

# Fundamentos HTML, CSS

## Introducción a las hojas de estilo con CSS3

Tony G. Bolaño

@tonybolanyo – tonybolanyo@gmail.com

KeepCoding Full Stack Web Developer

Junio 2019





# ■ Introducción a CSS3

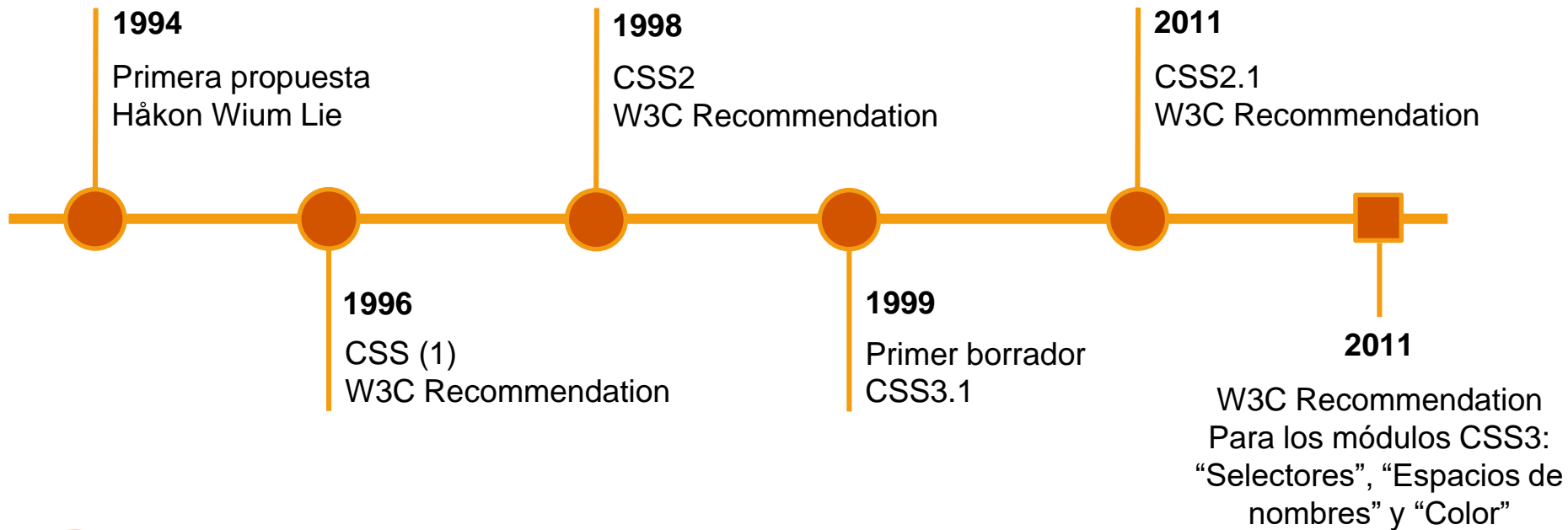


# ■ ¿Qué es CSS?

- CSS: Cascading Style Sheets (hojas de estilo en cascada).
- Lenguaje para definir la presentación de un documento estructurado escrito en un lenguaje de marcado.
- Diseñado para permitir la separación entre el contenido del documento y la forma de presentarlo.



# ■ Un poco de historia



# ■ ¿Cómo usar CSS?

- Estilos en línea, mediante el atributo `style`:

`<p style="propiedad: valor">`

- Estilos independientes. Uso de selectores:

- Embebidos con la etiqueta `<style>`
- En ficheros CSS independientes

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



# ■ Sintaxis: selectores, propiedades y directivas

directiva  
(at-rule)

selector

propiedades

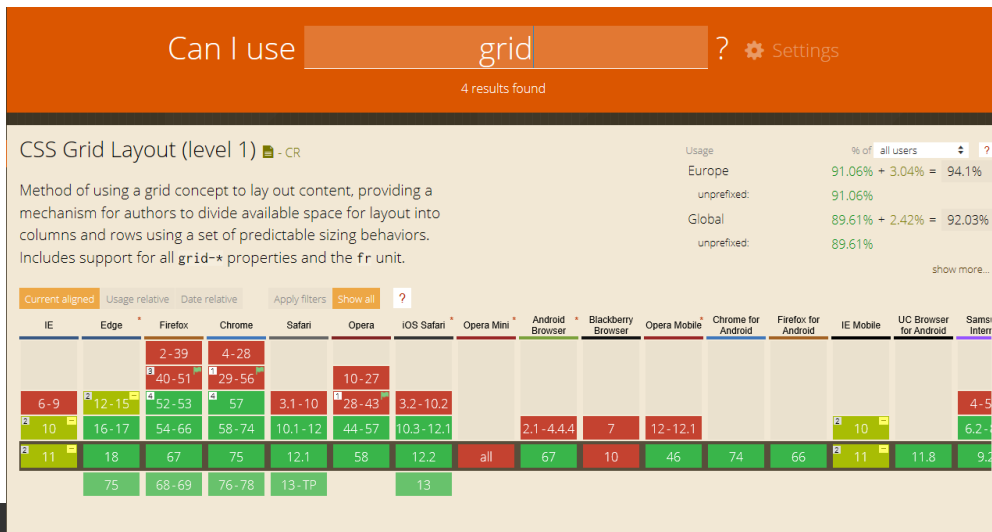
```
@import url('https://...');  
  
h1 {  
  color: darkorange;  
  font-size: 30px;  
}
```

valores

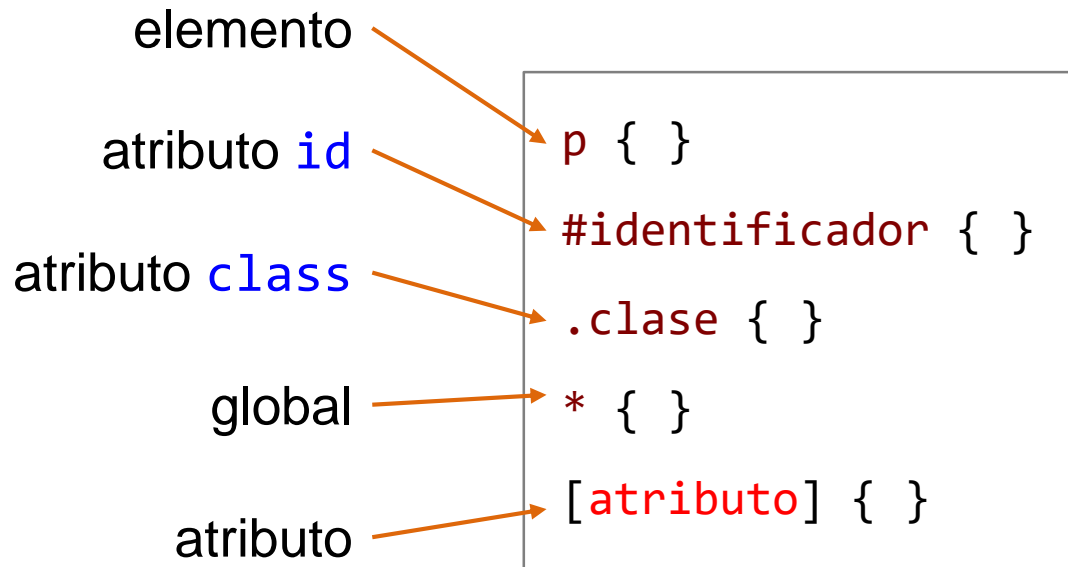


# Herramientas y referencias

- MDN: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
- Can I Use: <https://caniuse.com>
- Navegador: inspector



# ■ Selectores básicos



```
a[href$=".org"] {  
  font-style: italic;  
}
```

`[atributo]`

elemento que tienen el atributo

`[atributo="valor"]`

elementos con ese valor

`[atributo^="valor"]`

elementos que comienzan con el valor

`[atributo$="valor"]`

elementos que terminan con ese valor

`[atributo*="valor"]`

elementos que contienen el valor

`[atributo|= "valor"]`

valor exacto o principio seguido de guión

`[atributo~="valor"]`

lista de palabras y una es valor





# ■ Selectores múltiples y anidados

`h1, h2, h3 { }` • Aplica estilos a múltiples selectores

`div p { }` • Descendientes de cualquier nivel

`div > p { }` • “Hijos” directos

`div + p { }` • “Hermanos” adyacentes (inmediatos)

`div ~ p { }` • “Hermanos” en general



# ■ Pseudo-elementos y Pseudo-clases

- **Pseudoelementos:** elementos definidos desde CSS.
  - `::first-letter`
  - `::first-line`
  - `::before {content=""}`
  - `::after {content=""}`
- **Pseudoclasses:** clases aplicadas dinámicamente a elementos reales de HTML en función de su estado
  - `:link`, `:visited`, `:active`, `:focus`, `:hover`, `:target`, `:lang()`
  - `:nth-child(n)`, `:first-child`, `:last-child`, `:only-child`
  - `:nth-of-type(n)`, `:first-of-type`, `:last-of-type`,
  - `:only-of-type`, `:not()`



# ■ Aplicación de múltiples estilos

- **Herencia**

- Propiedades que se heredan en los elementos hijos  
`font-family`, `color`... => afectan al contenido
- Propiedades que no se heredan  
`margin`, `padding`, `border`... => afectan al contenedor

- **Cascada**

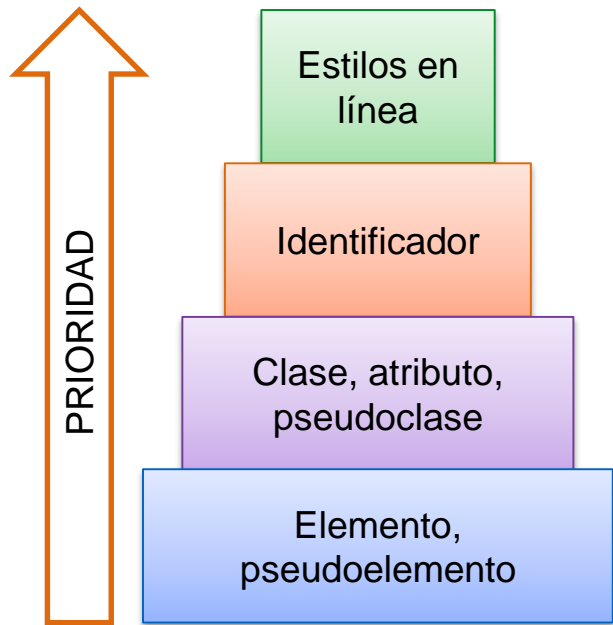
- agente de usuario: por defecto / definidos por el usuario
- bloques de CSS: externos / embebidos
- estilos en línea

- **Especificidad**



# ■ Especificidad

- Orden de prioridad (menor a mayor especificidad):
  - Selectores de tipo (`p`, `h1`) y pseudo-elementos (`::before`, `::after`)
  - Selectores de clase (`.ejemplo`), selectores de atributos (`[type="text"]`) y pseudo-clases (`:hover`)
  - Selectores ID (`#ejemplo`)
- Los estilos inline (`style="color: darkorange"`) tienen prioridad sobre los externos



# ■ Unidades de medida

UNIDADES ABSOLUTAS		UNIDADES RELATIVAS	
Unidad	Medida	Unidad	Relativo a
in	Pulgada (2.54cm)	em	Tamaño de letra del elemento
cm	Centímetro	ex	Altura de la “x” del elemento
mm	Milímetro	vw	Anchura del viewport
pt	Punto. 1pt = 1/72in	vh	Altura del viewport
pc	Pica. 1pc=12pt	%	% del tamaño del elemento
px	Relativos a la pantalla	rem	Tamaño de letra en el root <html>

```
.container {  
  font-size: 2rem;  
  width: 800px;  
  height: 100vh;  
}
```



# ■ Variables. Uso de calc()

- Variables definidas globalmente:

```
:root {  
  --color-principal: #06c;  
}
```

- Se utilizan mediante la función var()

```
#foo h1 {  
  color: var(--color-principal);  
}
```

- La función calc() opera con variables y números

```
.container {  
  --separacion: 20;  
  margin-top: calc(var(--separacion) * 2px);  
}
```





# ■ Propiedades básicas



# ■ Tipografía

- Podemos incluir nuevas fuente de varias formas:

- Mediante la propiedad `@font-face`

```
@font-face {  
  font-family: miFuente;  
  src: url();  
}
```

- Importándola de webs como Google Fonts

```
@import url();  
<link href='' type='text/css'>
```





# ■ Propiedades para fuentes

Propiedad	Significado
font-style	Define el estilo de un texto. Normal, italic,...
font-variant	Variante del tipo de letra. Normal, small-caps
font-weight	Peso de la letra. Normal, bold, 100, 900,...
font-size	Tamaño de la fuente.
font-family	Familia de la fuente
font	Abreviatura de todas las anteriores en el mismo orden de aparición

Otras  
propiedades:

- text-decoration, text-transform
- line-height, letter-spacing, word-spacing
- text-align, vertical-align, text-indent



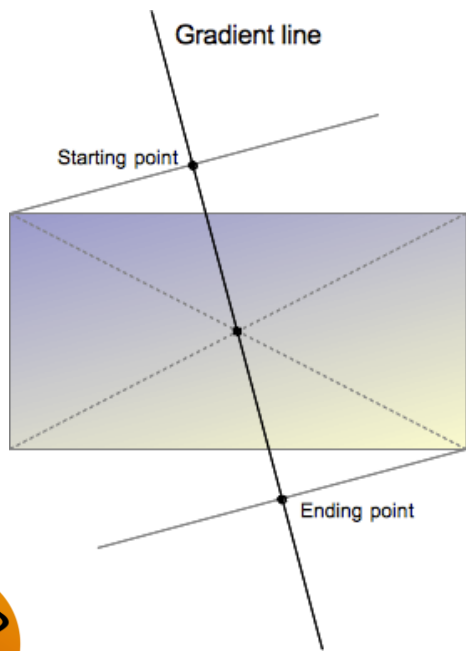
# ■ Colores

- Sistemas de colores
  - Nombres
  - `rgb()` - red, green, blue
  - hexadecimal (#)
  - `hsl()`
  - `hue°` (matiz), `saturation%`, `luminosity%`
- Transparencia canal alfa entre 0 (transparente) y 1 (opaco)
  - `rgba()`
  - `hsla()`
  - `Opacity` / se hereda

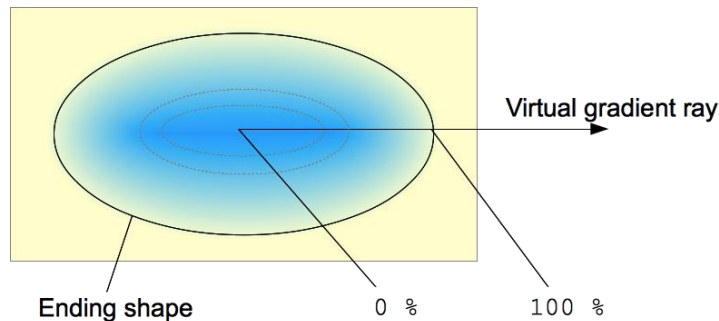


# Degradados

**background:** `linear-gradient(`  
    `dirección, color-stop1,`  
    `color-stop2, color-stop-n);`



**background:** `radial-gradient(`  
    `figura at posición,`  
    `color-stop1, color-stop2,`  
    `color-stop-n);`



# ■ Sombras

- Sombra de caja:

```
.container {  
  box-shadow: offset-x offset-y blur-radius spread-radius color  
}
```

- Sombra de texto:

```
.foo {  
  text-shadow: offset-x offset-y blur-radius color  
}
```



# ■ Backgrounds: imágenes y patrones para el fondo

```
background-image: url(./assets/about-bg.jpg);  
background-repeat: no-repeat;  
background-size: cover;  
background-position: center center;  
background-attachment: fixed;  
background-clip: border-box;  
background-blend-mode: normal;
```

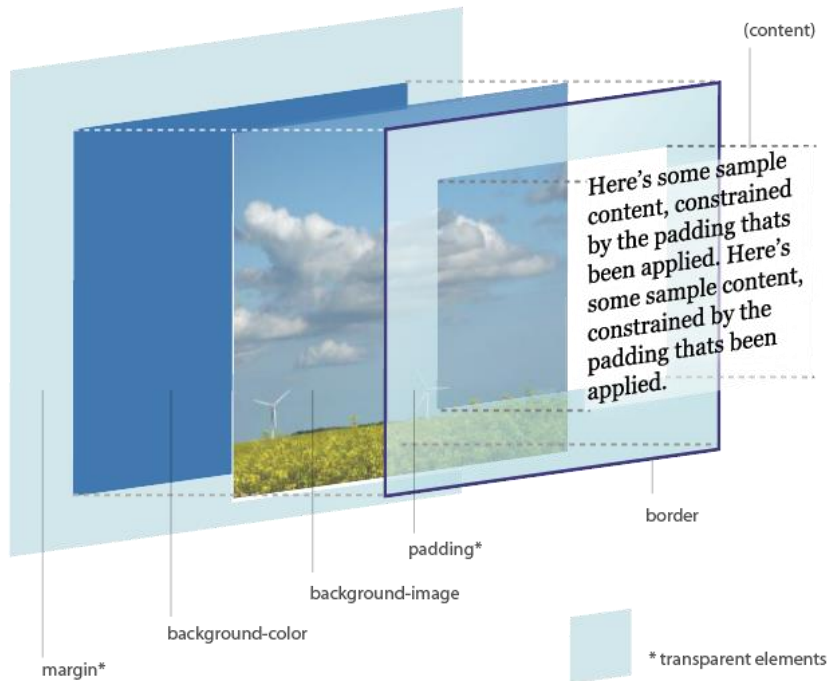




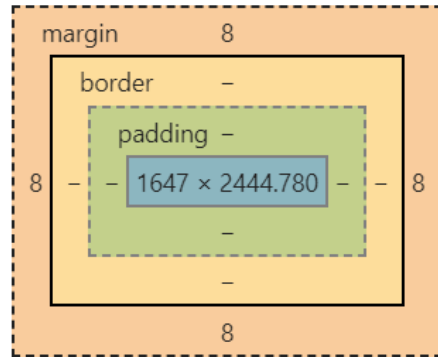
# Modelo de caja



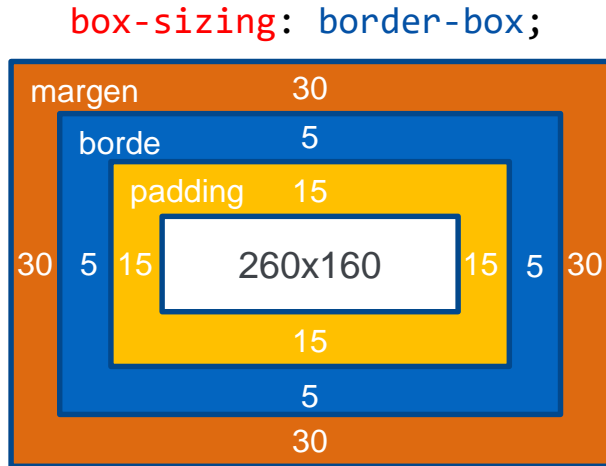
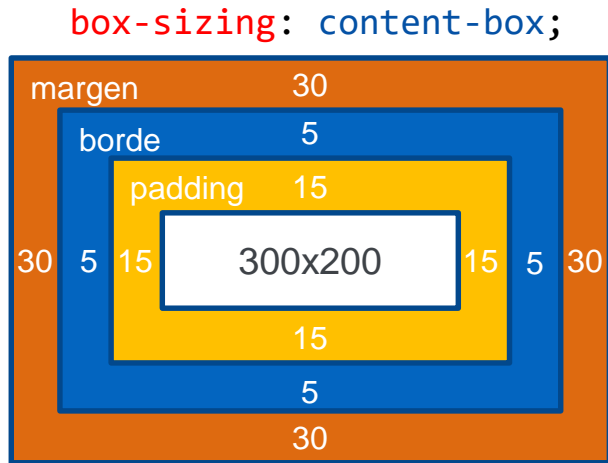
# Modelo de caja en CSS



- Contenido
- Espaciado interno (padding)
- Borde
- Imagen de fondo
- Color de fondo
- Margen



# Modelo de caja. Sistemas de medida



```
.container {  
  width: 300px;  
  height: 200px;  
  padding: 15px;  
  border: 5px solid gray;  
  margin: 30;  
}
```





# ■ Modelo de caja. Propiedades y usos.

- Propiedades
  - margin
  - padding
  - border
  - border-radius
  - border-image
- Usos
  - wrappers
  - centrado de bloques



# ■ Centrado de bloques

- Horizontal:
  - width: %
  - margin-left / margin-right: auto
- Vertical:
  - line-height = height (elementos de 1 linea)
  - display table
  - position absolute 50% / 50% margin -w/2 -h/2
  - position absolute 50% / 50% transform translate -50% -50%

Alternativa: display: flex





# ■ Display y posicionamiento



# ■ Display. Opciones básicas

- La propiedad `display` es la más importante para controlar la estructura de la web.
- Cada elemento HTML tiene un valor por defecto.
- Algunos elementos de “nivel bloque” son: `<div>`, `<h1>...<h6>`, `<p>`, `<form>`, `<section>`
- Algunos elementos de “nivel inline” son: `<span>`, `<a>`, `<img>`



## ■ display: block

- Un elemento block comienza en una nueva línea y se expande en el eje horizontal tanto como pueda.

`<div>`

`<div>`



## ■ display: inline

- Un elemento *inline* continua en la misma línea que el resto del contenido.
- No respeta ni márgenes, ni los paddings **top** ni **bottom**.

Esto es `<span>un elemento</span>` inline.



## ■ display: inline-block

- Se comporta igual que un elemento *inline*, pero a diferencia de él, puede tener tanto un **width** como un **height**.
- También respeta los márgenes y los paddings.

Esto es

```
<span>un elemento</span>
```

inline.



## ■ display: none

- Oculta el elemento sin dejar espacio en el sitio que debería ocupar.
- Usado por algunos elementos como la etiqueta `<script>`.





# ■ Posicionamiento

- La propiedad `position` nos permite crear estructuras más complejas en nuestros diseños.
- Por defecto, los navegadores posicionan todos los elementos de forma automática.
- Puede tomar los valores:
  - `static` (por defecto): sin posicionamiento específico.
  - `relative`: se posiciona como `static` y luego se desplaza
  - `absolute`: referencia al ancestro más cercano (no `static`).
  - `fixed`: variante del absoluto. Su posición es inamovible.



# ■ Posicionamiento

`position: static;`



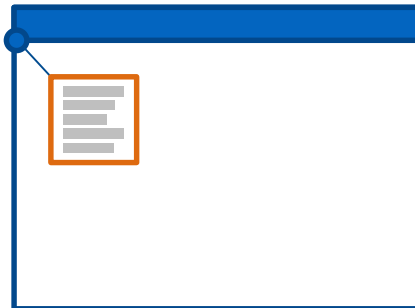
ESTÁTICO O  
NO POSICIONADO

`position: relative;`



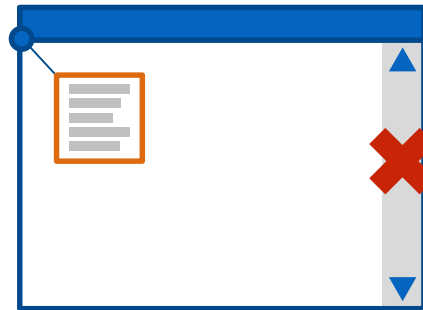
RELATIVO

`position: absolute;`



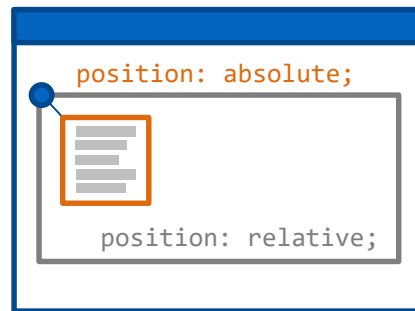
ABSOLUTO

`position: fixed;`



FIJO

`position: absolute;`



(RELATIVAMENTE) ABSOLUTO



# Cambios del flujo estático

- Float: left / right
- Clear: left / right / both
- Overflow: hidden / auto

Técnica de clear-fix con ::after

```
::after {  
  content : "";  
  display: block;  
  clear: both;  
}
```

The Last Great Time War.  
My people fought a race  
called the Daleks, for the  
sake of all creation. And  
they lost. We lost. Everyone  
lost.

I'm the Doctor, I can save  
the world with a kettle and  
some string! And look! I'm  
wearing a vegetable!

“Cuando un elemento  
tiene a todos sus hijos  
flotando, pierde su altura”





# GRACIAS

[www.keepcoding.io](http://www.keepcoding.io)

