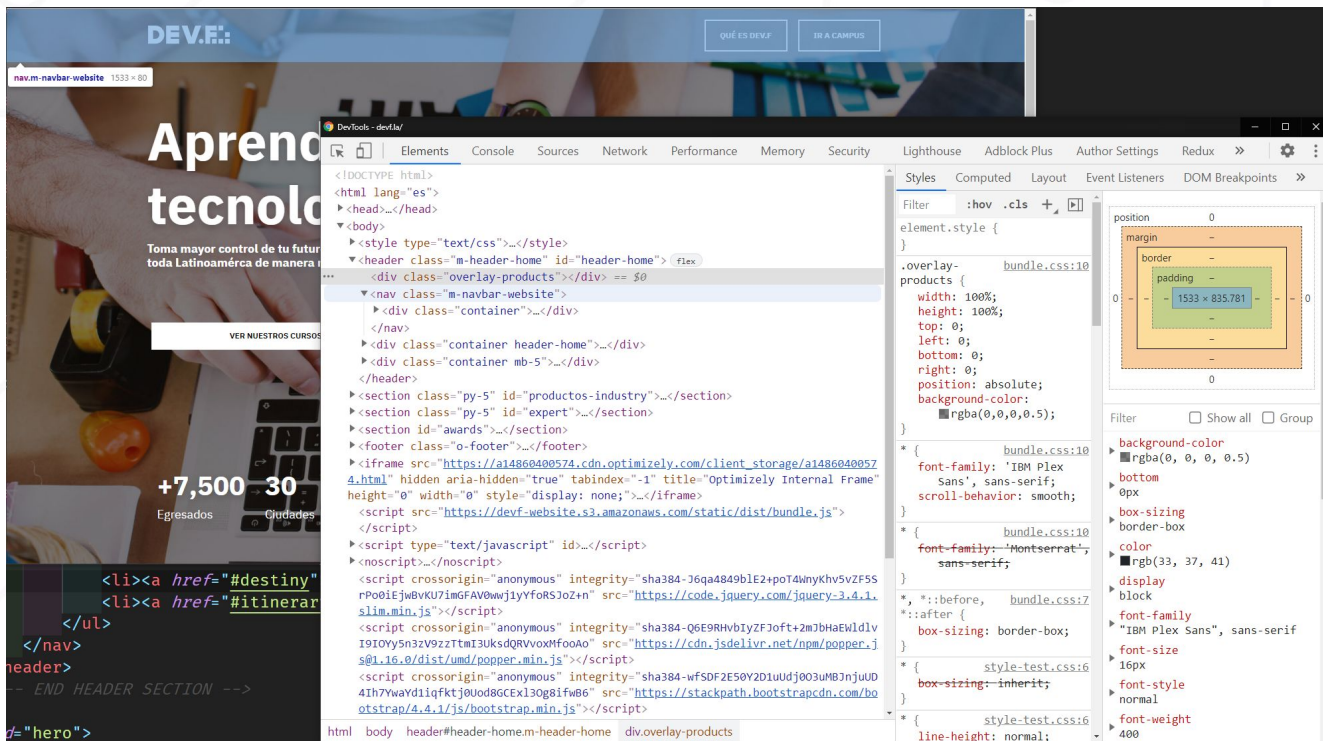


Introducción a CSS: Box Model, Divs y Display

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Inspector de Elementos

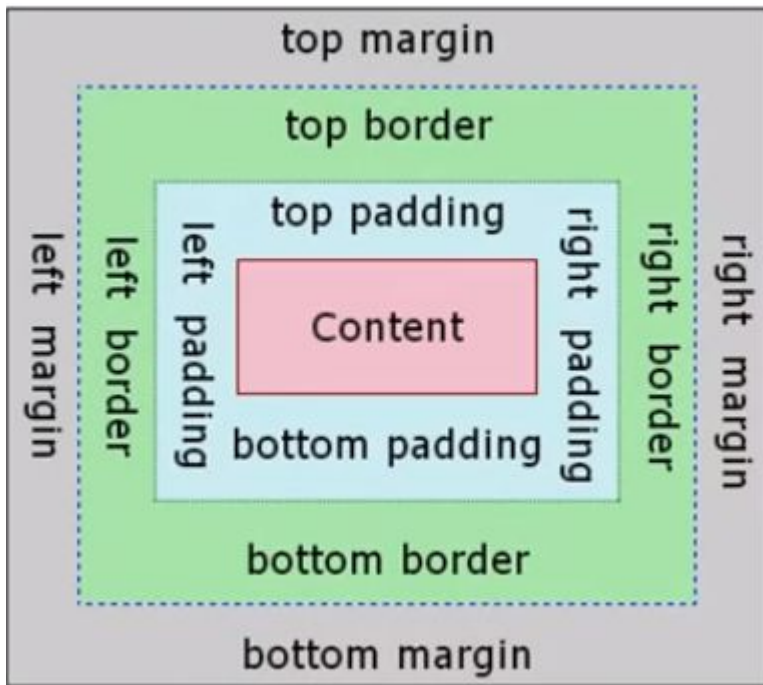


Atajo: Ctrl + Mayus + i

Box Model

DEV.F
DESARROLLAMOS(PERSONAS);

dev



Box Model

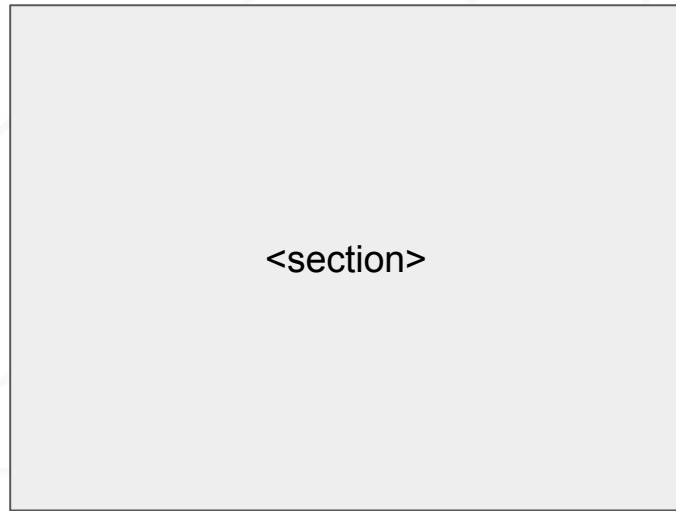
Todo navegador renderiza cada elemento de la página en forma rectangular.

El modelo de caja de CSS se refiere básicamente a las propiedades que tiene cada elemento rectangular:

1. **Content:** El contenido en sí
2. **Padding:** Relleno / Espacio Interior
3. **Border:** La línea del borde
4. **Margin:** Margen / Espacio Exterior

Width y Height

width: 800px



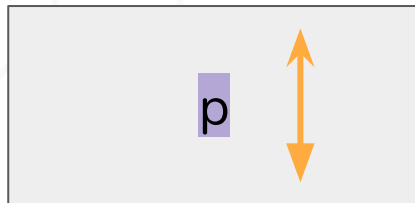
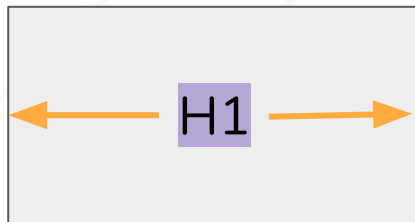
<section>

height: 600px

Padding

El padding se refiere al **espacio interior** del modelo de caja.

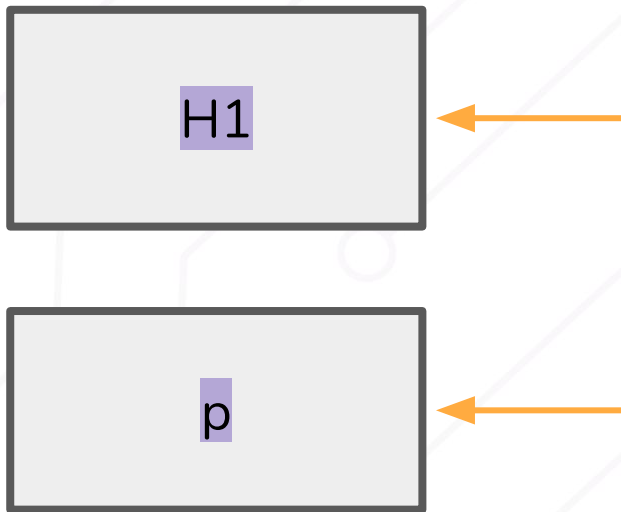
Sirve comúnmente para crear un espacio entre el contenido y el borde de la caja (cuando se utilizan bordes y fondos)



Borders

Borders se refiere literalmente al borde de la caja, y se coloca al “final” del padding.

Sirve mayormente con fines estéticos o para separar elementos visualmente.



Borders

solid

`border: black 5px solid;`

dashed

`border: black 5px dashed;`

dotted

`border: black 5px dotted;`

double

groove - 3D border dependent on border color

ridge - 3D border dependent on border color

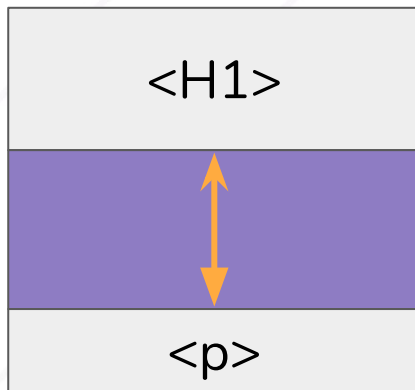
inset - 3D inset border dependent on border color

outset - 3D outset border dependent on border color

Margin

El margen se refiere al **espacio exterior** del modelo de caja.

Sirve para distanciar una caja de otra.



Maneras de Usar el Box Model

Funciona con: margin, padding, border.

```
p {  
  
  margin-top: 5px;  
  margin-bottom: 5px;  
  margin-right: 10px;  
  margin-left: 10px;  
  
}
```

```
3 p {  
4   padding: 10px 5px 5px 10px;  
5 }
```

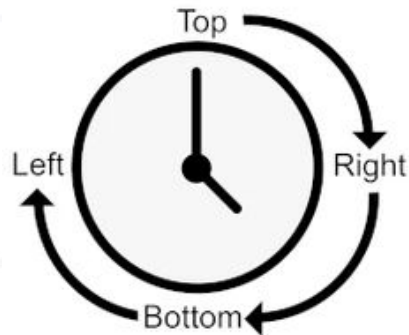
↓ ↓ ↓ ↓
TOP RIGHT BOTTOM LEFT

```
3 p {  
4   padding: 10px 5px 10px;  
5 }
```

↓ ↓ ↓
TOP RIGHT and LEFT BOTTOM

```
3 p {  
4   padding: 8px 12px;  
5 }
```

↓ ↓
TOP and BOTTOM RIGHT and LEFT



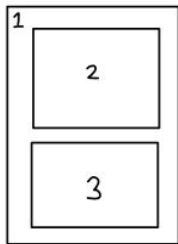
Resumen: Box Model (Modelo de Caja)

JULIA EVANS
@bork

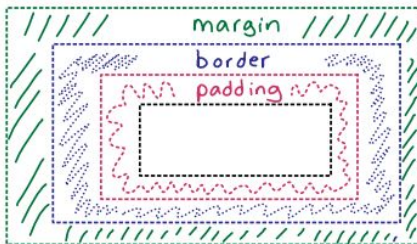
the box model

every HTML element
is in a box

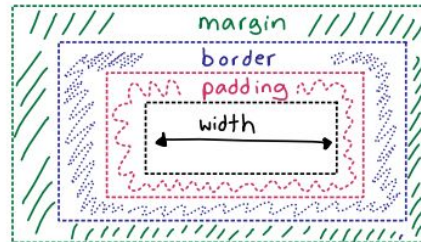
```
<div class="1">  
  <div class="2" />  
  <div class="3" />  
</div>
```



boxes have **padding**,
borders, and a **margin**



width doesn't include
margin/border/padding
by default



JULIA EVANS
@b0rk

centering in CSS

center text with
text-align

```
h2 {  
  text-align: center;  
}
```

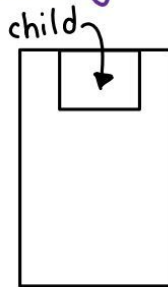
center block elements
with margin: auto

example HTML:

```
<div class="parent">  
  <div class="child">  
  </div>  
</div>
```

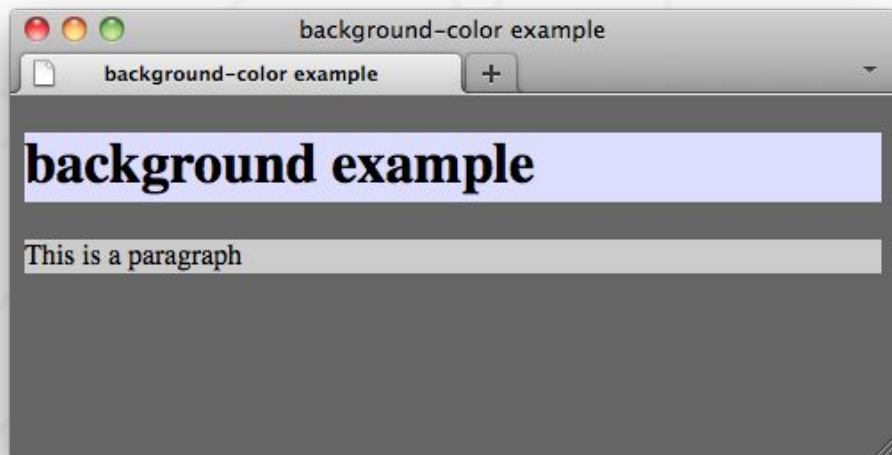
margin: auto
centers horizontally

```
.child {  
  width: 400px;  
  margin: auto;  
}
```



Referencia: Twitter de [Julia Evans](#)

Backgrounds



Algunas propiedades de background

background-image: url("mi-imagen.jpg");
background-repeat: no-repeat;
background-position: center;
background-attachment: fixed;
background-size: 100px 100px;
background: linear-gradient(to right,
green, yellow, orange, red)

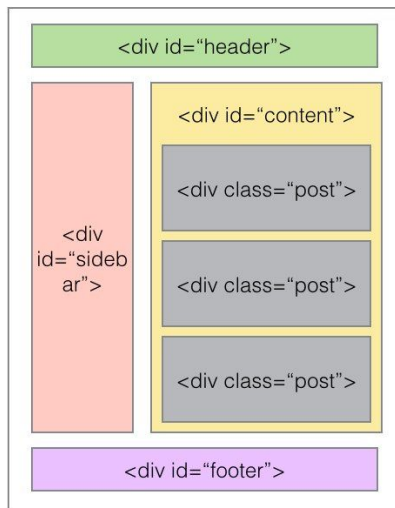
Otros temas

DEV.F
DESARROLLAMOS(PERSONAS);

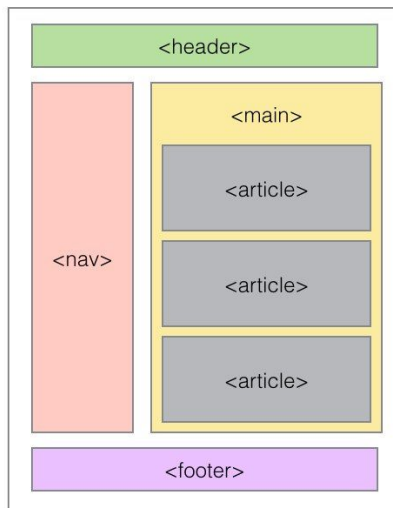
dev

HTML4 vs HTML5 Page Structure on a Blog

HTML4: Lots of Classes/IDs



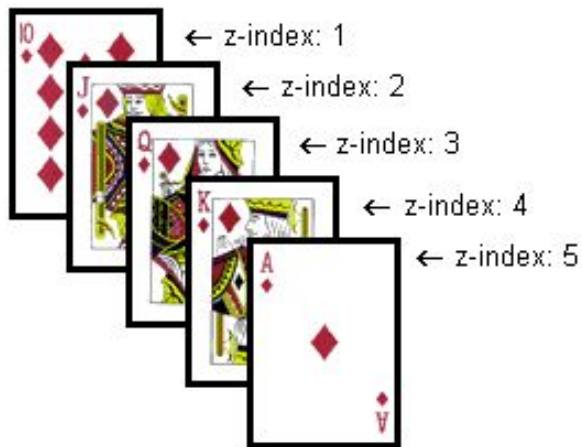
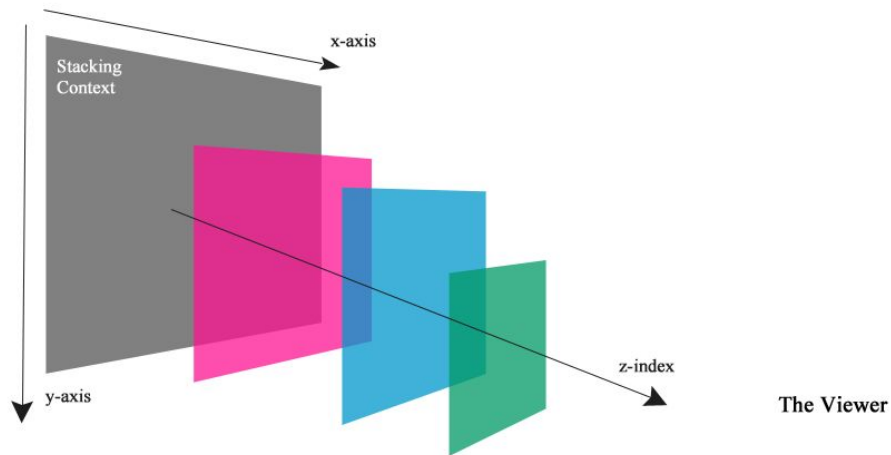
HTML5: Semantic Tags/Sections



Div (division)

Es un **contenedor genérico** sin un **significado semántico** en particular. Sirve para crear secciones o agrupar contenidos con fines mayormente estéticos (donde usamos atributos como ***class*** y/o ***id*** para ello).

Es buena práctica solo usarlo cuando existan casos en que no corresponda usar etiquetas como: **`<main>`**, **`<header>`**, **`<footer>`**, **`<nav>`**, **`<article>`** ó **`<section>`**.

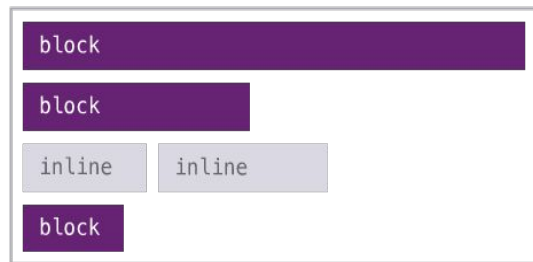


z-index

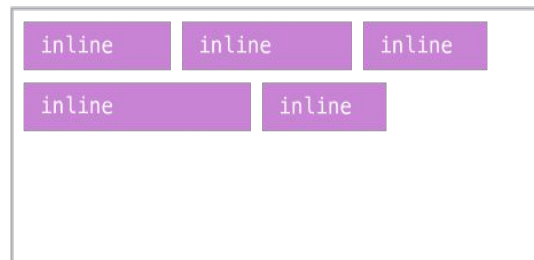
Sirve para indicar qué elemento iría encima y cual debajo en caso de que 2 elementos se superpongan, como si el documento tuviera profundidad.

Cosas interesantes saber:

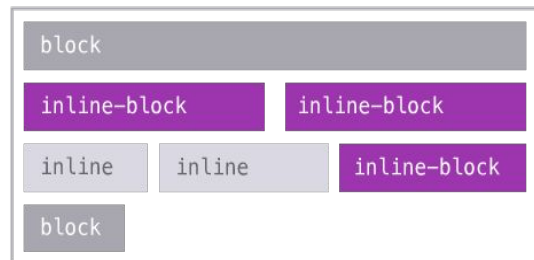
- Por defecto, todos los elementos tienen asignado un valor de z-index es **auto**, que corresponde a **0**.
- Un valor mayor hará que este se coloque delante de elementos con valores menores.
- Es posible usar valores en negativo.



✓ padding ✓ margin ✓ width



✗ padding ✗ margin ✗ width



✓ padding ✓ margin ✓ width

Imágenes cortesía de [CSS Layout Cheat Sheet](#)

Display

Esta propiedad nos ayuda a indicar el comportamiento de las “cajas” con respecto a otras “cajas”.

Las propiedades básicas de display son:

block: hace que el comportamiento del elemento sea como un bloque. Desplazando el siguiente bloque a la siguiente línea.

inline: el elemento se renderizará en línea con otros elementos (se coloca uno al lado de otro).

inline-block

Display en la práctica

block vs inline vs inline-block

Below are a bunch of `<div style="width: 50px"...>` with different `display` settings.

**display:
block**

display:
block

display:
block

display:
block

display: inline

display: inline display:
inline display: inline

**display: inline-
block**

display: inline-
block

display: inline-
block

display: inline-
block

inline vs block

HTML elements default
to **inline** or **block**

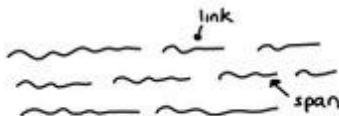
example block
elements

```
<p> <div>  
<ol> <ul> <li>  
<h1> - <h6>  
<table> <form>  
<article> <nav>
```

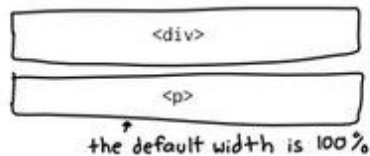
example inline
elements

```
<a> <span>  
<strong> <i>  
<button> <input>  
<small> <abbr>  
<textarea>
```

inline elements are
laid out horizontally



block elements are laid
out vertically by default



inline elements
ignore width & height

Setting the width is impossible,
but you can use line-height
to change the height

also, inline elements ignore
the vertical padding of other
inline elements

display can force an
element to be inline or block

display determines 2 things:

- ① whether the element itself is
inline, block, inline-block, etc
- ② how child elements are laid out
(grid, flex, table, default, etc)

display: inline-block;

inline-block makes a block
element that's laid out
horizontally like an inline element

