica la propietat float a un element, a aquest element se li ha de donar un valor a width, perquè si no ocupa tota l'amplada de la pantalla.

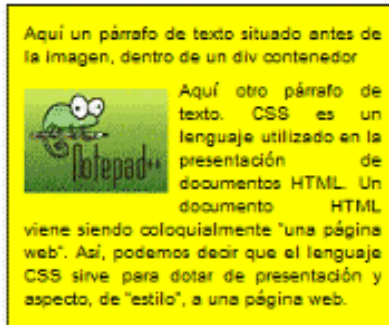


# 1. PROPIETAT FLOAT

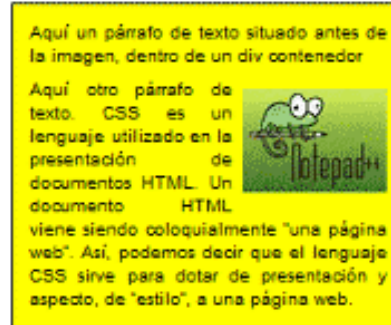
**float: none;**



**float: left;**

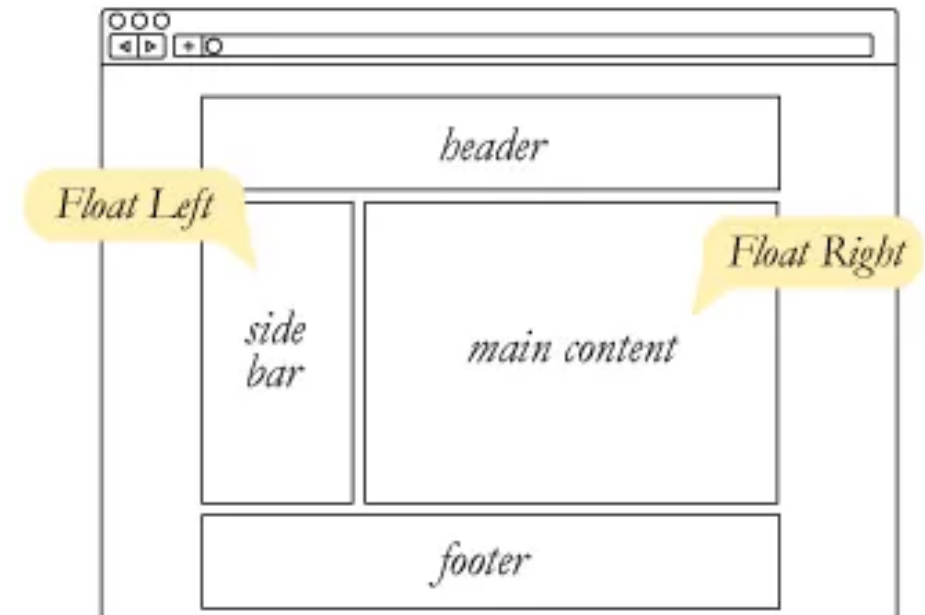


**float: right;**



Efectes de la propietat **float** sobre una imatge per als valors none, left i right..

La propietat **float** es pot aplicar a dos contenidors, per aconseguir situar-los un al costat de l'altre, tal com la mostra la figura de la dreta.

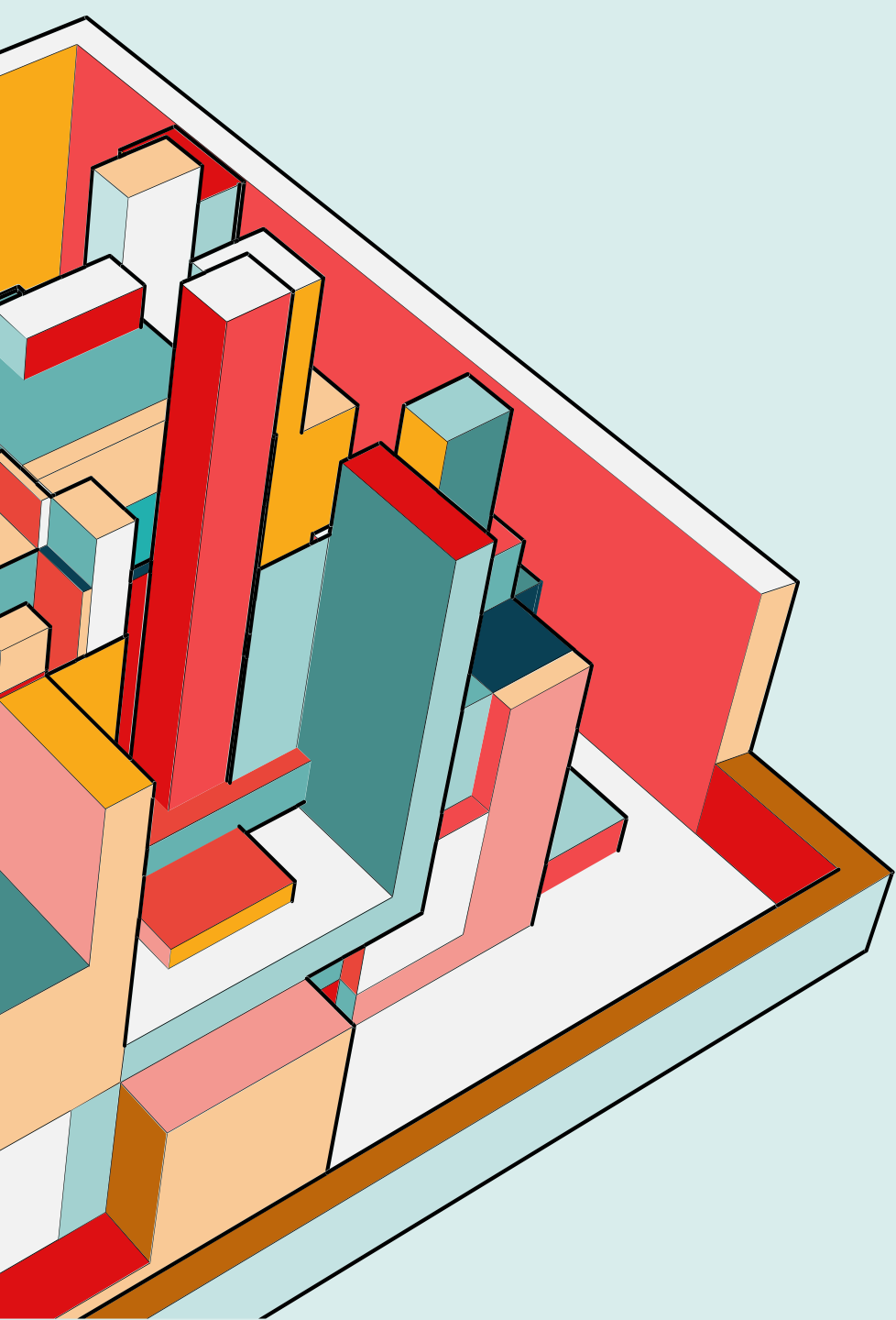


# 1. PROPIETAT FLOAT

El valor float:**clear** “**allibera**” a l’element que se li aplica haver-se de situar a l’esquerra o a la dreta d’altres elements. Recupera la disposició “normal” en HTML.

S’aplica a l’element <footer> per tornar-lo a ubicar en la posició normal a peu de pàgina.





# **CSS WEBS RESPONSIVES**

# MEDIA QUERIES



# 1. MEDIA QUERIES

Una **media query** és una funció de CSS que permet aplicar regles d'estil basades en les característiques del dispositiu on es visualitza la pàgina web. S'utilitzen per fer que els dissenys siguin responsius, adaptant-se a diferents pantalles, com mòbils, tauletes i ordinadors.

Una media query detecta propietats del dispositiu, com ara l'amplada de la finestra (viewport) i en funció d'aquestes característiques, permet aplicar diferents regles CSS per personalitzar l'experiència de l'usuari.

## sintaxi:

```
@media (condició) {  
    /* Regles CSS aplicades si la condició és certa */  
}
```

## Exemples de condicions més comuns:

**min-width:** Especifica un estil que s'aplica quan l'amplada és **com a mínim** el valor especificat. (**>**)

**max-width:** Especifica un estil que s'aplica quan l'amplada és **com a màxim** el valor especificat. (**<=**)



# 1. MEDIA QUERIES

## exemple:

```
div{  
  background-color: blue;  
  height: 200px;  
  width: 200px;  
}  
  
@media (max-width:600px) {  
  div{  
    background-color: blue;  
    height: 100px;  
    width: 100px;  
  }  
}
```

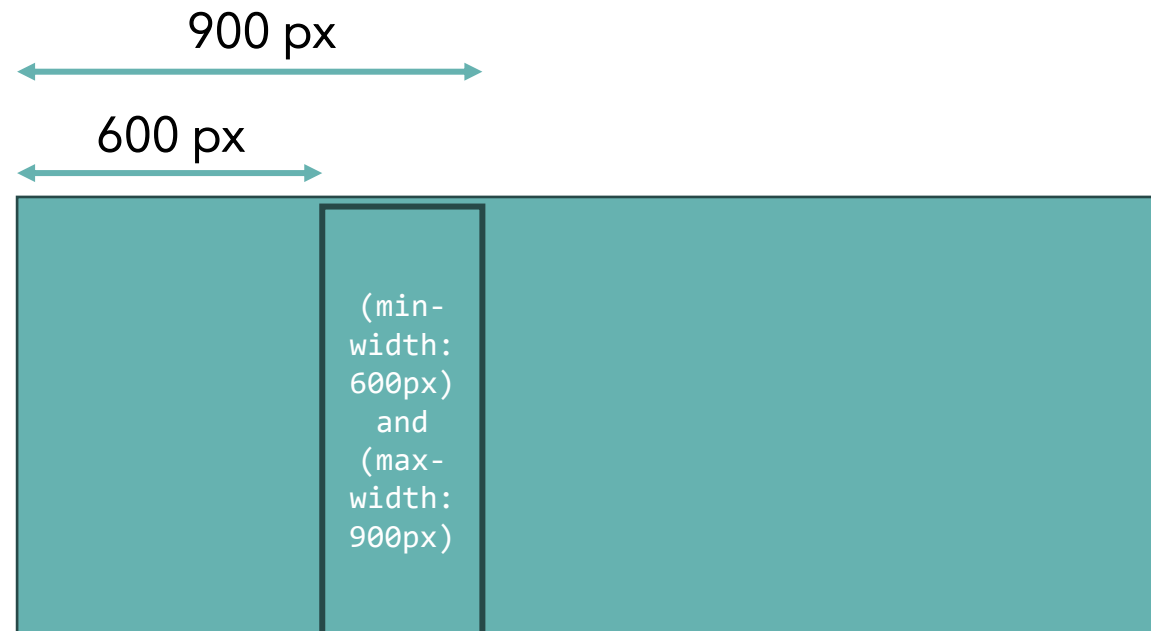


# 1. MEDIA QUERIES

Les media queries es poden combinar:

**exemple:**

```
@media (min-width: 600px) and (max-width: 900px) {  
    /* Regles CSS aplicades si la condició és certa */  
}
```



# FLEXBOX



# 1. FLEXBOX

**Flexbox** (o Flexible Box Layout) és un model de disseny utilitzat per distribuir i alinear elements dins d'un contenidor de manera eficient, fins i tot quan les mides dels elements són dinàmiques o desconegudes.

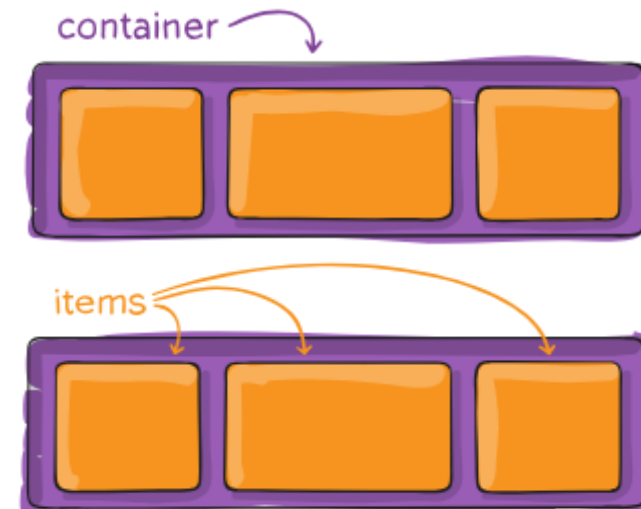
## Característiques clau:

- 1. Disseny d'una dimensió:** Flexbox funciona principalment al llarg d'una dimensió, ja sigui una fila (*row*) o una columna (*column*).
- 2. Control de l'espai:** Permet gestionar com es distribueixen els elements, la separació entre ells i com s'alineen dins del contenidor.
- 3. Adaptabilitat:** S'adapta automàticament a canvis en la mida del contenidor o els seus elements.

## Conceptes bàsics:

- **Flex container:** És l'element pare al qual s'aplica la propietat **display: flex**; Tots els seus fills es converteixen en flex items.

- **Flex items:** Són els elements fills directes d'un contenidor flex.



# 1. FLEXBOX

## Propietats de Flexbox:

### Propietats del contenidor

- **display: flex;** → Activa el comportament flexbox.
- **flex-direction** → Defineix la direcció principal (horitzontal o vertical).
  - row (**per defecte**), row-reverse, column, column-reverse.
- **Flex-wrap** → Controla si els ítems d'un contenidor salten a la següent línia o no en cas de no cabre al contenidor
  - nowrap (**per defecte**), wrap, wrap-reverse
- **justify-content** → Controla la distribució dels ítems al llarg de la direcció principal.
  - Ex.: flex-start (**per defecte**), center, space-between, space-around.

# 1. FLEXBOX

## Propietats de Flexbox:

### Propietats del contenidor

- **align-items** → Controla l'alineació al llarg de l'eix transversal. (només aplica quan flex-wrap: nowrap)
  - Ex.: flex-start (per defecte), stretch, center, flex-end.
- **align-content** → Gestiona l'alineació de múltiples línies en contenidors de diverses files o columnes. (només aplica quan flex-wrap: wrap)
- **gap** → Controla l'espai entre ítems (no amb els costats del contenidor)

```
.container {  
  display: flex;  
  ...  
  gap: 10px;  
  gap: 10px 20px; /* row-gap column gap */  
  row-gap: 10px;  
  column-gap: 20px;  
}
```

# 1. FLEXBOX

## Propietat display (contenidor)

### Codi HTML

```
<div class="container">
  <div class="red">Red</div>
  <div class="orange">Orange</div>
  <div class="yellow">Yellow</div>
  <div class="green">Green</div>
  <div class="blue">Blue</div>
  <div class="indigo">Indigo</div>
  <div class="purple">Purple</div>
</div>
```

### Codi CSS

```
.container{
  display: flex;
  background-color: yellow;
  border: 10px solid green;
  gap: 10 px;
}
```

`display: flex;`

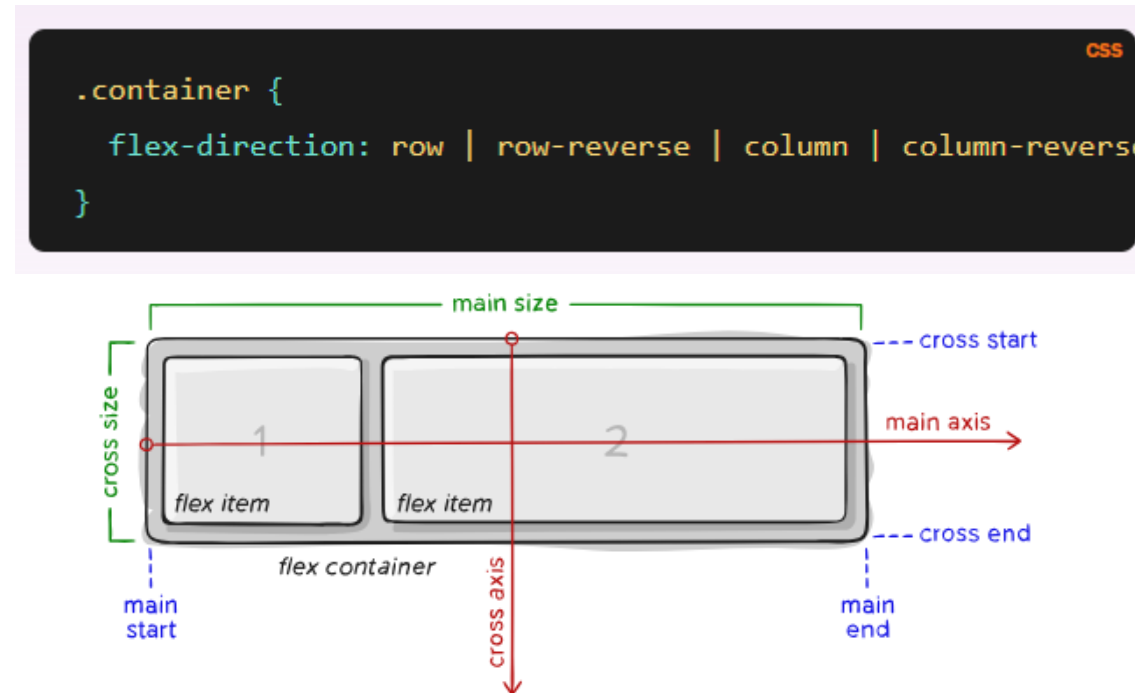
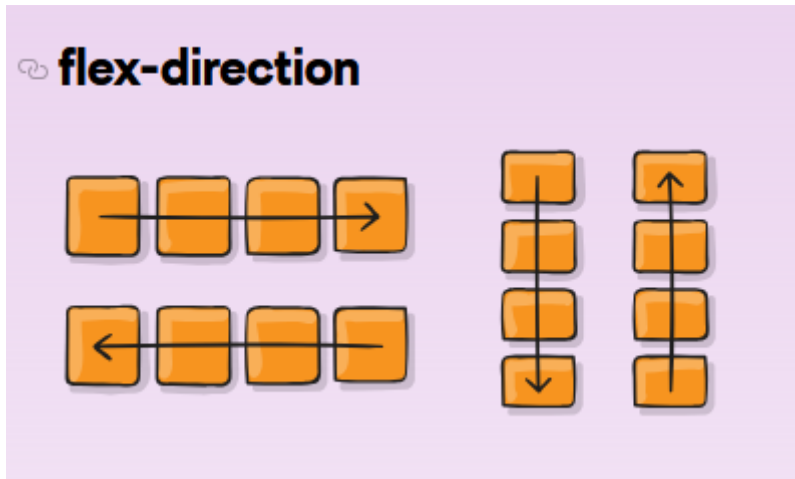
Red Orange Yellow Green Blue Indigo Purple

`display: inline-flex;`

Red Orange Yellow Green Blue Indigo Purple

# 1. FLEXBOX

## Propietat flex-direction (contenedor) Defineix la direcció principal



**eix principal:** l'eix principal és l'eix al llarg del qual es disposen els elements flexibles. Compte, no és necessàriament horitzontal; depèn de la propietat flex-direction

**eix secundari:** eix perpendicular al principal.



# 1. FLEXBOX

## Propietats del contenidor

La resta de propietats del contenidor a:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

## Propietats de Flexbox

### Propietats del ítems

- **flex-basis** → Defineix la mida inicial de l'item abans de distribuir espai
- **flex-grow** → Controla la proporció d'espai extra que ocupa un item.
- **flex-shrink** → Determina com es redueix un item quan hi ha poc espai.
- **extra.order** → Reorganitza l'ordre visual dels ítems.

# 1. FLEXBOX

## Propietat dels ítems

### Flex-basis

S'utilitza en un contenidor flexible per **establir la mida inicial d'un element** abans que es distribueixi l'espai restant dins del contenidor. És una de les tres propietats principals que determinen com es dimensionen els elements flexibles, juntament amb flex-grow i flex-shrink.

### Valors possibles:

**1.auto** (valor per defecte): L'element utilitza el valor de la propietat width o el contingut intrínsec per determinar la seva mida inicial.

**2. 0:** Si es posa a zero, no aplica el flex-basis i només controlen el tamany els paràmetres flex-grow i flex-shrink

**2.<mida>:** Pot ser una mida fixa, com ara:

- px (píxels): flex-basis: 200px;
- % (percentatge): flex-basis: 50%;
- em, rem, vw, etc.: flex-basis: 10em;

Aquest valor indica la mida inicial de l'element abans d'aplicar altres càlculs de flex.

# 1. FLEXBOX

## Propietat dels ítems

### Flex-grow

controla com un element flexible **creix per ocupar l'espai disponible en un contenidor** flexible (amb `display: flex;`).

#### Valors possibles:

- **0 (per defecte):** L'element no creixerà per ocupar espai addicional. La seva mida serà determinada únicament per `flex-basis` (o `width` si no es defineix `flex-basis`).
- **Número positiu:** Indica la proporció de creixement de l'element.  
Per exemple: Si un element té `flex-grow: 2` i un altre `flex-grow: 1`, el primer creixerà el doble que el segon.

# 1. FLEXBOX

## Propietat dels ítems

### Flex-shrink

controla com un element flexible **es redueix quan no hi ha prou espai disponible en un contenidor flexible** (és a dir, quan el contingut del contenidor excedeix la seva mida).

#### Valors possibles:

- **1 (per defecte):** L'element es pot reduir proporcionalment amb altres elements per ajustar-se a l'espai disponible.
- **0:** L'element no es reduirà mai, mantenint sempre la seva mida especificada (a través de flex-basis o width).
- **Número positiu:** Indica la proporció de reducció de l'element.
  - Per exemple: Si un element té flex-shrink: 2 i un altre té flex-shrink: 1, el primer es reduirà el doble que el segon.

# 1. FLEXBOX

## Propietat dels ítems

### Flex

La propietat flex és una abreviació de les 3 anteriors:

#### Sintaxi

CSS

```
flex: <flex-grow> <flex-shrink> <flex-basis>;
```

#### Valors abreujats:

**flex:1;** /\*Igual a flex:1 1 0;\*/

**flex:2;** /\*Igual a flex:2 1 0;\*/

**flex:3;** /\*Igual a flex:3 1 0;\*/

**flex:4;** /\*Igual a flex:4 1 0;\*/

Home

About

Contact Us

La Millor Escola De Programació  
Online

