

# Intro a HTML

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

HTML

# Intro a HTML

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# ¿Qué es HTML?



**H**yper  
**T**ext  
**M**arkup  
**L**anguage

# Estructura de un Programa HTML


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset=UTF-8>
    <title>Hello World!</title>
  </head>
  <body>
  </body>
</html>
```



# En código

```
<!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">
  <title>Bienvenido a Devf</title>
</head>
<body>
  <h1>Aprende el superpoder de la tecnología</h1>
  <p>América Latina necesita más coders!</p>
</body>
</html>
```

# W3Schools y SoloLearn

[Tutorials](#) [References](#) [Exercises](#)

[Spaces](#) [Paid Courses](#) [Log in](#)

# HTML

The language for building web pages

[Learn HTML](#)[HTML Reference](#)

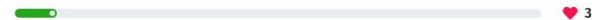
## HTML Example:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

[Try it Yourself](#)



Like

Many HTML elements require both opening and closing tags.

```
<button>Like</button>
```

HTML

Copy

Complete the line to code a Like button

```
  </button>
```

Like

```
<button>
```

Back

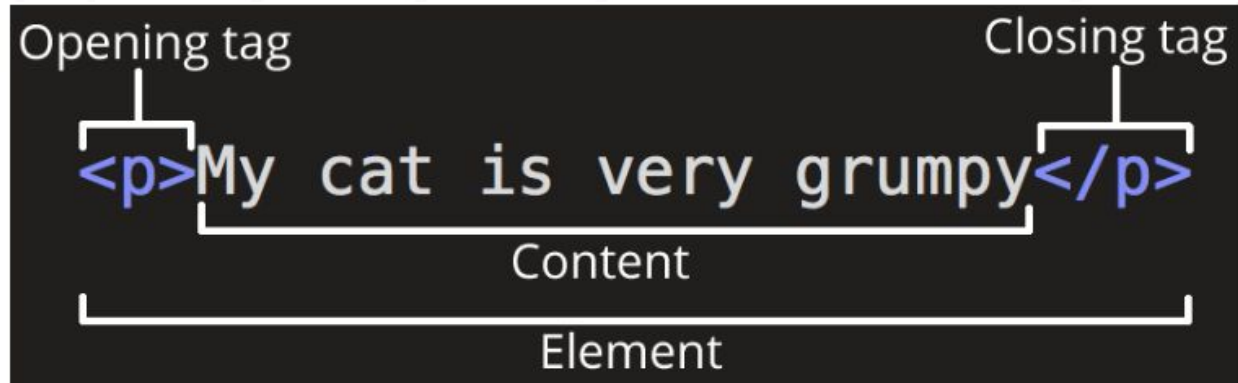


Check

DEV.F

# Sintaxis

- **Sintaxis** son las **reglas de escritura** de un lenguaje asociado a programación.
- **Html** es un **lenguaje de marcado** (etiquetas).





# Atributos

Proveen información adicional sobre el elemento.

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

```

```

# Accesibilidad

## No hacer

```
<h1>Cabecera nivel 1</h1>  
<h3>Cabecera nivel 3</h3>  
<h4>Cabecera nivel 4</h4>
```

## Sí hacer

```
<h1>Cabecera nivel 1</h1>  
<h2>Cabecera nivel 2</h2>  
<h3>Cabecera nivel 3</h3>
```

Una forma típica de navegación que utilizan las **personas no videntes** es moverse a través de la pantalla con las etiquetas de encabezado. De esta forma, pueden conocer rápidamente el contenido de la página en poco tiempo.

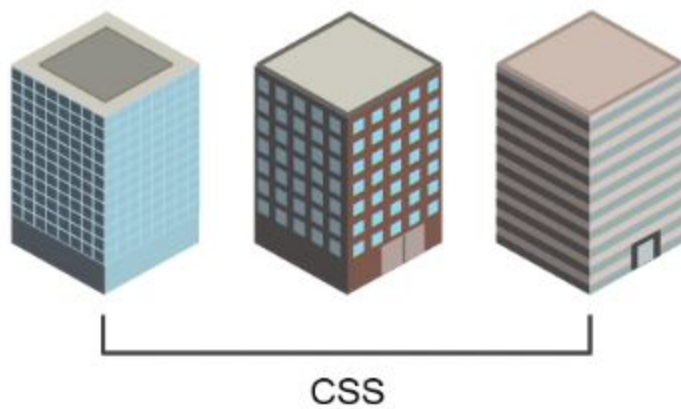
The logo consists of the text 'DEV.F!' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its top right corner. The background is a dark blue diamond shape.

# DEV.F!

## Actividad

Repasemos cómo crear una página personal empleando:

- h1 - h6 (headings)
- p (paragraphs)
- span
- br (break) (aunque sea mala práctica)
- hr (horizontal row)
- img (image)
- Table
- Listas ordenadas
- Listas desordenadas



# Atributo "style"

```
<tagname style="property:value;">
```

```
<body style="background-color:powderblue;">
```

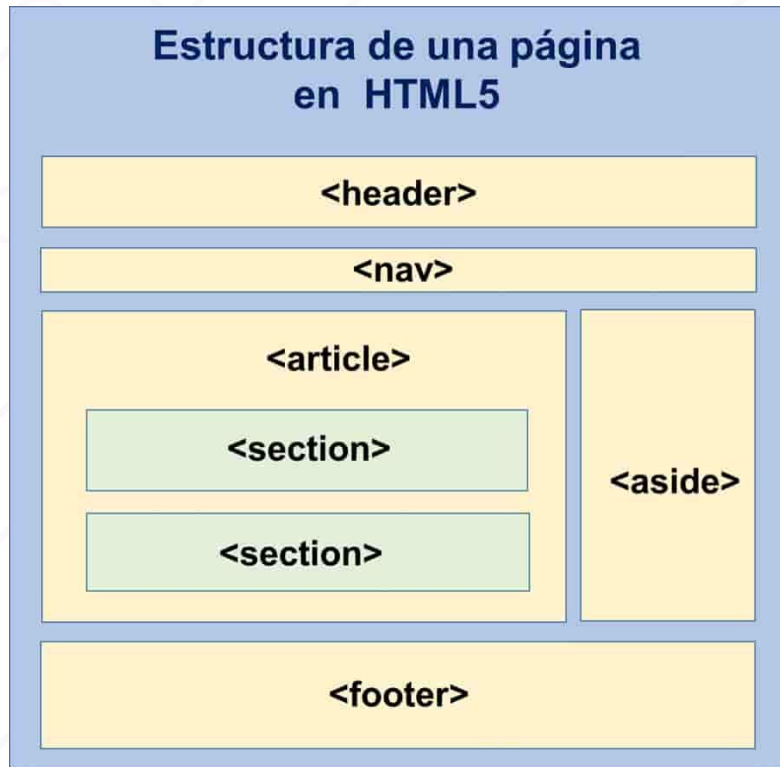
```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

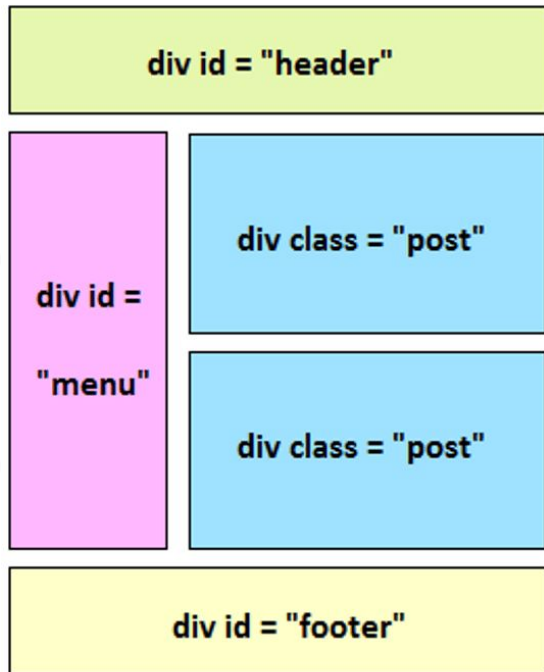
# Estructura básica

## Estructura de una página en HTML5



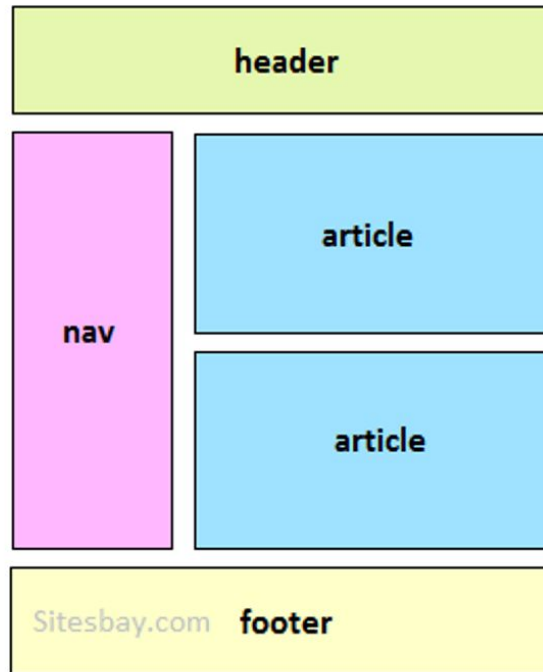
# Estructura básica

Html4

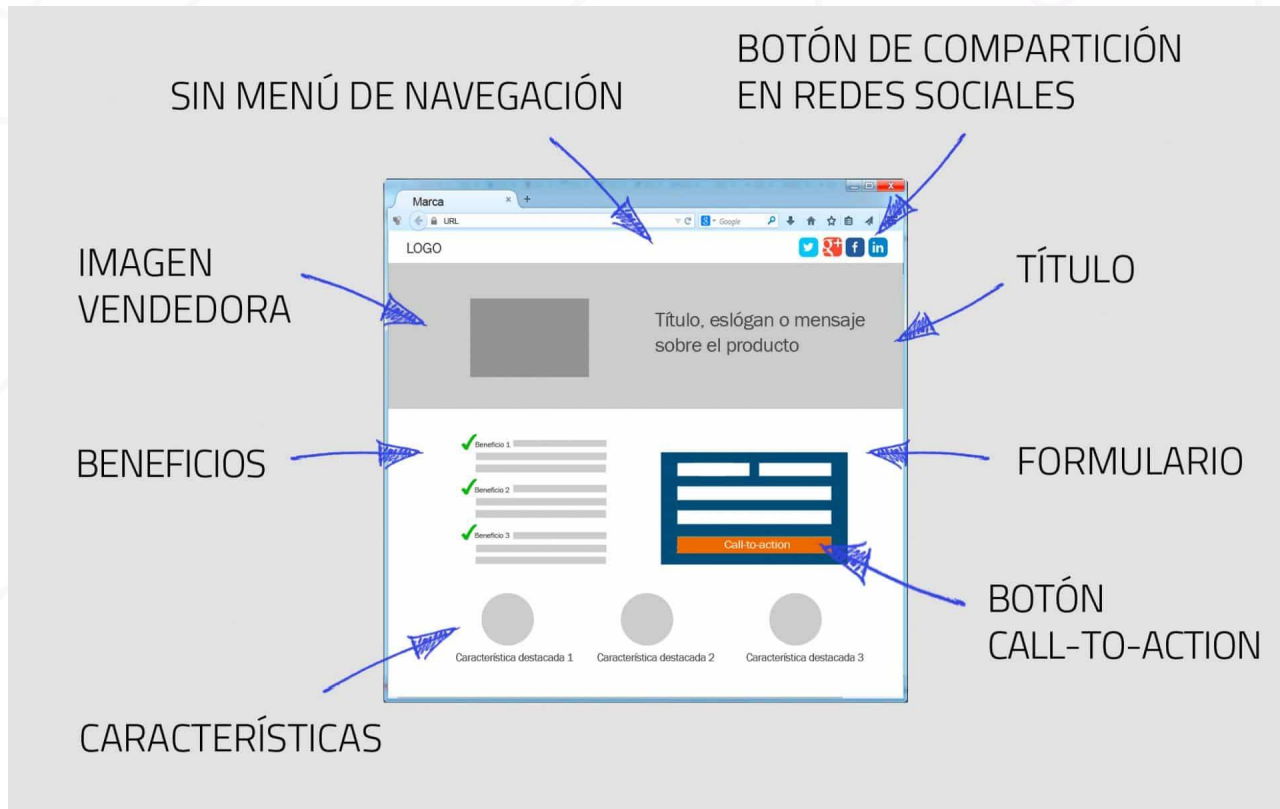


VS

Html5



# Estructura de una landing page agresiva





# Cómo ordenar los atributos de los elementos html

El objetivo de ordenar los atributos es **facilitar la lectura del código**, es seguir una buena práctica o tener un código más limpio.



# Orden de atributos

## Orden de atributos

Los atributos HTML deben venir en este orden en particular para facilitar la lectura del código.

- `class`
- `id`, `name`
- `data-*`
- `src`, ..., `for` `type` `href` `value`
- `title`, `alt`
- `role`, `aria-*`

Las clases son excelentes componentes reutilizables, por lo que son lo primero. Los ids son más específicos y deben utilizarse con moderación (por ejemplo, para marcadores en la página), por lo que son segundos.

```
<a class="..." id="..." data-toggle="modal" href="#">
```

Example link

```
</a>
```

```
<input class="form-control" type="text">
```

```

```

# Introducción a CSS

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev



**CSS**



# ¿Qué es CSS?

*Cascading* **Style** **Sheets**



Describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o cómo va a ser pronunciada la información presente a través de un dispositivo de lectura. Es decir, ofrece el **control total** sobre estilo y formato de sus documentos.

## CARACTERÍSTICAS

### DISEÑO

Definir fuentes, colores, estilos, disposición de elementos

### RESPONSIVO

Adaptar la pantalla a diferentes dispositivos

### INDEPENDENCIA

Facilidad de modificar estilos sin tocar el HTML

### MANTENIMIENTO

Modificar múltiples páginas a través de un único archivo CSS

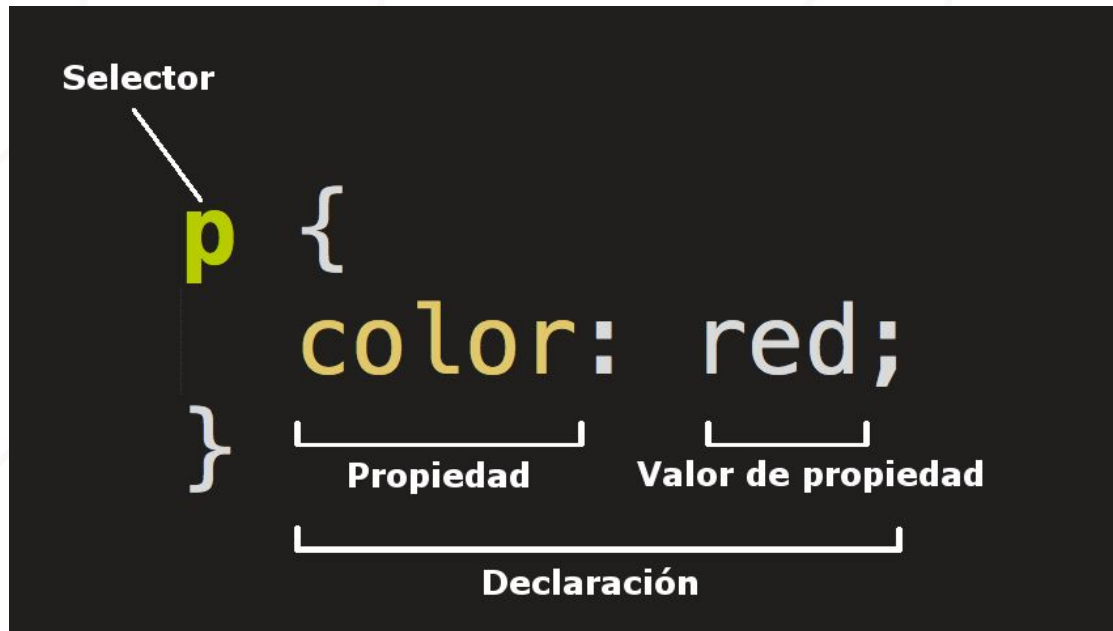
# Sintaxis básica

Consiste en uno o más **SELECTORES** (¿QUIEN? - sujeto)

¿H1?, ¿P?, ¿BODY?, ¿id?

seguido de una **DECLARACIÓN** (¿QUÉ? - predicado)

{ propiedad : valor; }



¿Y cómo lo uso?

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# CSS puede usarse de 3 formas

1

EN EL HTML (INTERNO)  
ENTRE EL <HEAD> Y </HEAD>

```
<head>
  <style type="text/css">
    h1{
      color: red;
      font-size: 24px;
    }
  </style>
</head>
```

2

INLINE (EN LÍNEA)  
CON EL HTML

```
<body>
  <h1 style="color: red;
  font-size: 24px;">
    MI TITULO CON ESTILO
  </h1>
</body>
```

3

EXTERNO  
REFERENCIADO EN EL  
<HEAD> Y </HEAD>

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

↓  
Archivo con  
extensión .CSS



Atajo VS Code: link



The logo consists of the text 'DEV.F.' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its base. The logo is centered within a dark blue diamond shape.

**DEV.F.**

*DISCUSIÓN*

*¿CUÁL SERÍA LA  
FORMA MÁS  
RECOMENDABLE  
DE USAR CSS?*

# Recomendaciones de Uso de CSS

1

EN EL HTML (INTERNO)  
ENTRE EL <HEAD> Y </HEAD>

```
<head>
  <style type="text/css">
    h1{
      color: red;
      font-size: 24px;
    }
  </style>
</head>
```



2

INLINE (EN LÍNEA)  
CON EL HTML

```
<body>
  <h1 style="color: red;
  font-size: 24px;">
    MI TITULO CON ESTILO
  </h1>
</body>
```



3

EXTERNO  
REFERENCIADO EN EL  
<HEAD> Y </HEAD>

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

Archivo con  
extensión .CSS



Nota: Esto hablando de HTML con CSS puro, puede variar dependiendo la tecnología.

# Hablemos de Selectores

(El ¿Quién?)

**DEV.FX**  
DESARROLLAMOS(PERSONAS);

dev

En CSS...

```
body{  
  height: 100%;  
  width: 100%;  
}
```

← Por Etiqueta

```
.class{  
  background-color: ■aquamarine;  
  border: solid □black 1px;  
  width: 80%;  
}
```

← Por clase

```
#id {  
  margin: 2px;  
  font-size 12px;  
  text-transform: capitalize;  
}
```

← Por id

# Selectores

Un selector CSS es la primera parte de una regla CSS que **indica al navegador qué elementos HTML se seleccionan** para aplicarles dicha regla.

Los más comunes son:

**Etiqueta** < >: Como h1, body, p, etc.

**Clase**: Indicado con el atributo class="mi-clase" en HTML.

**Id**: Indicado con el atributo id="mi-id" en HTML

# Selectores avanzados

```
body .class .other-class{
  height: 100%;
  width: 100%;
}

.class p {
  background-color: ■aquamarine;
  border: solid ■black 1px;
  width: 80%
}

#id .class p{
  margin: 2px
  font-size 12px;
  text-transform: capitalize;
}
```

Se puede hacer combinación de selectores mientras cumpla la regla de que cada miembro esté dentro del miembro anterior ejemplo:

```
<body>
  <div class="class">
    Primer Div
    <div class="other-class">
      Segundo Div
    </div>
  </div>
</body>
```

# Pseudo-clases y Pseudo-elementos

## *PSEUDO-CLASSES*

:hover  
:link  
:active  
:target  
:not(s)  
:focus  
etc.

## *PSEUDO-ELEMENTS*

:first-letter  
:first-line  
:before  
:after  
::selection

```

1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Prueba de Especificidad</title>
9   <style>
10     .main-heading {
11       color: red;
12     }
13
14     h1 {
15       color: blue;
16     }
17   </style>
18 </head>
19
20 <body>
21   <h1 class="main-heading">Hola Dev.F</h1>
22 </body>
23
24 </html>
25

```

Prueba de Especificidad

127.0.0.1:5500/test-master-code-g9/01.IntroALaWeb/CSS-Especificidad/index.html

Hola Dev.F

# Especificidad

La especificidad es el modo que tiene el navegador de **decidir qué regla se aplica** si diversas reglas tienen selectores diferentes pero podrían aplicarse a un mismo elemento.

# ¿Por qué se llaman “en Cascada”?

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev



```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Prueba de Especificidad</title>
8   <style>
9     h1 {
10       color: red;
11     }
12     h1 {
13       color: blue;
14     }
15   </style>
16 </head>
17 <body>
18   <h1>Hola Dev.F</h1>
19 </body>
20 </html>
21

```

Prueba de Especificidad

127.0.0.1:5500/test-master-code-g9/01.IntroALaWeb/CSS-Especificidad/index.html

Hola Dev.F

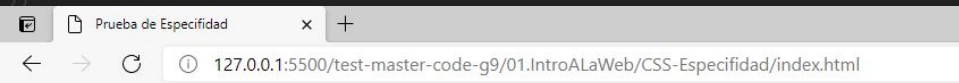
# En CSS: El Orden de los Factores **SÍ** altera el producto

Cuando dos reglas tienen la misma **especificidad**, se aplica la que aparece en último lugar en el CSS.

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Prueba de Especificidad</title>
8   <style>
9     h1 {
10       color: red!important;
11     }
12
13     h1 {
14       color: blue;
15     }
16   </style>
17 </head>
18 <body>
19   <h1 class="main-heading">Hola Dev.F</h1>
20 </body>
21 </html>

```



**Hola Dev.F**

Resultado de la primera regla sin !important



**Hola Dev.F**

Resultado de la primera regla con !important

# Propiedad !important

Se utiliza para convertir una propiedad:valor particular en el elemento más específico, de modo que se invalidan las reglas normales de la cascada. Usar con precaución.

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Prueba de Especificidad</title>
8   <style>
9     body {
10       font-size: 16px;
11       color: green;
12     }
13     h1 {
14       color: blue;
15     }
16     .naranja {
17       color: orange;
18     }
19   </style>
20 </head>
21 <body>
22   <h1 class="main-heading">Hola Dev.F</h1>
23   <p>Independientemente del texto</p>
24   <ul>
25     <li>Como estoy dentro de body</li>
26     <li>Por defecto heredo el color de mi padre</li>
27     <li class="naranja">A menos que utilice una regla más específica</li>
28   </ul>
29 </body>
30 </html>

```

# Herencia

Algunos valores de las propiedades CSS de los **elementos padre** los heredan los **elementos descendientes**.

Por ejemplo, es buena idea definir en body el tamaño, tipo de fuente y color de fuente por defecto.

## Hola Dev.F

Independientemente del texto

- Como estoy dentro de body
- Por defecto heredo el color de mi padre
- A menos que utilice una regla más específica

Fuente: [Mozilla](#) - Cascada y Herencia

DEV.F

The logo for DEV.FL, featuring the text 'DEV.FL' in a bold, white, sans-serif font. The 'F' is stylized with a grid of small squares. The logo is centered within a dark blue diamond shape.

**DEV.FL**

*KATA SELECTORES*

*CSS DINNER*

*<https://flukeout.github.io/>*

dev

# Hablemos de Declaraciones

(El ¿Qué?)

## ¿Cuáles puedo usar?

**DEV.F**  
DESARROLLAMOS(PERSONAS);

# Color y Tipos de Colores en CSS

## *Colors In CSS*

```
body {  
  color: red;  
  background: coral  
}  
  
h1 {  
  color: #00ff00;  
}  
  
p {  
  color: rgb(0,0,255);  
}
```

- Color Names
- HTML5 Color names
- Hexadecimal
- RGB

# Familias de Fuentes



Serif



Sans Serif

Examples	Font Families
font-family	Arial, Helvetica, sans-serif
<b>font-family</b>	<b>"Arial Black", Gadget, sans-serif</b>
font-family	"Bookman Old Style", serif
font-family	"Comic Sans MS", cursive, sans-serif
font-family	Courier, monospace
font-family	"Courier New", Courier, monospace
font-family	Garamond, serif
font-family	Georgia, serif
<b>font-family</b>	<b>Impact, Charcoal, sans-serif</b>
font-family	"Lucida Console", Monaco, monospace
font-family	"Lucida Sans Unicode", "Lucida Grande", sans-serif
font-family	"MS Sans Serif", Geneva, sans-serif
font-family	"MS Serif", "New York", sans-serif
font-family	"Palatino Linotype", "Book Antiqua", Palatino, serif
font-family	Symbol, sans-serif
font-family	Tahoma, Geneva, sans-serif
font-family	"Times New Roman", Times, serif
font-family	"Trebuchet MS", Helvetica, sans-serif
font-family	Verdana, Geneva, sans-serif

# font-family: Familias de Fuentes Web Safe

**La propiedad font-family nos permite escoger la fuente a utilizar.**

Las fuentes Web Safe son aquellas que normalmente vienen instaladas en los Sistemas Operativos por defecto y garantizan una correcta visualización.

- Arial (sans-serif)
- Verdana (sans-serif)
- Helvetica (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

**font-family:** Arial, sans-serif;

**font-family:** Verdana, sans-serif;

**font-family:** Helvetica, sans-serif;

**font-family:** Tahoma, sans-serif;

**font-family:** 'Trebuchet MS', sans-serif;

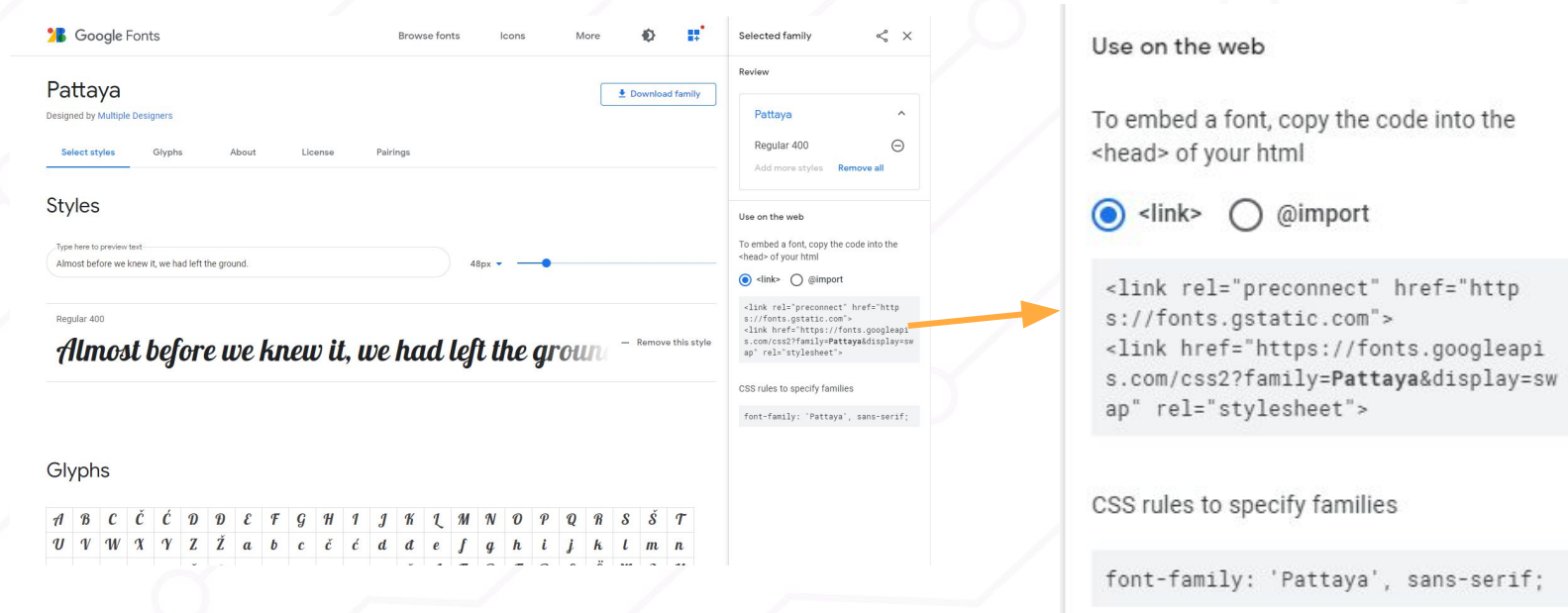
**font-family:** 'Times New Roman', serif;





# font-family: Externas desde Google

También es posible usar alguna fuente en particular que no venga por defecto en el sistema operativo. Un servicio popular es el de Google Fonts.



Google Fonts

Browse fonts Icons More

Selected family

Review

Pattaya

Regular 400

Add more styles Remove all

Use on the web

To embed a font, copy the code into the <head> of your html

☒ <link> ☐ @import

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Pattaya&display=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Pattaya', sans-serif;
```

Use on the web

To embed a font, copy the code into the <head> of your html

☒ <link> ☐ @import

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Pattaya&display=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Pattaya', sans-serif;
```

<https://fonts.google.com/>

## Otras propiedades con font

**font-weight: bold**

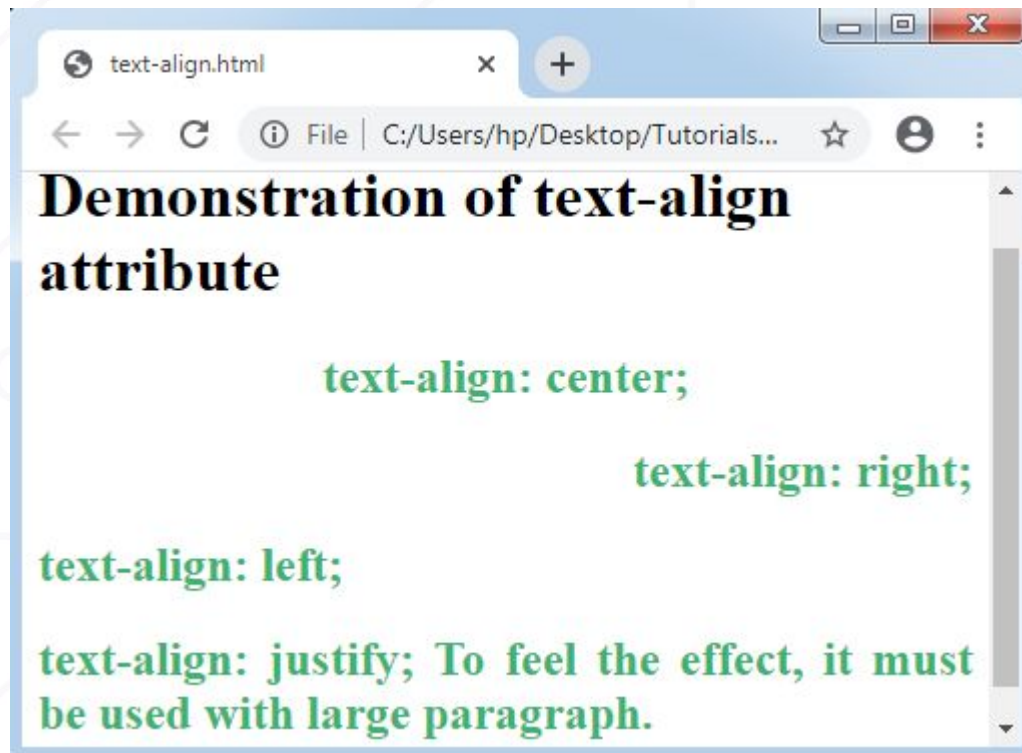
*font-style: italic*

FONT-VARIANT: SMALL-CAPS

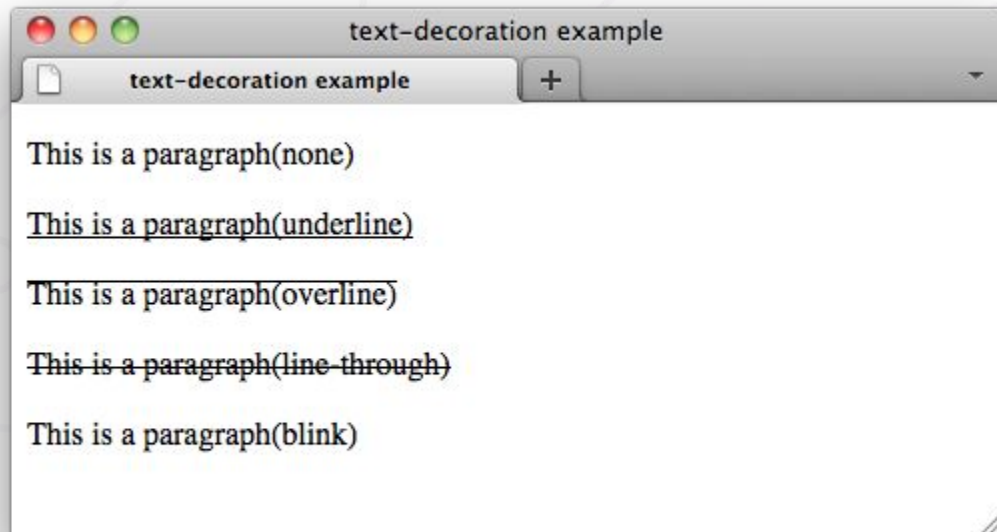
TEXT-TRANSFORM: UPPERCASE

font-size: 40px;

# text-align



# text-decoration



The logo for DEV.F. is displayed in white, bold, sans-serif capital letters. The 'F' is stylized with three small squares at its base. It is centered within a dark blue diamond shape.

**DEV.F.**

*¿PREGUNTAS?*

# Introducción a CSS: Box Model, Divs y Display

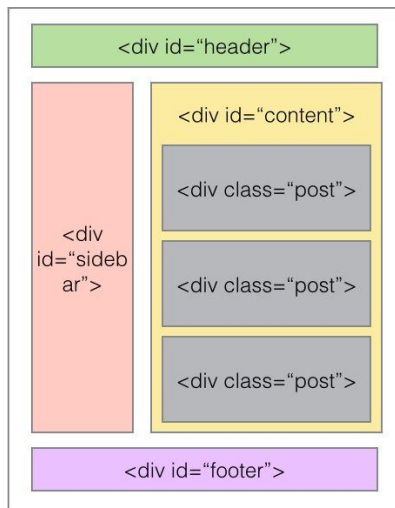
**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

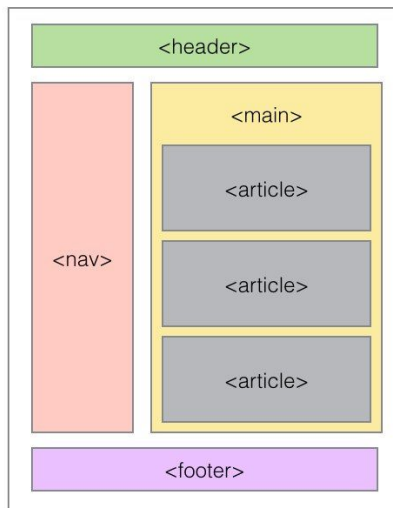
- **Div**
- **Inspector de elementos**
- **Display**
- **Box Model**
- **Centrar Elementos**
- **Backgrounds**
- **z-index**

## HTML4 vs HTML5 Page Structure on a Blog

HTML4: Lots of Classes/IDs



HTML5: Semantic Tags/Sections



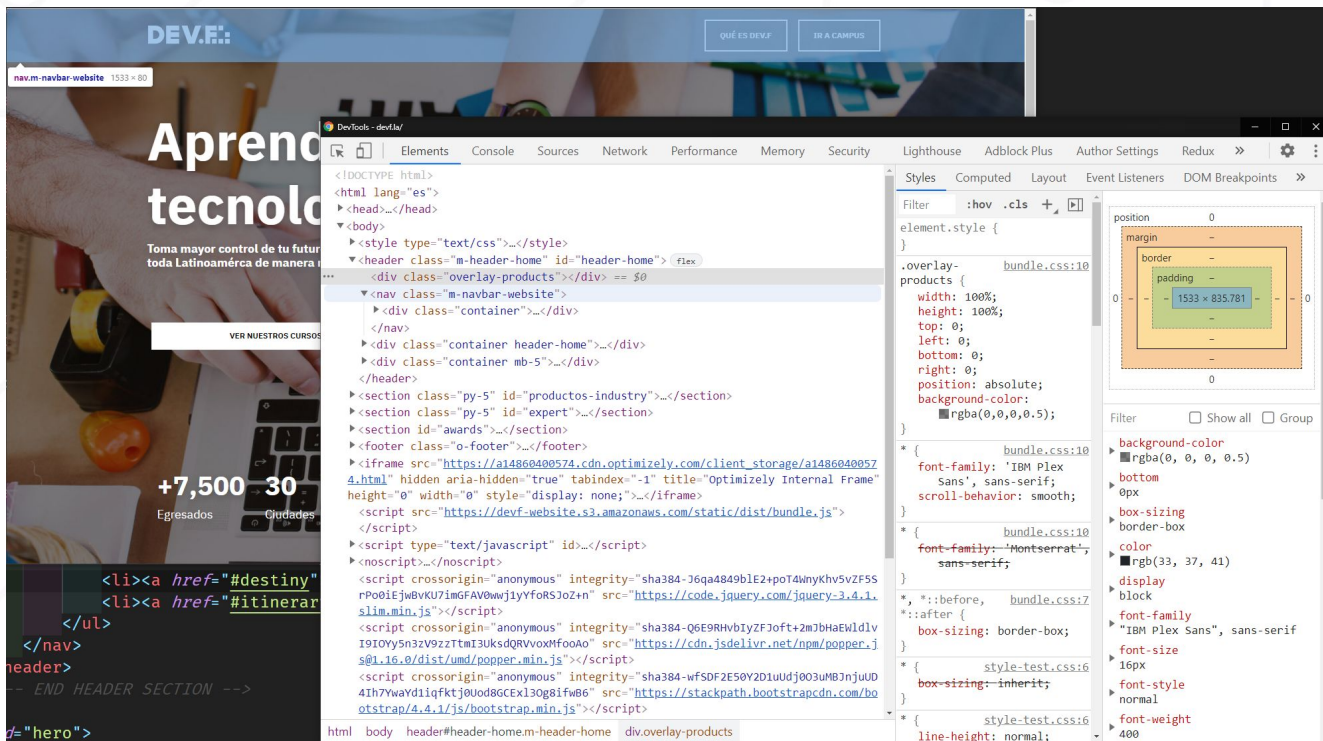
# Div (division)

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>`**.



# Inspector de Elementos



Atajo: Ctrl + Mayus + i

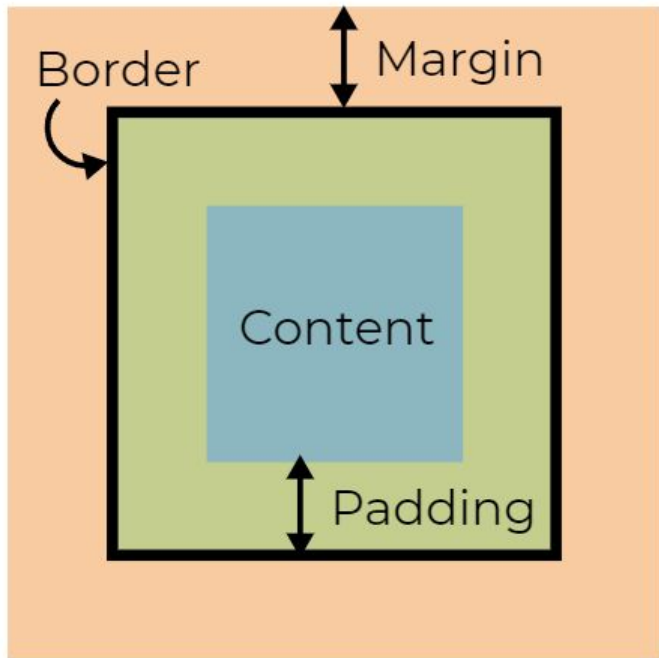
# Box Model

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

Box-sizing: content-box;

Box-sizing: border-box;



## 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

# ¿Tamaño del box?

Contenido

+

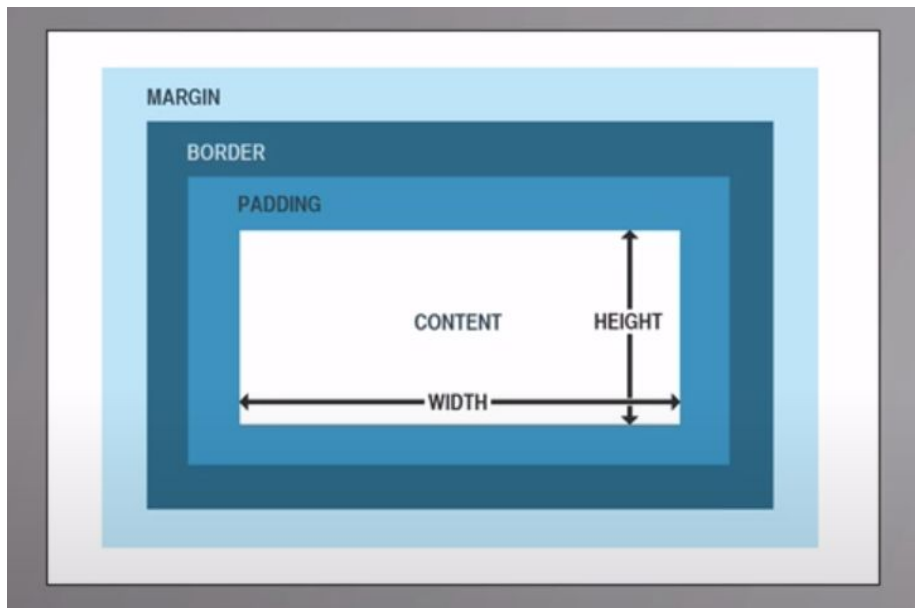
Padding

+

Border

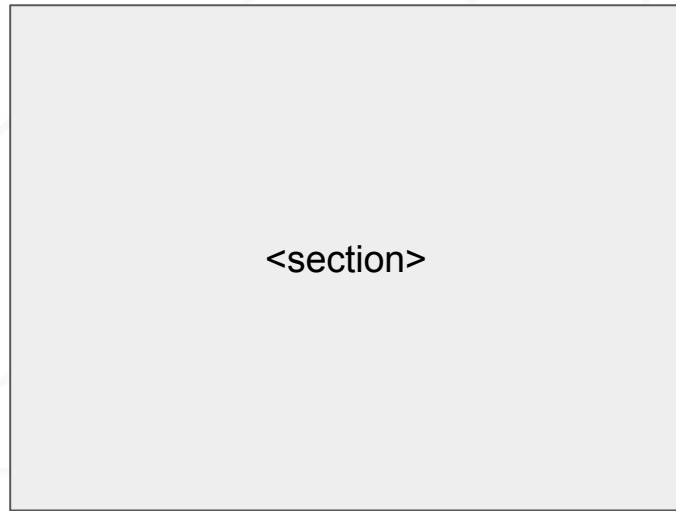
+

Margin



# Width y Height

width: 800px

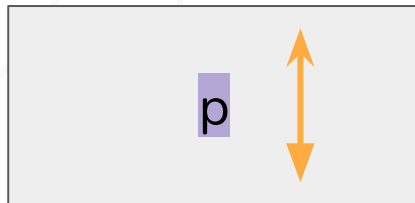
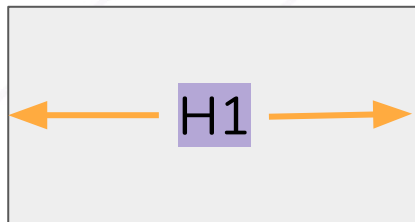


<section>

height: 600px

# Padding

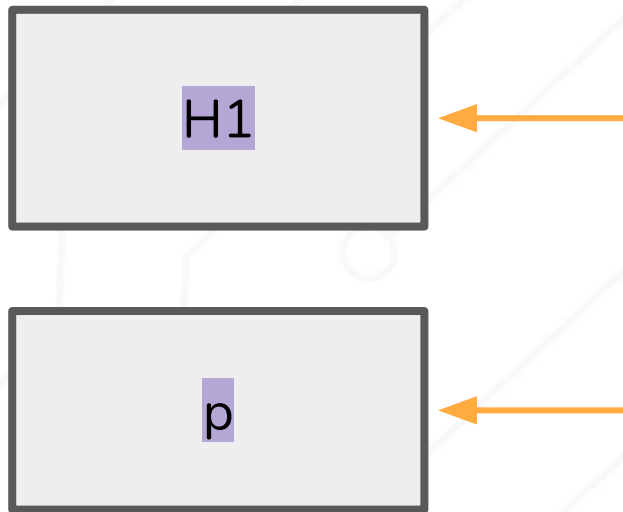
El padding se refiere al **espacio interior** del modelo de caja. Sirve comúnmente para crear un espacio entre el contenido y el borde de la caja (cuando se utilizan bordes y fondos)



# 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**

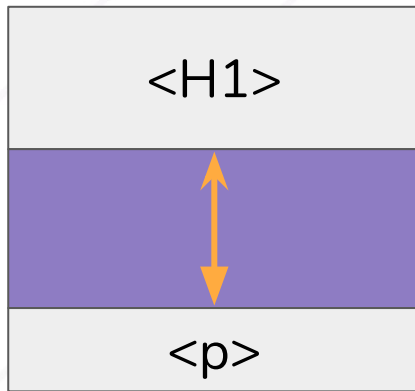
**inset - 3D inset border dependent on border color**

**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 }
```

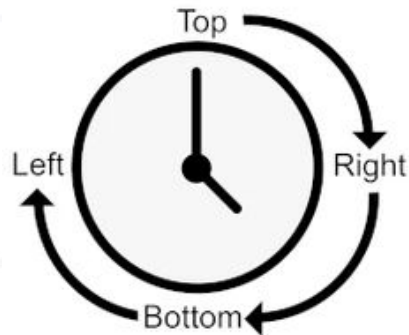
↓   ↓   ↓   ↓  
TOP   RIGHT   BOTTOM   LEFT

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

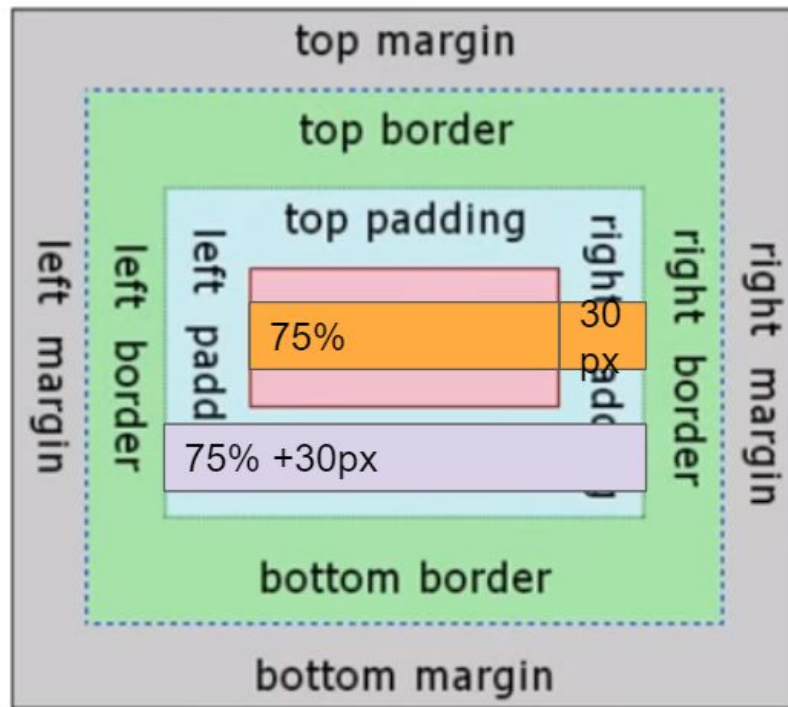
↓   ↓   ↓  
TOP   RIGHT and LEFT   BOTTOM

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

↓   ↓  
TOP and BOTTOM   RIGHT and LEFT



# Box Model completo



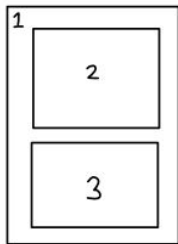
# Resumen: Box Model (Modelo de Caja)

JULIA EVANS  
@bork

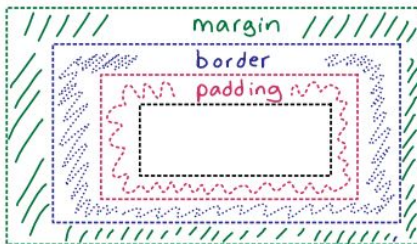
## the box model

every HTML element  
is in a box

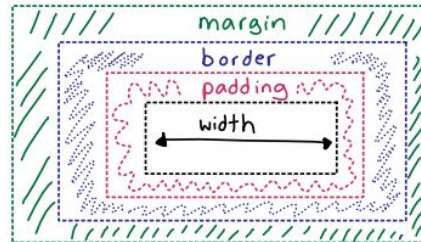
```
<div class="1">  
  <div class="2" />  
  <div class="3" />  
</div>
```



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



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



# Otros temas

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

JULIA EVANS  
@b0rk

# centering in CSS

center text with  
text-align

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

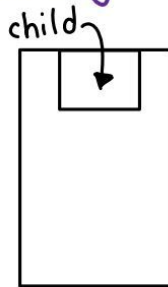
center block elements  
with margin: auto

example HTML:

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

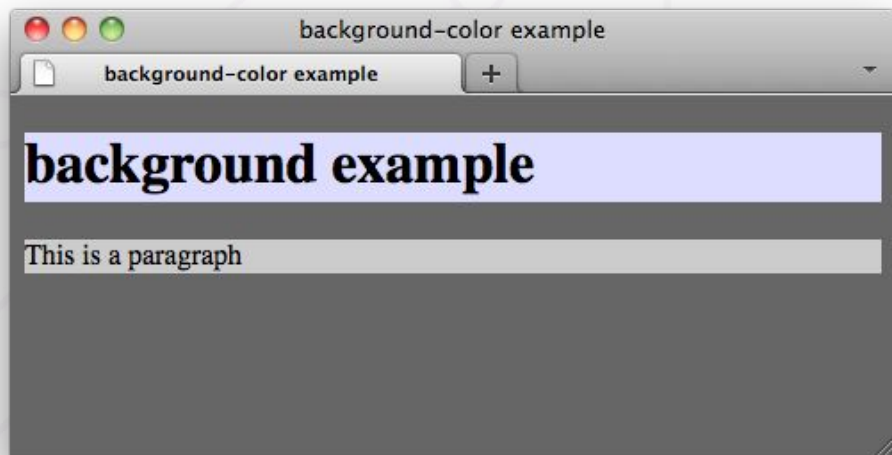
margin: auto  
centers horizontally

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



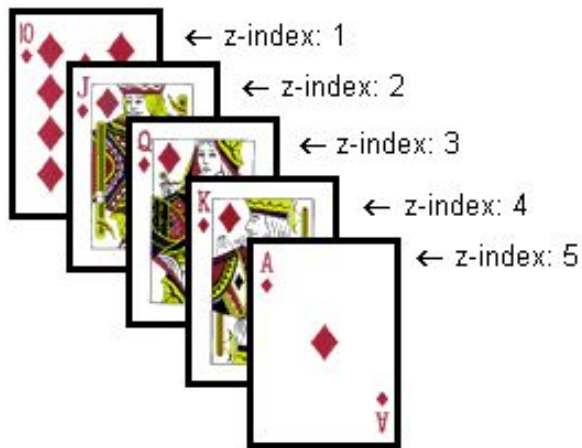
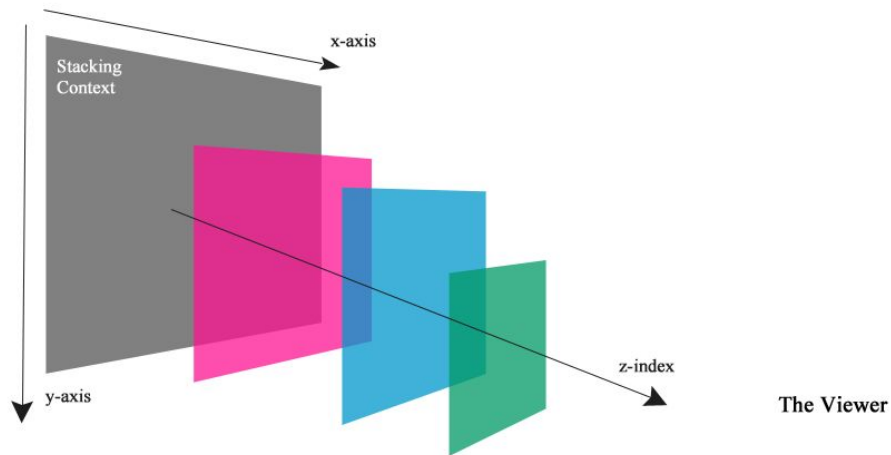
Referencia: Twitter de [Julia Evans](#)

# 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)



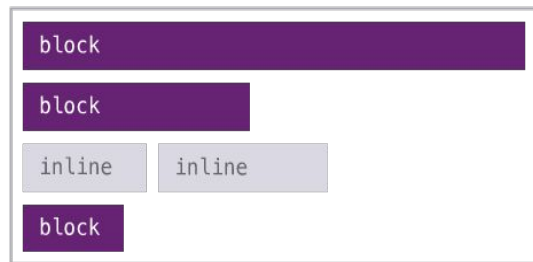
## z-index

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

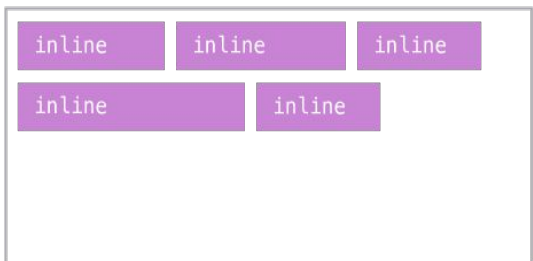
Cosas interesantes saber:

- 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.

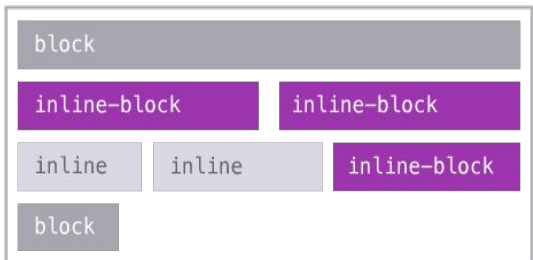




✓ padding ✓ margin ✓ width



✗ 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**

# inline vs block

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

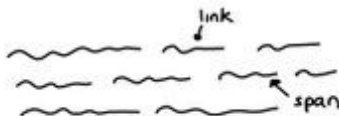
example block  
elements

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

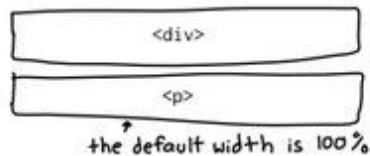
example inline  
elements

```
<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

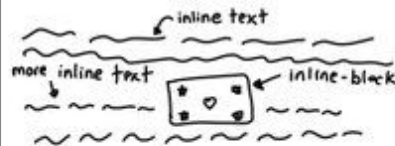
display can force an  
element to be inline or block

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



# 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