

keep coding

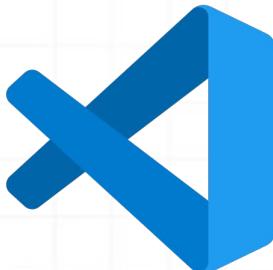
# Fundamentos HTML & CSS

keep coding



# Requisitos

- VSCode (Crear archivos, borrarlos y moverse por el editor)
- Live server extension
- Git



# Frontend development



El desarrollo frontend es la creación de la interfaz de usuario de un sitio web utilizando **HTML, CSS** y JavaScript, permitiendo a los usuarios ver e interactuar con el sitio.

**Solo 3 herramientas,  
entonces es más sencillo** 🤔

# CHARACTER DEVELOPMENT

As if the Looney Tunes didn't have enough troubles, Warner Bros. Feature Animation created a whole new set of bad guys to give them a hard time. Here's how they came to life.

| [BUPKUS](#) | [NADA](#) | [NULL](#) | [VOID](#) | [ZILCH](#) |

## BUPKUS



Search game title



## Console games

## PC games

Viewing 1-20 of 364 results

Action & Adventure ×[CLEAR FILTERS](#)Sort: [Newest](#) ▾

## Collections

### All

[Most popular](#)  
[Recently added](#)  
[Coming soon](#)[Leaving soon](#)[Optimized for Xbox Series X|S](#)[Bethesda Softworks](#)[EA Play](#)[ID@Xbox](#)[Family friendly](#)[Games to conquer your resolutions](#)[Play day one](#)[Heroes and villains](#)

## Filters

[CLEAR FILTERS](#)

## Play on



TUNIC



Tom Clancy's Rainbow Six® Siege Deluxe Edition



Marvel's Guardians of the Galaxy

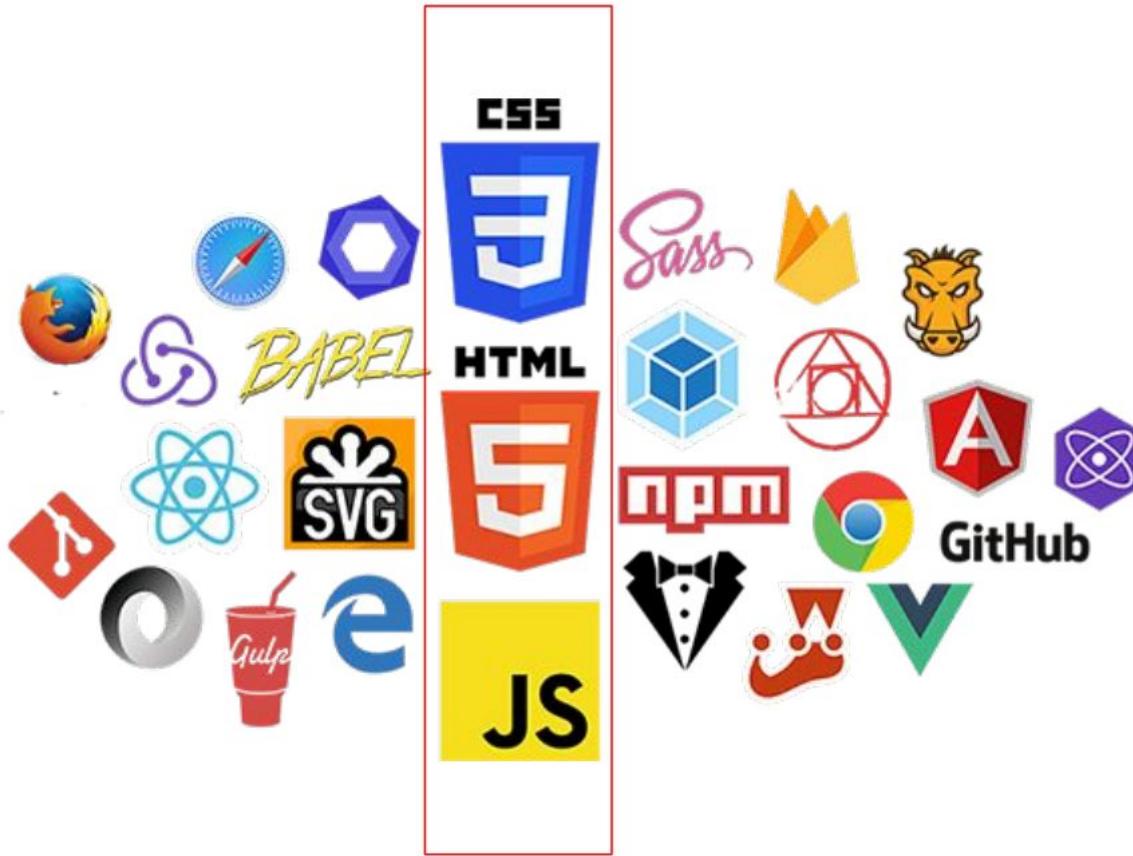


Paradise Killer

keep coding



keep coding



# Curso

- Vamos a crear un sitio web desde el minuto 1.
- Vamos a dividirlo en retos
- No te preocupes, en cada reto vamos a revisar las bases y aplicarlas a el proyecto



keep coding

# CODEFLIX

keep coding

**CODEFLIX**

All shows Sign in

Unlimited movies, TV shows, and more

Watch anywhere. Cancel anytime

Ready to watch?



watch everywhere

Stream unlimited movies and TV shows on your, phone, tablet, laptop, and TV.



Newest



spy masters House of dragon Love & death succession

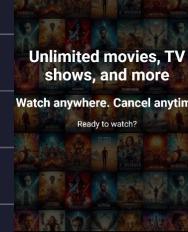
Popular



spy masters House of dragon Love & death succession

2023 Codeflix Website  
Powered By Keepcoding

CODEFLIX Sign in



Unlimited movies, TV shows, and more

Watch anywhere. Cancel anytime

Ready to watch?



watch everywhere

Stream unlimited movies and TV shows on your, phone, tablet, laptop, and TV.



spy masters House of dragon Love & death succession



spy masters House of dragon Love & death succession

2023 Codeflix Website  
Powered By Keepcoding



## Sign in

Email

Age

password

Phone

Format is 600-000-0000

Avatar image

Choose file No file chosen

Fav category

Shows  Docus

Bio

Remember me

country

Select one country

reset

Sign in



## Sign in

Email

Age

password

Phone

Format is 600-000-0000

Avatar image

Choose file No file chosen

Fav category

Shows  Docus

Bio

Remember me

country

Select one country

reset

Sign in



www.keepcoding.io



cursos@keepcoding.io



(+34) 916 33 1779

# Reto 1

**Objetivo:** Crear un proyecto de HTML y CSS

**Temario:**

- Entorno de trabajo VSCode & live server
- Intro a HTML y estructura básica
- Etiquetas básicas
- Etiqueta HEAD
- Intro a CSS

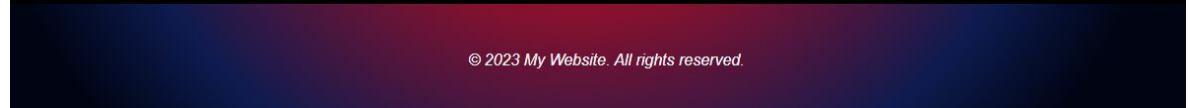


# Reto 2

**Objetivo:** Crear un footer

**Temario:**

- HTML text tags
- CSS selectores
- CSS variables
- Font styles
- Paddings/Margins



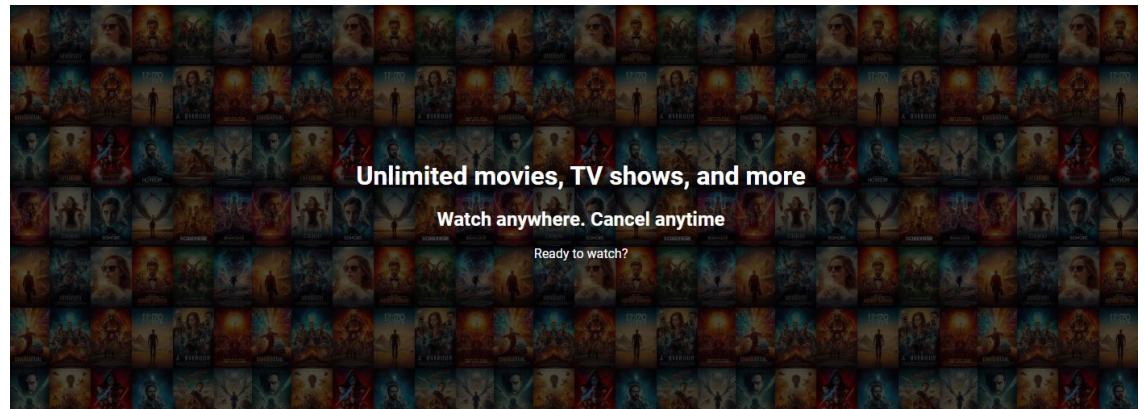
© 2023 My Website. All rights reserved.

# Reto 3

**Objetivo:** Crear un Banner

**Temario:**

- Div
- Text tags
- positions
- viewport sizes
- background images
- opacity

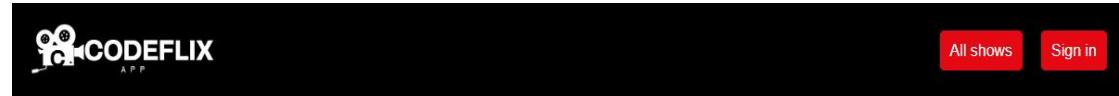


# Reto 4

**Objetivo:** Crear un Header con navegación.

**Temario:**

- Semantic HTML
- Lists
- Hide elements
- CSS components
- Display flex

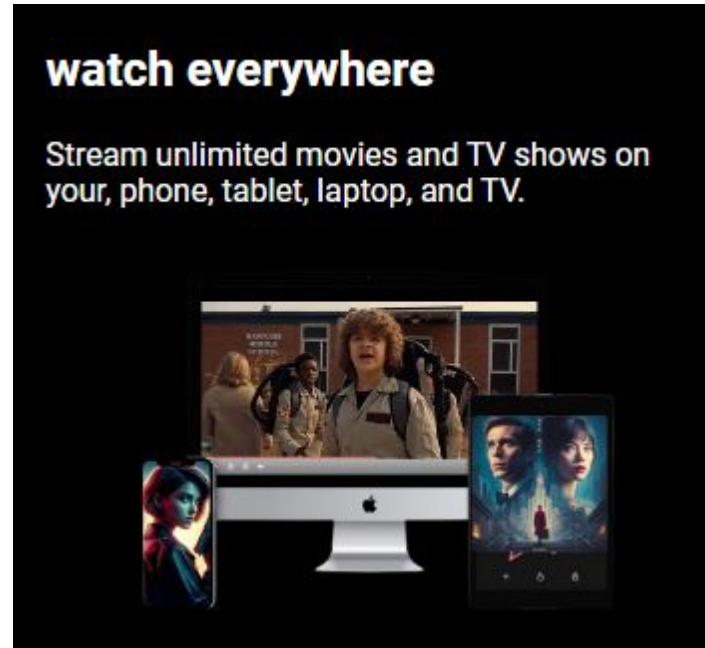


# Reto 5

**Objetivo:** Info section

**Temario:**

- video tags
- img
- positions

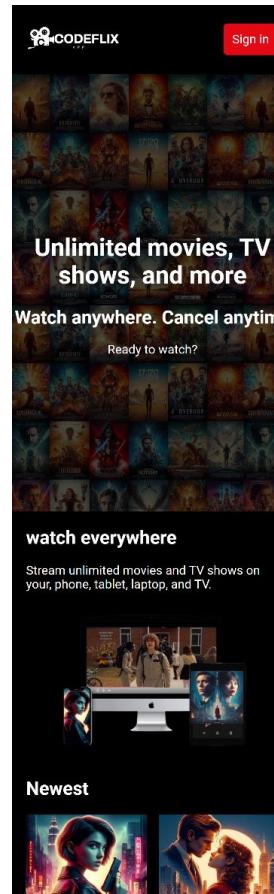


# Reto 6

**Objetivo:** Mobile first

**Temario:**

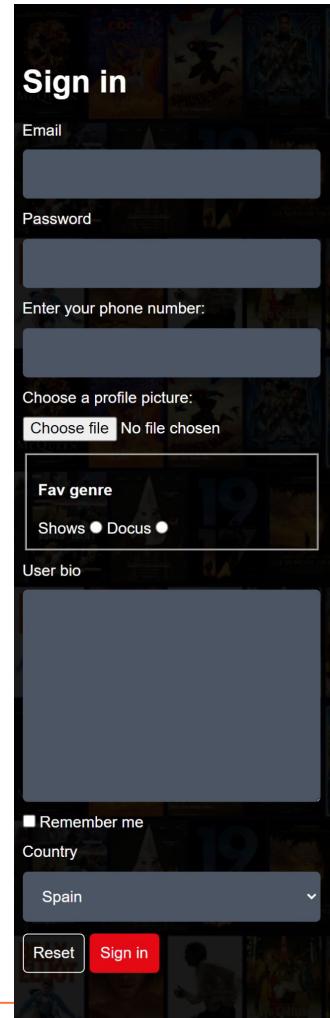
- Responsive design
- Media queries



# Reto 7

**Objetivo:** Formulario de registro  
**temario:**

- HTML form tags
- CSS components
- Form validation



keep coding

# Reto 8

**Objetivo:** Grid

**Temario:**

- Responsive
- Media queries
- CSS grid

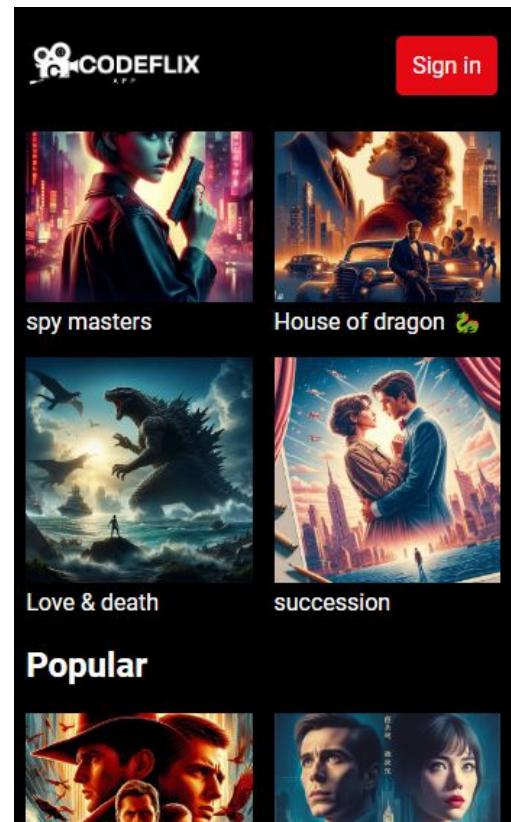


# Reto 9

**Objetivo:** Transiciones y animaciones

**Temario:**

- Open graph
- Hover selector
- Transitions
- Animations
- gradient



# Lets go!



# Reto 1

**Objetivo:** Crear un proyecto de HTML y CSS

**Temario:**

- Entorno de trabajo VSCode & live server
- Intro a HTML y estructura básica
- Etiquetas básicas
- Etiqueta HEAD
- Intro a CSS



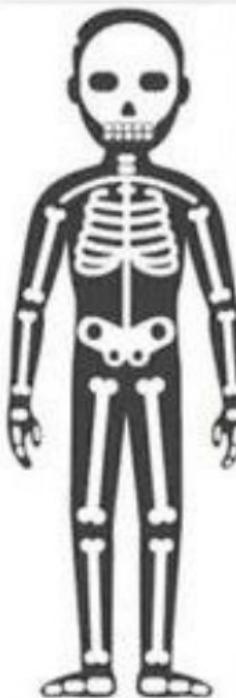
# HTML



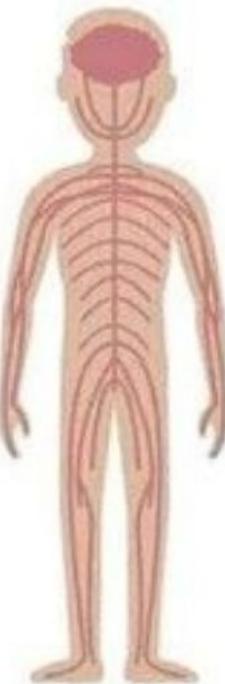
HTML (HyperText Markup Language) es el componente básico de la Web. Define el significado y la estructura de los contenidos web.

Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o funcionalidad/comportamiento (JavaScript).

HTML



JS



CSS

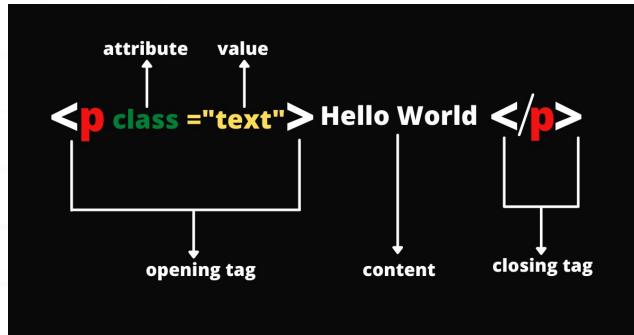


HTML utiliza "marcado" para anotar texto, imágenes y otro contenido para mostrar en un navegador web. El marcado HTML incluye elementos especiales como `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, `<span>`, `<img>`, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>`, `<ul>`, `<ol>`, `<li>` y muchos otros.

# Etiquetas básicas



# Paired



# Unpaired

```
<hr>
<hr />
```

# html:5



# Estructura básica

- El <!DOCTYPE html> es una declaración especial que se coloca al comienzo de un documento HTML para indicar al navegador web qué versión de HTML se está utilizando. Esta declaración debe estar presente en todas las páginas HTML válidas.
  - <html></html> envuelve todo el documento HTML y sirve como contenedor principal.
  - <head></head> Se utiliza para contener información y metadatos relacionados con el documento HTML, pero que no se muestra directamente en el navegador. Es decir, se utiliza para configurar y definir la estructura de la página, pero no para mostrar contenido visible.
  - <body></body> Se utiliza para delimitar el contenido principal de una página HTML
-

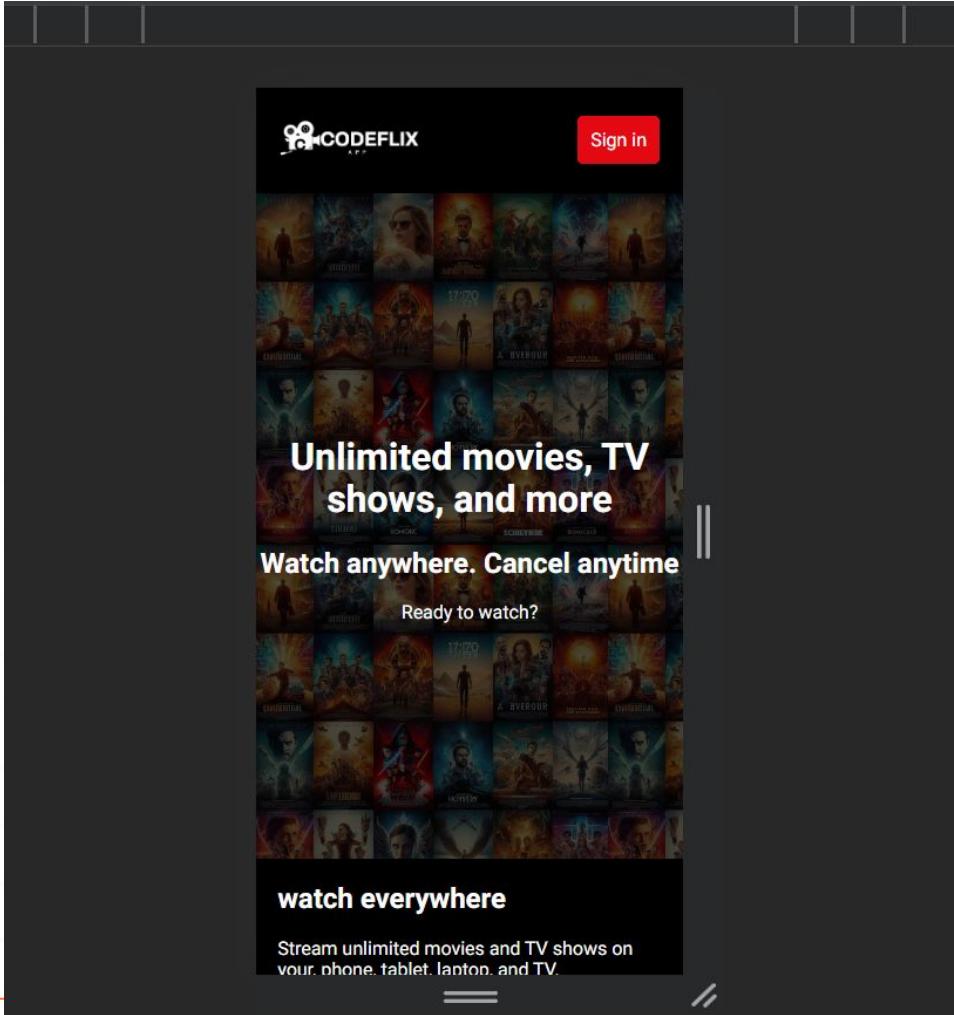
# head



```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

keep coding



Dimensions: Responsive ▾

364

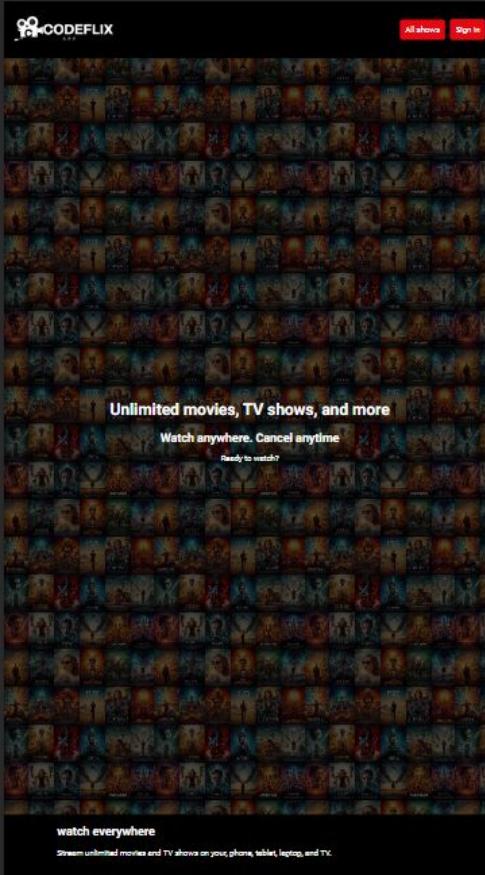
x 757

100% ▾

DPR: 3.0 ▾ No throttling ▾



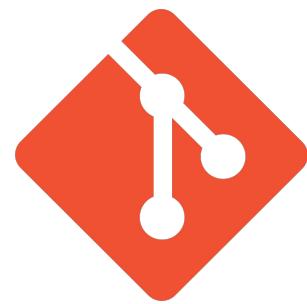
Deep coding



# VSCODE live server



keep coding



---

<https://gist.github.com/lancejpollard/1978404>

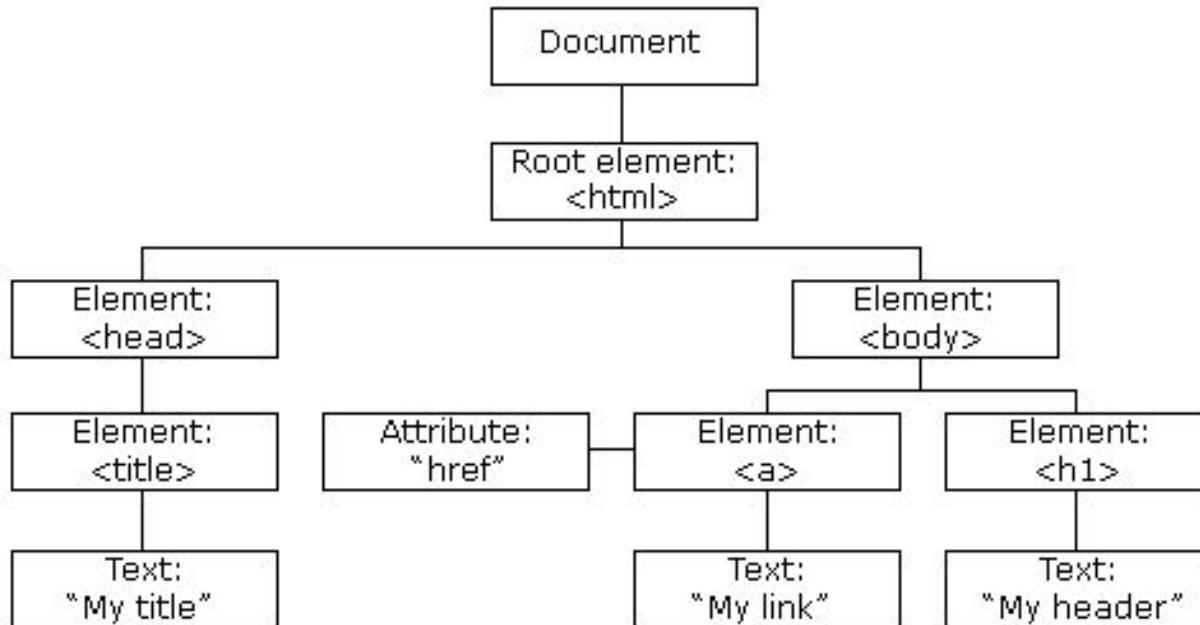
# headings



# DOM Tree

When a web page is loaded, the browser creates a **Document Object Model** of the page.

The HTML **DOM** model is constructed as a tree of Objects.



# CSS



CSS es un lenguaje que especifica cómo se presentan los documentos a los usuarios: cómo se les aplica el estilo.



# CSS Syntax



# CSS syntax

```
h1 {
```

1. Selector

*What are you selecting? What element do you want to apply CSS to e.g. H1.*

```
color: red;
```

3. Value

*What's the value? Properties have specific values that are applicable e.g. red.*

2. Property

*What's the property?  
What are you trying to change?  
E.g. 'color'*



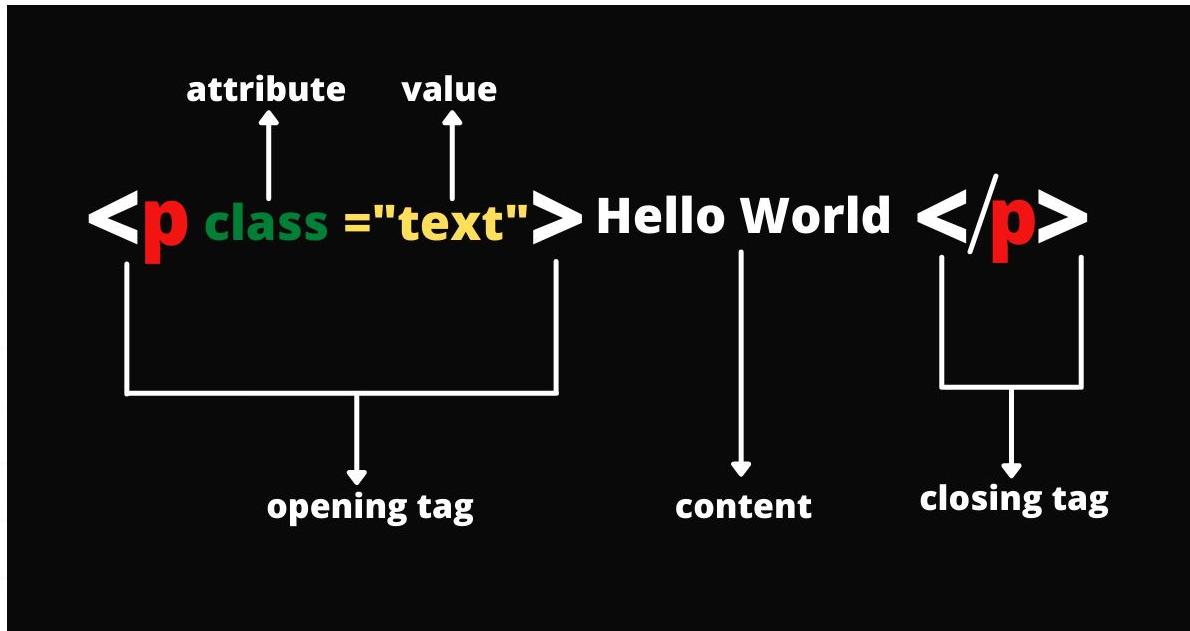
# ¿Cómo utilizarlo?



# ¿Cómo utilizarlo?

- Inline styles (Nada recomendado)
- <head><style></style></head> tag
- Archivos externos <link />





# CSS basic



# Selectors

```
p { }  
#id{ }  
.class{ }  
* { }  
[attribute] { }
```

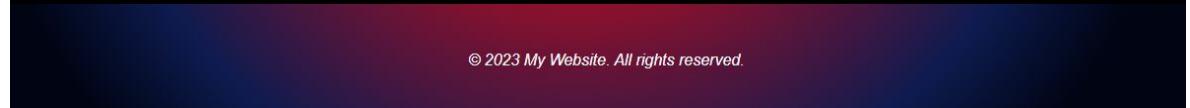
---

# Reto 2

**Objetivo:** Crear un footer

**Temario:**

- HTML text tags
- CSS selectores
- CSS variables
- Font styles
- Paddings/Margins



© 2023 My Website. All rights reserved.

# Paragraphs / texts

# Nested tags

```
139 | <!-- Basic tags -->
140 | <footer class="footer">
141 |   <p>&copy; 2023 My Website. All rights reserved.</p>
142 | </footer>
```

# Block / inline-block



# Selectors

p { }

#id{ }

.class{ }

\* { }

[attribute] { }

# colors

## Sistemas de colores

- names
- rgb() -red, green, blue
- hexadecimal (#)
- hsl()

Transparencia canal alfa entre 0 (transparente) y 1 (opaco)

- rgba()
- hsla()
- Opacity



# padding / margins



# Unidades

- Absolutes
  - px
  - cm
  - in
  - pt. ipt = 1/72in
- Relatives
  - em
  - rem
  - vw
  - vh
  - %



# font properties

- font-style
  - font-weight
  - font-size
  - font
  - text-decoration
  - text-transform
  - line-height
  - text-align
-

# variables

```
element {  
  --main-bg-color: brown;  
}
```

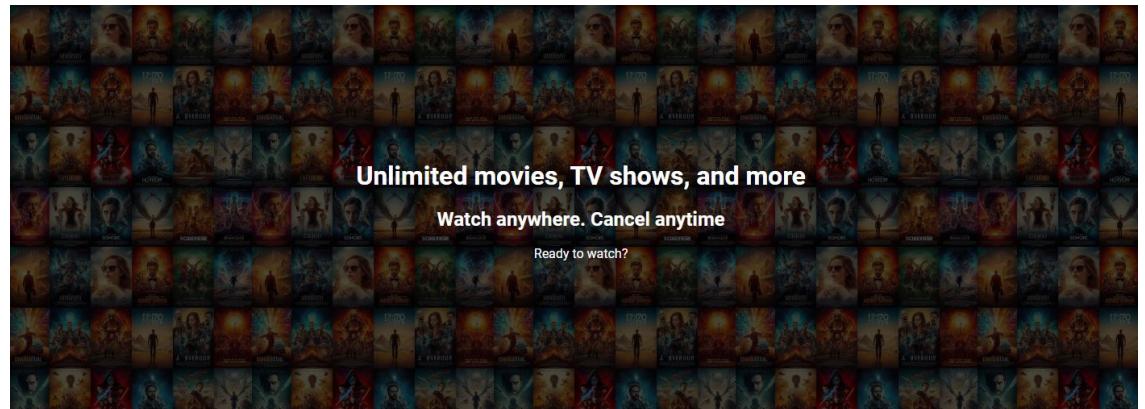
```
:root {  
  --main-bg-color: brown;  
}
```

# Reto 3

**Objetivo:** Crear un Banner

**Temario:**

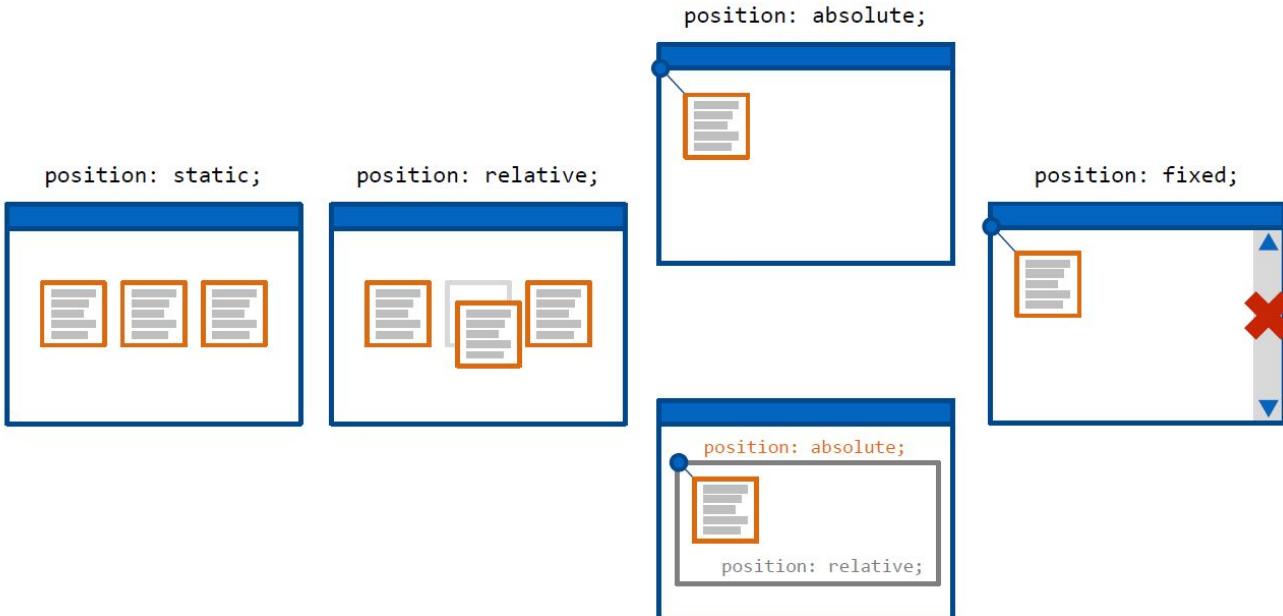
- Div
- Text tags
- positions
- viewport sizes
- background images
- opacity



# <div>

# position

- absolute
- fixed
- static
- relative

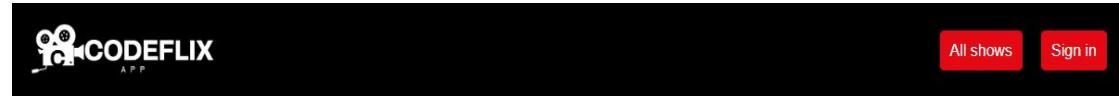


# Reto 4

**Objetivo:** Crear un Header con navegación.

**Temario:**

- Semantic HTML
- Lists
- Hide elements
- CSS components
- Display flex



# Lists



# Links



# Semantic HTML



# Semantic HTML beneficios

- Los motores de búsqueda considerarán su contenido como palabras clave importantes para influir en el ranking de búsqueda de la página (SEO).
- Ayuda a los lectores de pantalla a tener más claro la estructura.
- Es más sencillo de navegar por el código.



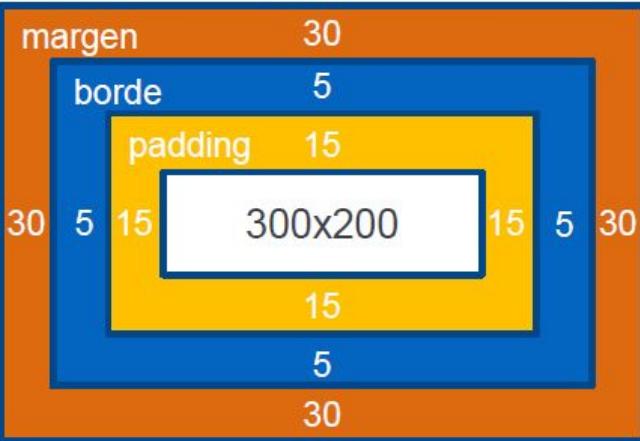
```
<div class="main">
  <div class="section">
    <div>
      <div class="list-item">List item</div>
      <div class="list-item">List item</div>
      <div class="list-item">List item</div>
      <div class="list-item">List item</div>
    </div>
  </div>
  <div class="section">
    <div class="image">
    </div>
  </div>
</div>
```

```
<main class="main">
  <section class="section">
    <ul>
      <li class="list-item">List item</li>
      <li class="list-item">List item</li>
      <li class="list-item">List item</li>
      <li class="list-item">List item</li>
    </ul>
  </section>
  <section class="section">
    
  </section>
</main>
```

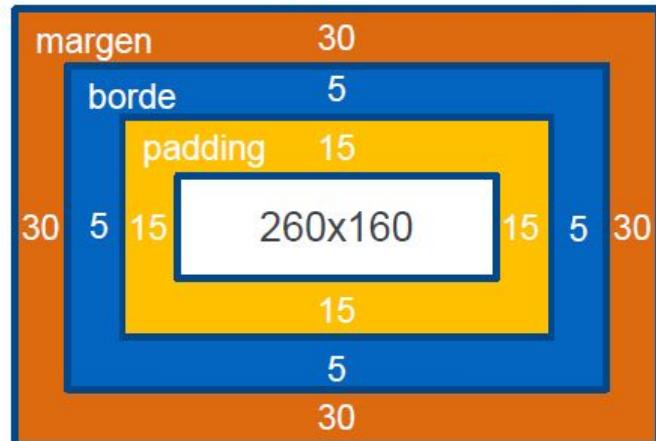
HTML should be coded to represent the data that will be populated and not based on its default presentation styling. Presentation (how it should look), is the sole responsibility of CSS.

```
10 <table width="100%">
11   <tr>
12     <td>This</td>
13     <td>is</td>
14   </tr>
15   <tr>
16     <td>bad</td>
17     <td>layout?</td>
18   </tr>
19 </table>
20
21 <div class="container-fluid">
22   <div class="row">
23     <div class="col-md-6">But</div>
24     <div class="col-md-6">this</div>
25   </div>
26   <div class="row">
27     <div class="col-md-6">is</div>
28     <div class="col-md-6">ok?</div>
29   </div>
30 </div>
```

**box-sizing: content-box;**



**box-sizing: border-box;**



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

# display

- block
- inline-block
- none



# Flexbox

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

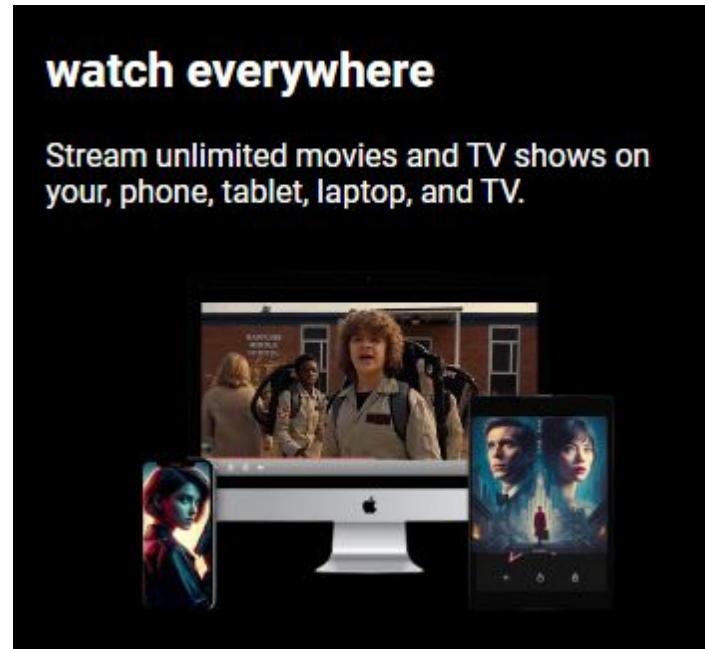
---

# Reto 5

**Objetivo:** Info section

**Temario:**

- video tags
- img
- positions



# Images / video / iframes

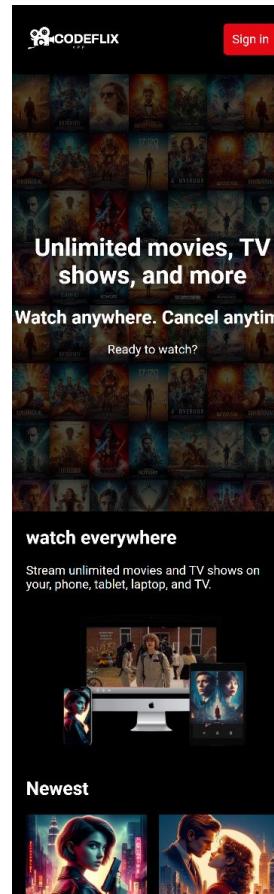


# Reto 6

**Objetivo:** Mobile first

**Temario:**

- Responsive design
- Media queries



# Responsive design



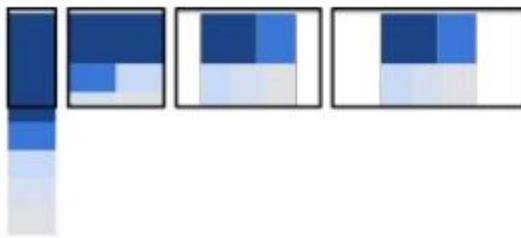
Diseño responsive es la adaptación de un sitio web, aplicación o presentación a diferentes dispositivos y tamaños de pantalla para garantizar una visualización óptima en todos ellos.



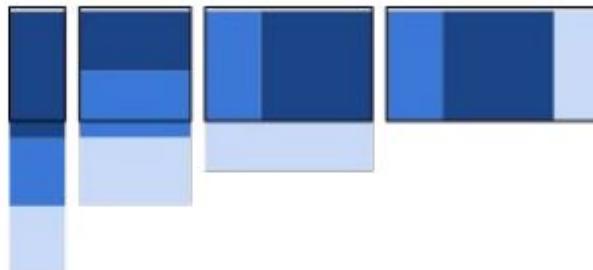
keep coding



### Mostly Fluid



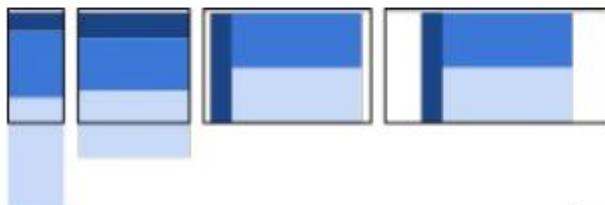
### Column drop



### Off canvas



### Layout shifter



### Tiny tweaks



<https://web.dev/learn/design/>

# media queries



Media queries are useful when you want to modify your site or app depending on a device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width).



# media queries

- type

```
@media screen, print { ... }
```

- features

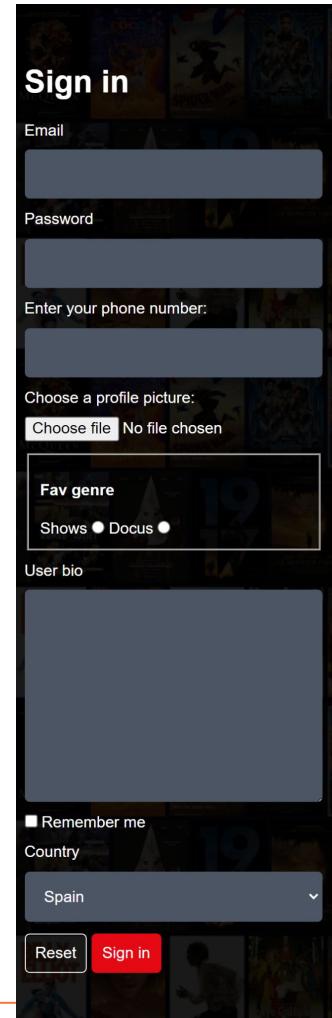
```
@media (max-width: 12450px) { ... }
```



# Reto 7

**Objetivo:** Formulario de registro  
**temario:**

- HTML form tags
- CSS components
- Form validation



keep coding

# HTML forms



La etiqueta HTML <form> se utiliza para crear un formulario en una página web. Permite a los usuarios ingresar y enviar datos, como texto, opciones seleccionadas o archivos, al servidor web para su procesamiento.

# forms elements



El formulario puede contener diversos elementos interactivos, como campos de texto, casillas de verificación, botones de opción y botones de envío. Cuando se envía el formulario, los datos ingresados se pueden procesar, almacenar o enviar por correo electrónico según la configuración del servidor.



# forms attributes



# Form validation



<b>Attribute</b>	<b>Description</b>
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field



<b>Attribute</b>	<b>Description</b>
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field

Attribute	Description
value	value
data-*	Custom Attributes that we can use in our javascript code

# Reto 8

**Objetivo:** Grid

**Temario:**

- Responsive
- Media queries
- CSS grid



# Grid

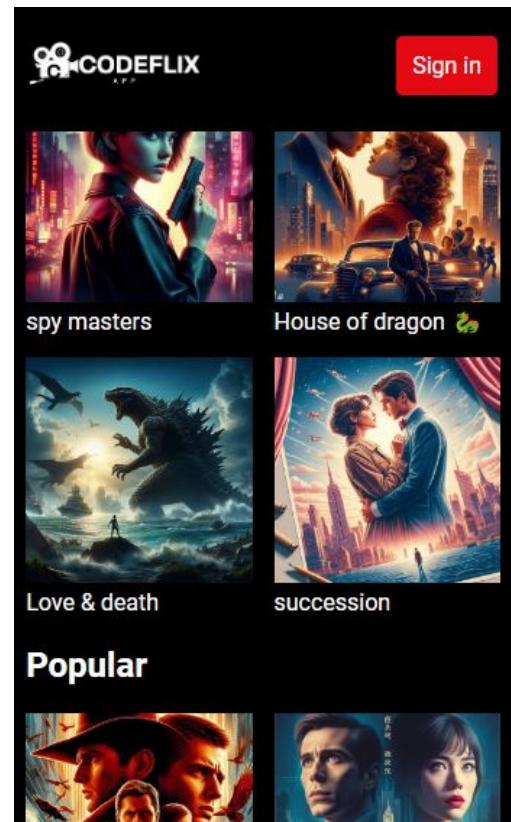
<https://css-tricks.com/snippets/css/complete-guide-grid/>

# Reto 9

**Objetivo:** Transiciones y animaciones

**Temario:**

- Open graph
- Hover selector
- Transitions
- Animations
- gradient



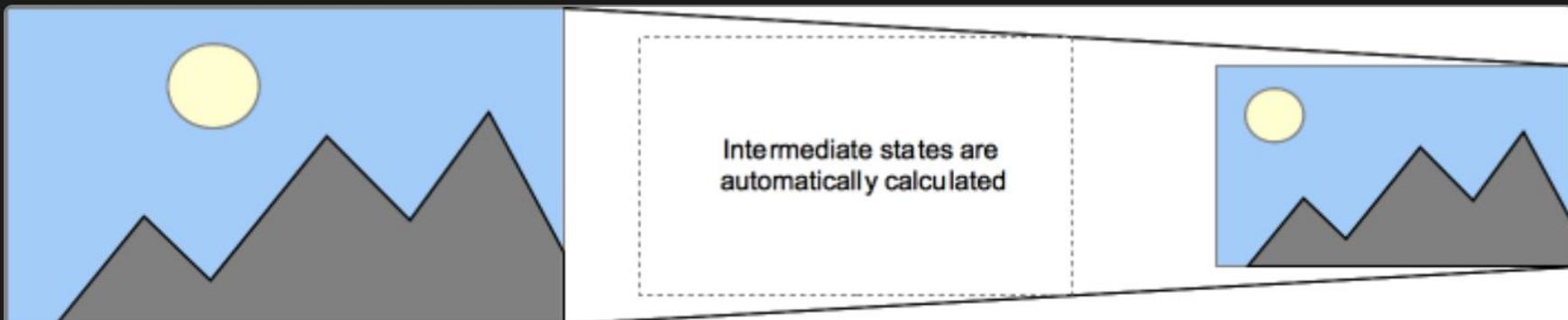
# Open graph tags



```
<meta property="og:image"  
content="https://developer.mozilla.org/static/img/opengraph-logo.png">  
<meta property="og:description" content="The Mozilla Developer Network (MDN)  
provides  
information about Open Web technologies including HTML, CSS, and APIs for both Web  
sites  
and HTML5 Apps. It also documents Mozilla products, like Firefox OS.">  
<meta property="og:title" content="Mozilla Developer Network">
```

# Transitions





Initial state

Final state

```
#delay {  
    font-size: 14px;  
    transition-property: font-size;  
    transition-duration: 4s;  
    transition-delay: 2s;  
}  
  
#delay:hover {  
    font-size: 36px;  
}
```

<https://matthewlein.com/tools/ceaser>

# animations



Una animación permite que un elemento cambie gradualmente de un estilo a otro.

Puedes cambiar tantas propiedades CSS como deseas, tantas veces como deseas.

Para utilizar animaciones CSS, primero debes especificar algunos "keyframes" (fotogramas clave) para la animación.

Los keyframes indican qué estilos tendrá el elemento en momentos específicos.

---

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

# Selectors

ul > li { }

label + input { }

ul li:first-child { }

ul li:nth-child(1) { }

# Browser support





<https://caniuse.com/>



# Goodbye, Internet Explorer 11!

Internet Explorer is finally dead and has been dead for

<https://death-to-ie11.com/>

keep coding



---

# Responsive images

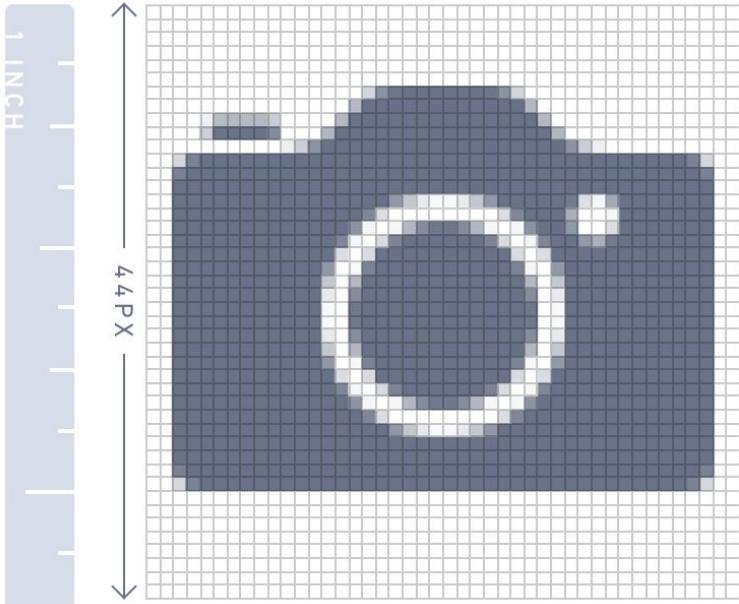


<picture><source>

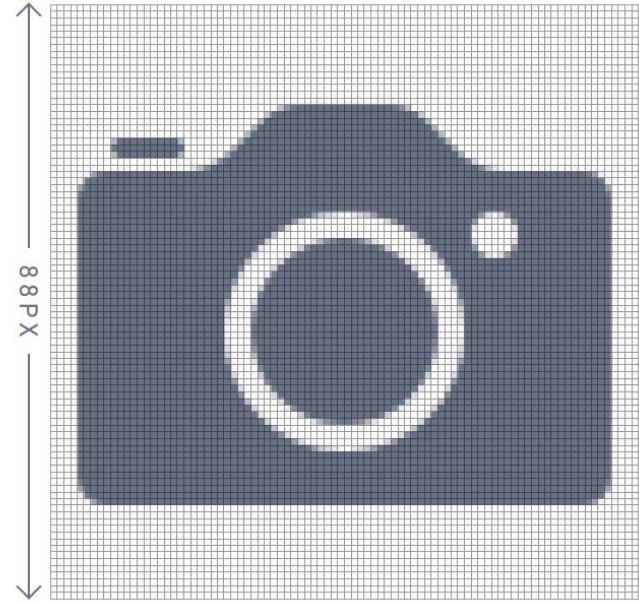
# srcset



coding



Pre-Retina Display (1x)



Retina Display (2x)



→ Increasing Dots Per Inch →

# fonts



keep coding



# Deployment



# Tables



<https://github.com/mailchimp/email-blueprints>

# keep coding

<https://www.linkedin.com/in/kevinjmartinez/>



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



[cursos@keepcoding.io](mailto:cursos@keepcoding.io)



(+34) 916 33 1779