

Llenguatges de Marques i Sistemes de Gestió d'Informació

UD 4.4 CSS, el model de caixa



IES Jaume II El Just
Tavernes de la Valldigna
Departament d'Informàtica
Curs 2024-25

Índex

1	Model de caixes (Box Model)	2
1.1	Amplada i alçada	2
1.2	Margin i padding	3
1.3	Border	4
1.4	Vores arrodonides	5
1.5	Propietat box-sizing	5
2	Elements de bloc i en línia	7
2.1	Visibilitat dels elements	8
2.2	Crear un menú de navegació amb CSS	8
2.3	Referències	11

1 Model de caixes (Box Model)

Quan es visualitza qualsevol element HTML és tractat com si estiguera **dins d'una caixa** rectangular.

Cada caixa té quatre components:

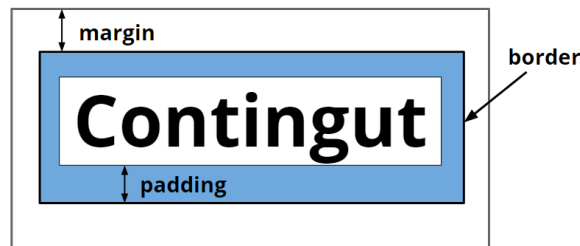


Figura 1: Model de caixa

- **Margin (marge):** distància des del border fins l'element en què està contingut l'objecte.
- **Border (vora):** línia que separa el margin del padding.
- **Padding (farcit):** distància entre el border i el contingut.
- **Contingut:** text, imatge, taula, etc.

L'**ordre de visualització** des del punt de vista de l'usuari és:

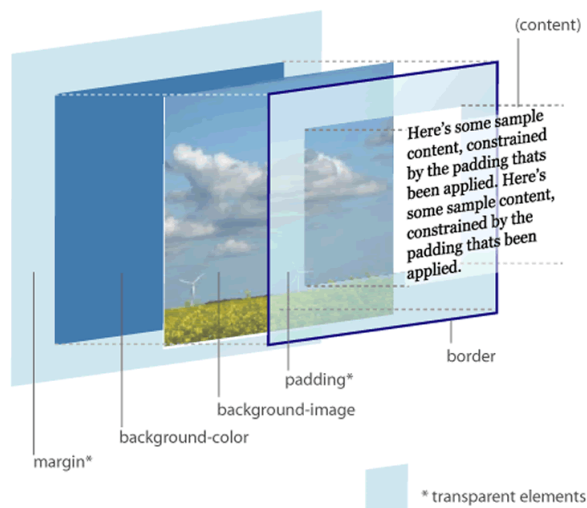


Figura 2: Ordre de renderitzat

1.1 Amplada i alçada

Els atributs **width** i **height** permeten definir la mida de la caixa.

```
h1 {  
  width: 300px;  
  height: 100px;  
}
```

Important: quan configurem les propietats `width` i `height` d'un element amb CSS, estem establint l'amplada i l'alçada de l'àrea de contingut. Per calcular la mida completa d'un element, també hem d'afegir farcit, vores i marges.

- **Amplada total** de l'element = `width` + farciment esquerre + farciment dret + vora esquerra + vora dreta + marge esquerre + marge dret
- **Alçada total** de l'element = `height` + farciment superior + farciment inferior + vora superior + vora inferior + marge superior + marge inferior

Si usem **`max-width`** en lloc de `width`, l'element usa l'ample especificat, però si no cap en l'ample de la finestra, es redimensiona a aquesta: [exemple de max-width](#).

1.2 Margin i padding

Per defecte els valors de margin, padding i border estan a zero.

Podem canviar-ne els valors amb les propietats:

- *`margin-left`, `margin-right`, `margin-top`, `margin-bottom`*: estableixen els marges esquerre, dret, superior i inferior de l'element, respectivament.
- *`margin`*: estableix els 4 marges en aquest ordre: top, right, bottom, left. També es pot utilitzar amb 3 valors (top, right-left, bottom), 2 valors (top-bottom, right-left) o un valor:

```
/* top=25px, right=50px, bottom=75px, left=100px */  
p {  
  margin: 25px 50px 75px 100px;  
}
```

```
/* top=25px, right=50px, bottom=75px, left=50px */  
p {  
  margin: 25px 50px 75px;  
}
```

```
/* top=25px, right=50px, bottom=25px, left=50px */  
p {  
  margin: 25px 50px;  
}
```

```
/* top=25px, right=25px, bottom=25px, left=25px */
p {
  margin: 25px;
}
```

Padding: funciona igual que *margin*, amb les propietats *padding-left*, *padding-right*, *padding-top*, *padding-bottom* i *padding*.

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}

div {
  padding: 25px 50px 75px 100px;
}
```

Es pot establir la propietat de *margin* a **auto** per centrar horitzontalment l'element dins del seu contenidor. L'element ocuparà l'amplada especificada i l'espai restant es dividirà a parts iguals entre els marges esquerre i dret.

```
div {
  width: 300px;
  margin: auto;
}
```

1.3 Border

La propietat **border** serveix per a dibuixar una vora entre el padding i el margin.

Amb el border hi ha moltes possibilitats:

- **border-style:** pot ser none (sense vora), solid (línia continua), dashed (línia discontinua), dotted (línia de punts), double (doble), groove, ridge, inset, outset...
- **border-width:** especifiquem l'amplada. El més corrent és fer-ho amb píxels (Per exemple: 10px)
- **border-color:** el color que tindrà.
- **border:** s'especifica amplada, estil i color, en aquest ordre.

```
p { border-color: #000000; border-style:solid; }
```

```
p { border: 2px solid #000000; }
```

[Més exemples de border.](#)

1.4 Vores arrodonides

La propietat **border-radius** serveix per a arrodonir les vores:

```
p {  
  border: 2px solid red;  
  border-radius: 5px;  
}
```

1.5 Propietat box-sizing

Des dels inicis del CSS, el box model (model de caixa) ha funcionat així:

width real visible d'un element de caixa = width + padding + border + margin

height real visible d'un element de caixa = height + padding + border + margin

```
div {  
  width: 310px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 5px;  
}
```

width real =
310px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 10px (left + right margin)
= 350px

Figura 3: Ample real

Això provoca que l'amplada varie molt d'uns elements als altres, de manera que s'ha creat una nova propietat CSS3 anomenada **box-sizing**.

Quan s'estableix **box-sizing: border-box** sobre un element, el padding i el border d'aquest element ja no fa augmentar la seva amplada ni alçada total.

```
* { box-sizing: border-box; }
```

Aquesta regla assegura que tots els elements sempre estan dimensionats d'una manera més intuïtiva.

```
div {  
  box-sizing: border-box;  
  width: 200px;  
  padding: 20px;  
  border: 1px solid #DDD;  
}
```

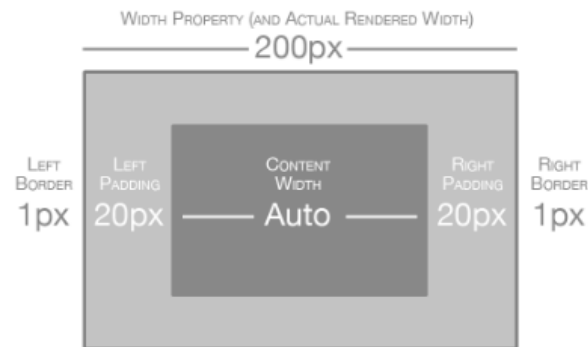


Figura 4: Box sizing

[Exemples](#)

2 Elements de bloc i en línia

Les caixes es poden comportar de formes diferents respecte a les altres.

El més corrent són dos comportaments:

- **block**: Són posicionats verticalment. Blocs de contingut que ocupen tot l'espai horitzontal. El final del bloc provoca un salt de línia.
- **inline**: Són posicionats horitzontalment. Les caixes permeten que les altres es posen al seu costat.

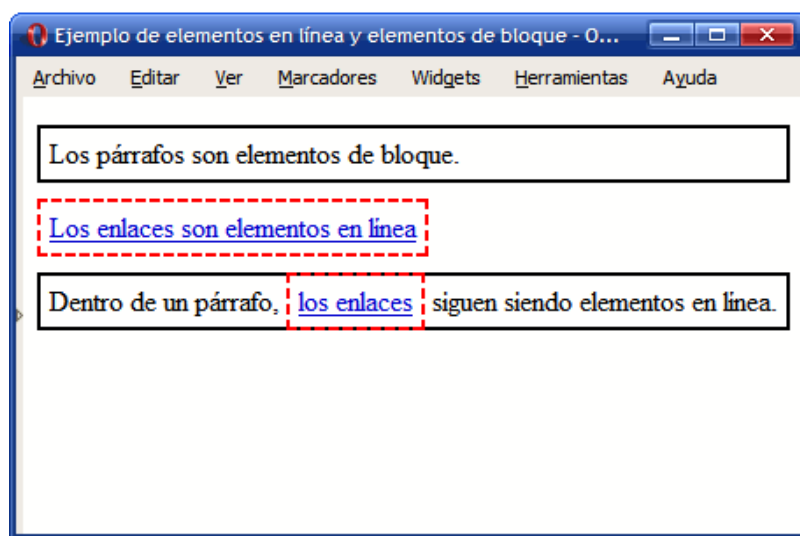


Figura 5: Element en bloc i en línia

- El primer element és un paràgraf que ocupa tota la línia perquè és un element de block.
- El segon element és un enllaç que ocupa només l'espai necessari pel seu contingut ja que és un element inline.

Podem canviar la disposició amb la propietat **display**:

- **display: inline**: mostra un element com si fora en línia.
- **display: block**: mostra un element com si fora de bloc.
- **display: inline-block**: mostra un element en línia, però permet afegir-li width i height.

```
li {  
  display: inline; /* Mostra els li uns al costat dels altres */  
}
```



```
a {  
  display: block; /* Mostra els a uns baix dels altres */  
}  
  
li {  
  display: inline-block; /* Mostra els li uns al costat dels altres */  
  width: 2px; /* amb 2px d'ample */  
}
```

2.1 Visibilitat dels elements

També podem usar **display: none** per a amagar contingut:

```
h1.hidden {  
  display: none;  
}
```

Amb aquesta propietat, l'element no es veu i s'allibera l'espai que ocupa.

Amb la propietat **visibility: hidden** també podem amagar un element, però es manté l'espai que ocupa.

```
h1.hidden {  
  visibility: hidden;  
}
```

Per a tornar a mostrar els elements, establim les propietats a **initial**:

```
h1 {  
  display: initial;  
}  
  
h2 {  
  visibility: initial;  
}
```

2.2 Crear un menú de navegació amb CSS

Amb el posicionament i altres propietats CSS podem fer un menú de navegació per al nostre lloc web.

Es recomana tindre els enllaços dins d'una **llista sense ordre**, la qual ha d'estar dins d'un element `nav` o `div` identificat amb un `id`:

```
<nav id="menu">
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="news.html">News</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

Primer, eliminem els marcadors, *margins* i *padding*s de la llista:

```
nav#menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
```

També eliminem el subratllat dels enllaços:

```
nav#menu li a {
  text-decoration: none;
}
```

El següent pas serà canviar el display dels **li** a **inline-block** i els dels enllaços a **block**:

```
nav#menu li {
  display: inline-block;
}
```

```
nav#menu li a {
  display: block;
  text-decoration: none;
}
```

Per a finalitzar, canviem els colors, alineacions i padding's dels elements. Altres millores interessants són utilitzar el selector **a:hover** per a canviar els colors quan passem per damunt d'un element o usar un selector de classe per a mostrar d'un color diferent l'element del menú en el qual ens trobem:

```
<style>
nav#menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  background-color: #333;
  height: 50px;
```

```
}

nav#menu li {
  display: inline-block;
}

nav#menu li a {
  display: block;
  color: white;
  text-align: center;
  padding: 15px;
  text-decoration: none;
}

nav#menu li a:hover {
  background-color: #111;
}

.active {
  background-color: #04AA6D;
}
</style>

...

<nav id="menu">
  <ul>
    <li><a class="active" href="#">Home</a></li>
    <li><a href="news.html">News</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```



Figura 6: Menú formatat amb CSS

Més exemples de menús.

2.3 Referències

- [W3Schools](#)
- [Mozilla](#)
- [W3C CSS specifications](#)