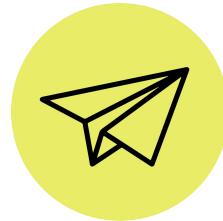


# CÓDIGO PA' LANTE

EidoS  
ALWAYS LEARNING

accenture >



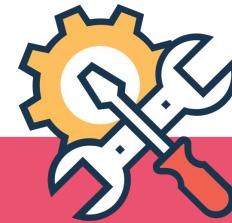
**¡Les damos la bienvenida!**  
**CLASE 4**



**Etiquetas  
estructurales  
semánticas**



**Enlaces y  
rutas**



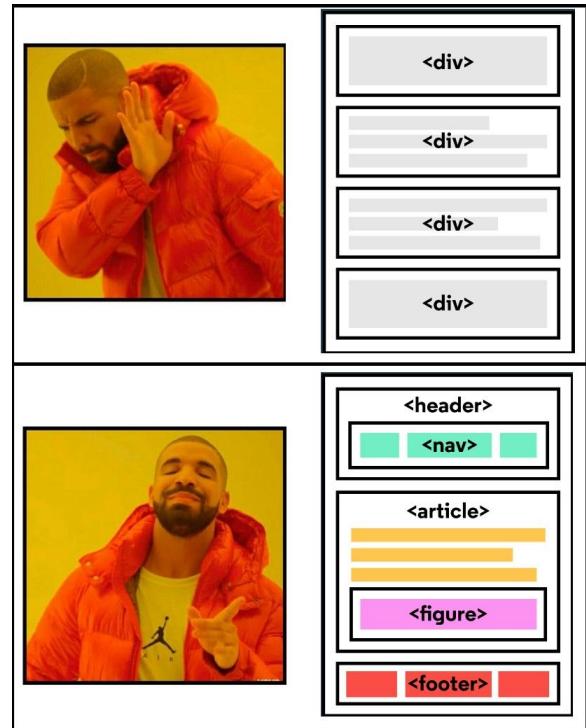
**Herramientas**

Fonts, íconos, inspector,  
unidades de medida.

# Etiquetas estructurales semánticas

# Etiquetas estructurales semánticas

En versiones anteriores a HTML5, al crear la estructura de una página, normalmente se utilizaban etiquetas `<div>` para ir agrupando secciones de contenido. En HTML5 se incorporaron las etiquetas semánticas, las cuales describen en su nombre el contenido que encierran.



# Scanner de etiquetas



¿Qué color es cada etiqueta?

<footer></footer>  
<nav></nav>  
<section></section>  
<header></header>  
<article></article>  
<main></main>

**Etiquetas estructurales**

Inicio Sobre mí Contacto

**Lorem Ipsum**

**Lorem Ipsum**

Codear Latam 2022

## etiquetas-estructurales.html

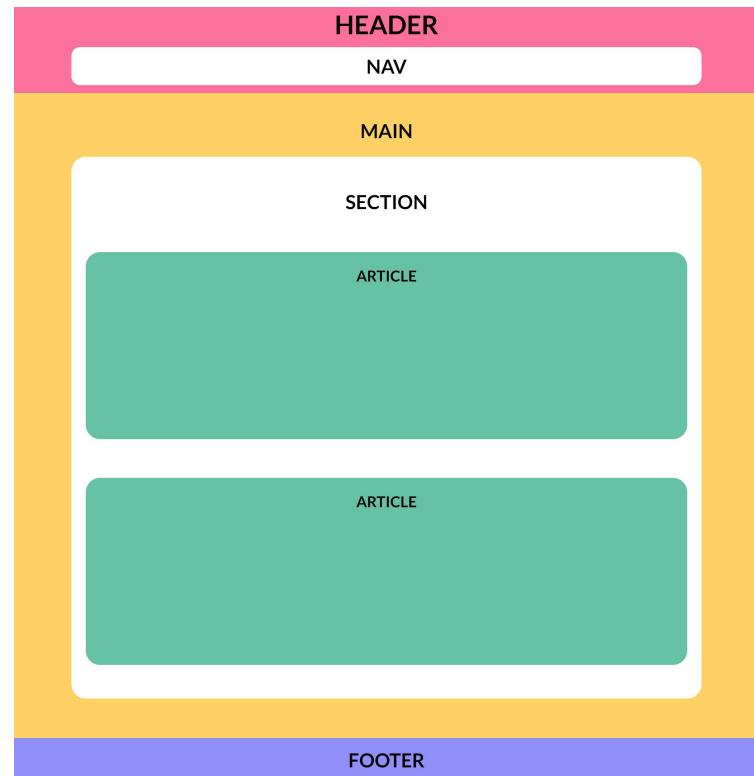
```
<body>

    <header>
        <nav></nav>
    </header>

    <main>
        <section>
            <article></article>
            <article></article>
        </section>
    </main>

    <footer></footer>

</body>
```





## ¿Qué función cumple cada una?

- <header> suele contener elementos introductorios como logos, menús de navegación, buscadores, encabezados. Puede utilizarse dentro de otras secciones también.
- <nav> contiene los enlaces de navegación. Suele incluir un <ul> cuyos <li> contienen los enlaces.
- <main> representa al contenido principal de nuestro documento. Puede estar subdividido en secciones.
- <section> se utiliza para agrupar contenido, definir secciones en la página.
- <article> define una porción de contenido independiente, que encierra sentido por sí misma.
- <footer> pie de página. Suele contener textos legales, información sobre el autor, datos de contacto, enlaces.

# Enlaces y rutas

# Enlaces y rutas

Lo que caracteriza a la web como medio de comunicación es la navegación a través de enlaces.

En HTML usamos el elemento `<a>` (*anchor*) para crearlos. Este elemento lleva el atributo `href` cuyo valor es la ruta a donde dirigirnos. Estos enlaces pueden ser **internos**, **externos** o **anclas** y las rutas pueden ser **relativas** o **absolutas**.





# Rutas relativas y absolutas

Las **rutas relativas** definen la ubicación de un archivo en relación a la ubicación del archivo desde donde se está escribiendo el enlace, las vamos a usar para vincular entre sí archivos html, css, js, imágenes, etc. que conforman nuestro sitio.

Las **rutas absolutas** contienen la ubicación completa y exacta de un archivo, las utilizamos cuando queremos insertar un enlace a otro sitio web.



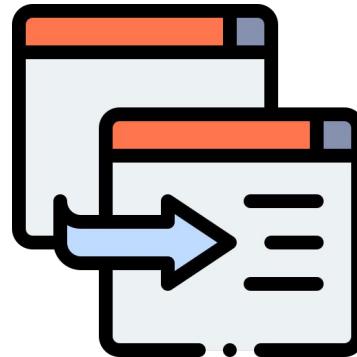
**Rutas relativas:**  
img/mi-foto.jpg  
css/styles.css

**Rutas absolutas:**  
<https://eidosglobal.org/>



# Enlaces internos

Se trata de enlaces para movernos dentro un mismo sitio, por ejemplo, a través de un menú de navegación. Los vamos a escribir usando rutas relativas.



```
<a href="contacto.html">Contacto</a>
```

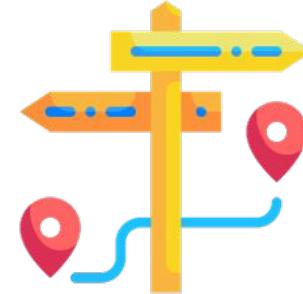


Vínculo con un archivo HTML que contiene otra sección del sitio, está dentro de la misma carpeta.



# Enlaces externos

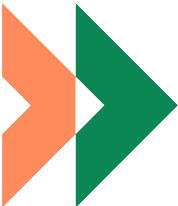
En este caso, vamos a dirigirnos fuera de nuestro sitio, para lo cual usamos rutas absolutas. Solemos usar el atributo `target="_blank"` para que se abra en una nueva pestaña.



```
<a href="https://www.otrositio.com" target="_blank">Ir a otro sitio web</a>
```



Ruta absoluta que nos dirige a otro sitio en la web.



# Anclas

También podemos encontrar enlaces dentro de un mismo documento, llamados "anclas" (esta fue la función original de la etiqueta `<a>`, de ahí su nombre). En este caso utilizaremos el `id` del elemento al que nos queremos desplazar, por ejemplo, el id de una sección específica.

```
<a href="#contacto">Contacto</a>
```

ID de la sección "contacto", en este caso, no se trata de otro documento HTML, sino que la sección está dentro de la misma página.



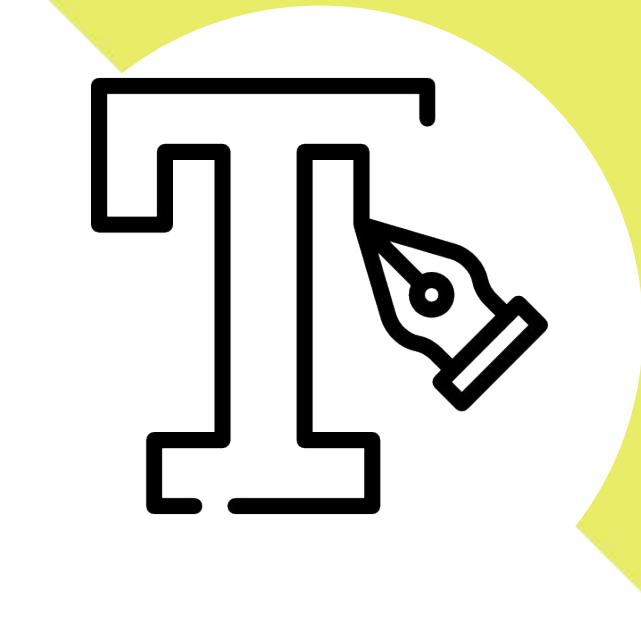
Ejemplo de "One page" con navegación por anclas  
<https://eidosglobal.org/>

# Herramientas para Devs

# íconos y fonts

# Jugando con fonts

Las **fonts** son una parte importante de nuestra web, transmiten más de lo que parece. Existen diferentes familias, estilos y combinaciones, existen **fuentes seguras** soportadas por todos los navegadores y también podemos importar fuentes desde distintas **librerías**. La diferencia más básica de estilo es entre serif y sans-serif, les recomendamos **este tutorial** de W3Schools que profundiza en detalle.



# Google fonts

Es una librería de fuentes libres de derechos y optimizadas para web.

The screenshot shows the Google Fonts homepage. At the top, there's a search bar with 'Search fonts' and a sentence input field with 'Type something'. Below the search bar are filters for 'Categories', 'Language', 'Font properties', and a checkbox for 'Show only variable fonts'. The main content area displays two font families: 'Roboto' by Christian Robertson and 'Roboto Serif' by Commercial Type, Greg Gazdowicz. Each family card shows a preview of the font in a large text block: 'Almost before we knew it, we had left the ground.' The Roboto card also indicates '12 styles'.

The screenshot shows the Google Fonts interface for the Roboto font family. At the top, there's a navigation bar with 'Roboto' selected, and buttons for 'Download family', 'Select styles', 'Glyphs', 'About', and 'License'. Below this, there are three examples of the font: 'Light 300' (normal weight), 'Light 300 italic' (italic weight), and 'Regular 400' (normal weight). Each example has a 'Select this style' button. To the right, a sidebar titled 'Selected family' lists the chosen styles: 'Review', 'Roboto', 'Light 300', and 'Regular 400'. It also includes 'Add more styles' and 'Remove all' buttons. Below this is a section for 'Use on the web' with code for embedding the font via '@import'. At the bottom of the sidebar are 'API docs' and 'Download all' buttons.

Seleccionar los estilos de la fuente que utilizaré en mi web

Elegir la opción "@import"

Copiar el @import url (...);

Pegarlo en la primera línea del archivo style.css

# Fonts seguras e importadas

**font-family** es la propiedad para cambiar el tipo de fuente

**Fonts seguras:** están disponibles en todos los navegadores y no es necesario importarlas.

**Fonts importadas:** externas, traídas desde librerías como Google Fonts.

```
style.css
```

```
.p1 {  
    font-family: "Times  
New Roman", Times,  
serif;  
}
```

Fuentes alternativas:  
se especifican  
separadas por comas

```
style.css
```

```
@import  
url( https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,300;0,400;1,  
400&display=swap );
```

```
.p1 {  
    font-family: "Roboto", san  
s-serif;  
}
```

Importante: siempre poner **fuentes alternativas** en caso de que la principal no cargue.

# Fonts en CSS -propiedades-

**font style** y **font-weight**: suelen usarse para incorporar *cursiva* o **negrita**. Su valor por default es "normal".

**font-size**: puede definirse a través de valores absolutos (px) o relativos (como em o rem).

style.css

```
.p1 {  
    font-style: italic;  
    color: red;  
}  
  
.p2 {  
    font-weight: bold;  
    color: #C70039;  
}
```

style.css

```
.p3 {  
    font-size: 32px;  
}  
  
.p4 {  
    font-size: 2em;  
}  
  
.p5 {  
    font-size: 60%;  
}
```

La propiedad *color* sirve para modificar el color de fuente en las etiquetas de texto.

# ¡Paciencia!

Por el momento **nos alcanza con saber que podemos modificar estas propiedades.** Sobre los elementos "padres", las herencias y el tamaño de ventana del navegador hablaremos más en la próxima clase.

Aprender a usar font-size en unidades relativas nos permitirá crear una web *responsive* que se ajuste a diferentes dispositivos y navegadores.

style.css

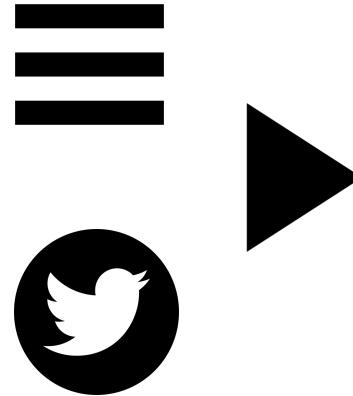
```
.p1 {  
    font-style: italic;  
}  
  
.p2 {  
    font-weight: bold;  
}
```

style.css

```
.p3 {  
    font-size: 32px;  
}  
  
.p4 {  
    font-size: 2em;  
}  
  
.p5 {  
    font-size: 60%;  
}
```

# Sumar íconos

Hay muchos tipos de **íconos**, aquí les compartimos un [tutorial](#) de la librería **Font Awesome** que explica cómo importarlos y modificar tamaño y color. También pueden conseguir íconos gratuitos de forma más sencilla en [este sitio](#).

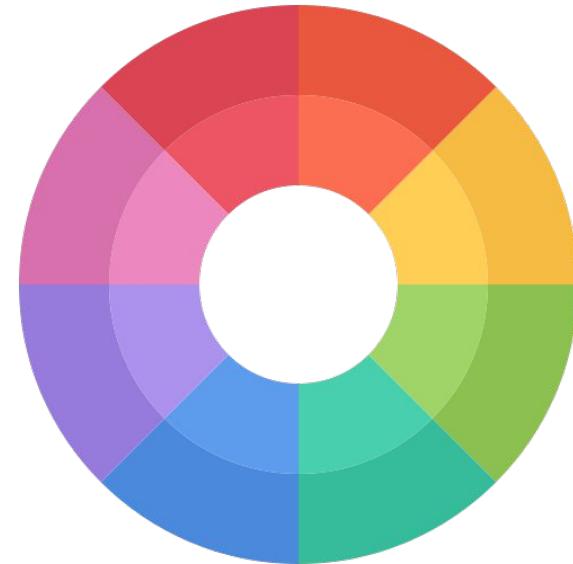


# Paletas de color

Elegir las combinaciones de colores para la web no es tarea fácil, ¡por suerte tenemos sitios que nos asisten!

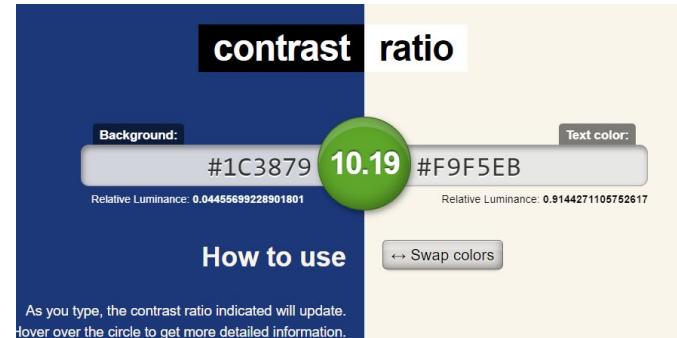
[colors.co](http://colors.co)

[colorhunt.co](http://colorhunt.co)

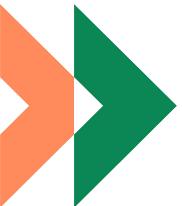


# Contraste

Cuando combinemos colores en nuestra web, es importante cuidar el contraste entre los colores de texto y fondo para que el contenido sea fácil de leer. Ante la duda, existen herramientas que nos permiten comprobar si los colores que estamos superponiendo tienen buen contraste.



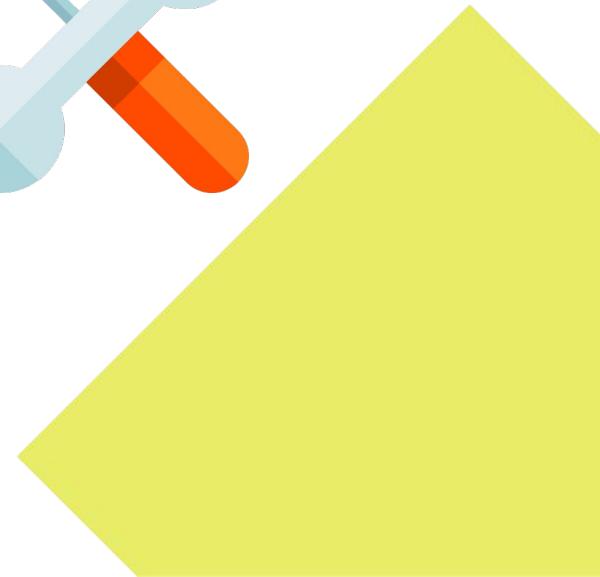
# Emmet



# Emmet, una gran herramienta

Viene instalado por defecto en VSCode y nos va a ser de ayuda a la hora de escribir código HTML.

Conoceremos algunos atajos para aprovechar esta herramienta.





# Atajos básicos en Emmet

- Crear estructura básica documento HTML: escribir ! y presionar **Enter** o TAB
- Crear una etiqueta: empezar a escribir el nombre de la misma y presionar Enter
- Crear etiqueta y agregarle una clase: **etiqueta.clase** y presionar Enter, por ejemplo, si escribimos **p.info** el resultado es: `<p class="info"></p>`
- Crear etiqueta y agregarle un id: **etiqueta#id** por ejemplo, **section#noticias crea <section id="noticias></section>**
- Escribir un elemento varias veces: **etiqueta\*cantidad** , por ejemplo: **li\*5 crea 5 elementos <li>**
- Anidar elementos: **elemento padre>elemento hijo**, por ejemplo: **ul>li\*3 crea un <ul> con 3 <li> dentro**

# ¡Desafío!

Utilizando operadores como `>`, `+`, `^`, `*` podemos escribir rápidamente estructuras complejas con la ayuda de Emmet. **¿Qué estructuras resultaran de estas combinaciones?**

**¡Tienen 2min para abrir el editor de código y probarlas!**

emmet.html

- `nav>ul#menu>li*4`
- `article>h2+p+figure>img`
- `header+main>section+section^footer`

emmet.html

```
<nav>  
  
  <ul id="menu">  
    <li></li>  
    <li></li>  
    <li></li>  
    <li></li>  
  </ul>  
  
</nav>
```

emmet.html

```
<article>  
  
  <h2></h2>  
  
  <p></p>  
  
  <figure>  
    <img src="" alt="">  
  </figure>  
  
</article>
```

emmet.html

```
<header></header>  
  
<main>  
  <section></section>  
  <section></section>  
</main>  
  
<footer></footer>
```

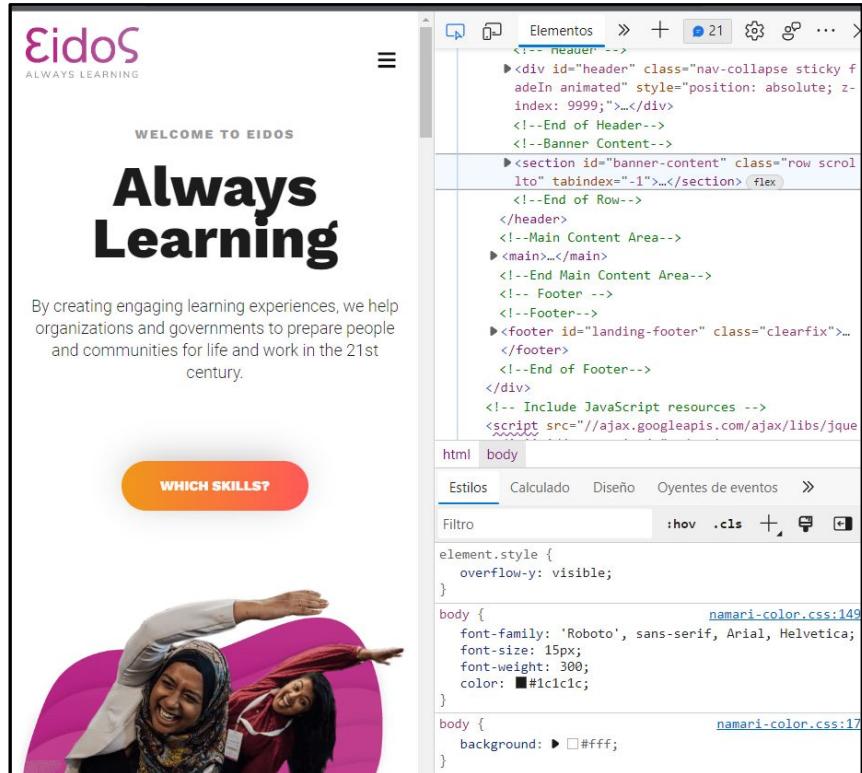
En este enlace tienen más atajos 😊

# Inspector



# Herramientas del navegador

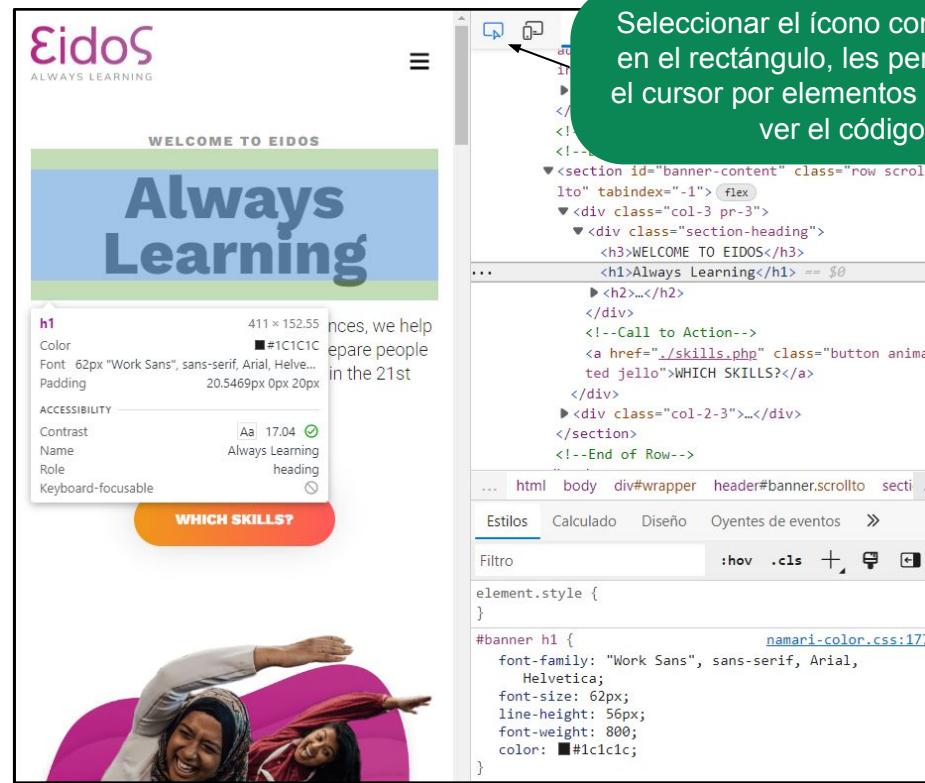
Al abrir una página podemos clickear "**Ctrl + Shift + i**" y eso me abre las "Herramientas para desarrolladores" (también llamadas "el inspector") en las cuales puedo visualizar y manipular el código HTML, CSS y JS



# ¿Para qué usamos el inspector?

Si seleccionamos este ícono de la flechita, podemos clickear sobre elementos de la web y el inspector nos muestra su código HTML y CSS.

Podemos jugar y probar cosas con el código HTML y CSS sin que este quede guardado.



# Styles

En la pestaña styles podremos ver y probar cosas en el código CSS que se aplica a cada elemento. Si scrolleamos hasta abajo nos encontraremos con esta representación de los elementos, con sus medidas y espaciados...



The screenshot shows the Chrome DevTools Styles tab. At the top, there are tabs for Styles (which is selected), Computed, Layout, and Event Listeners. Below the tabs is a Filter input field and a toolbar with icons for :hover, .cls, +, and others.

The main area displays the following CSS code:

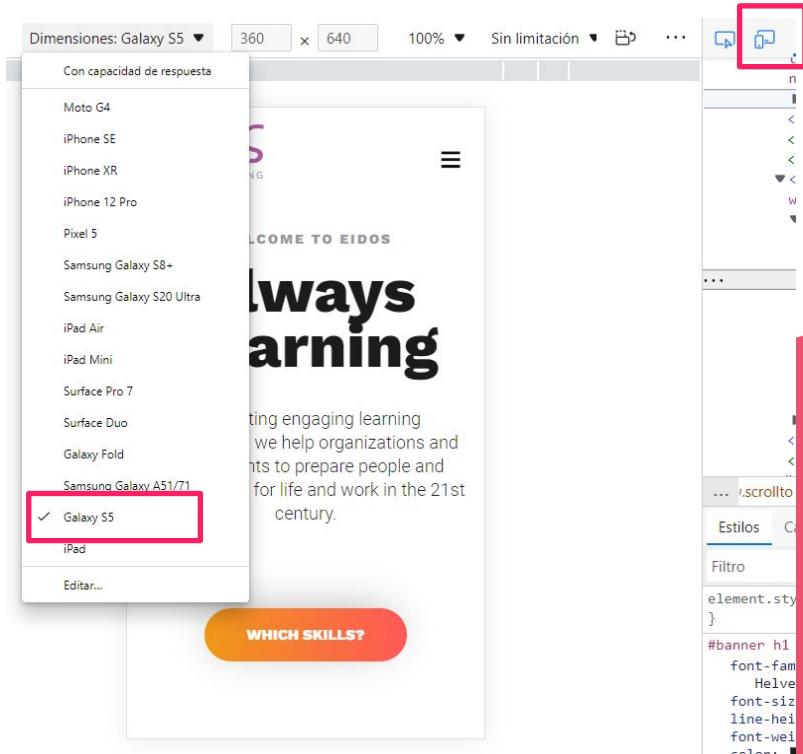
```
body {  
    overflow-x: hidden;  
    -webkit-font-smoothing: antialiased;  
}  
  
Pseudo ::before element  
*, *:before, *:after {  
    box-sizing: border-box;  
}  
  
Pseudo ::after element  
*, *:before, *:after {  
    box-sizing: border-box;  
}
```

Below the code, there is a detailed box model visualization for an element. The box is represented by concentric rectangles with the following dimensions:

- Outermost rectangle (orange): position: 0; margin: -; border: -; padding: -; height: 0; width: 0.
- Second rectangle (yellow): position: 0; margin: -; border: -; padding: -; height: 0; width: 0.
- Third rectangle (green): position: 0; margin: -; border: -; padding: -; height: 0; width: 0.
- Innermost rectangle (blue): 1178x936 pixels.

# Ver el sitio en diferentes tamaños

Además, si clickeamos ese ícono con forma de celular y tablet, el sitio se pone en tamaño *mobile* (celular), y podemos elegir de entre una lista de dispositivos el tamaño en el cual queremos verlo.



# Unidades de medida



# Tipos de unidades

Usamos dos tipos de unidades para determinar tamaños: **absolutas** y **relativas**. Las absolutas siempre tienen el mismo tamaño en todos los dispositivos, en CSS usamos una sola de este tipo: el pixel(px). Las relativas, justamente dependen de otro tamaño: la ventana del navegador, el tamaño de un elemento contenedor o el tamaño de la letra (% , vh, vw, rem, em).

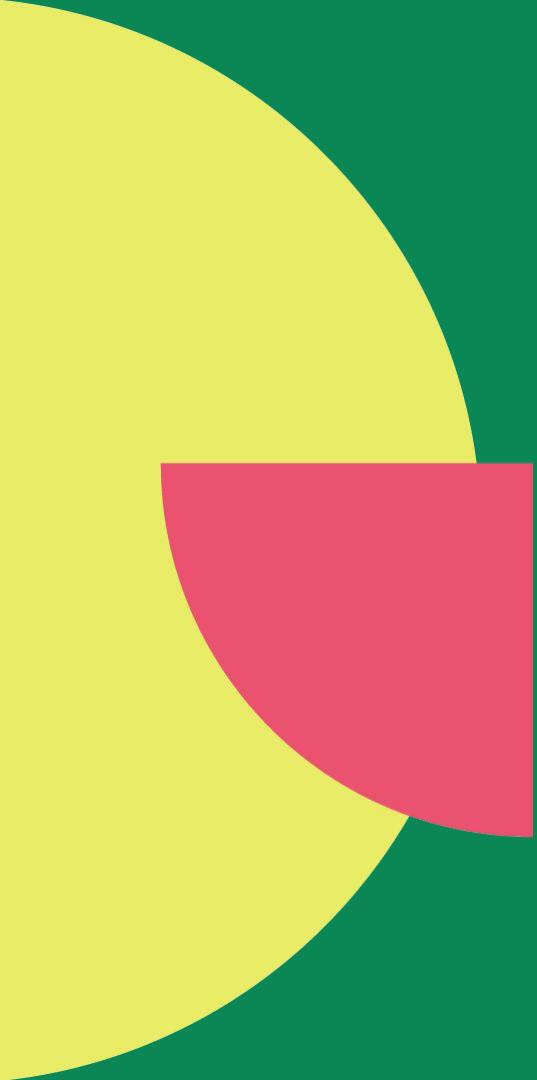
styles.css

```
article {  
    width: 50%;  
    font-size: 18px;  
}  
  
button {  
    height: 2rem;  
    padding: 0.5rem;  
}  
  
section {  
    min-height: 100vh;  
}
```

# Tabla de unidades

unidad	descripción
px (absoluta)	El píxel es la más sencilla de usar: el valor indicado es el que se verá en pantalla. Desventaja: no es <i>responsive</i> , si aplicamos un tamaño en px puede funcionar en pantallas grandes, pero no en un celular, por ejemplo.
% (relativa)	Porcentaje, es relativo al tamaño del elemento contenedor.
em (relativa)	Relativa al tamaño de fuente actual del elemento, si el font-size es 10px, entonces 1em = 10px
rem (relativa)	Relativa al font-size del elemento raíz (<html>), por defecto es 16px. 1rem = 16px a menos que cambiemos el font-size de <html>
vh y vw (relativas)	"Viewport units". Relativas al ancho(vw) y alto(vh) de la ventana del navegador donde se está visualizando el sitio.

👉 [Más sobre unidades de medida](#)



# **Contenido ampliatorio: Formularios**

# Formularios



Los formularios son herramientas que podemos incluir en un sitio para permitir a los usuarios insertar información, tomar decisiones o cambiar el comportamiento de una aplicación.

formularios.html

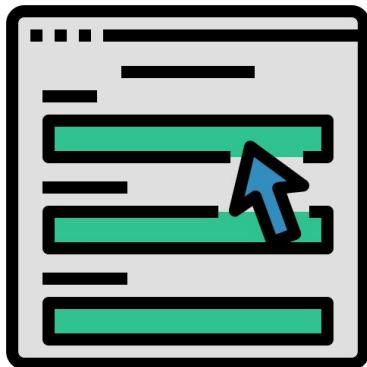
```
<form>
  <label for="user">
    Usuarie:
    <input type="text"
      id="user"
      name="user"
      placeholder="Ingresá tu nombre de usuarie"
    />
  </label>
  <label for="password">
    Contraseña:
    <input type="password"
      id="password"
      name="password"
      placeholder="Ingresá tu contraseña"
    />
  </label>
  <button type="submit">
    Ingresar
  </button>
</form>
```



A screenshot of a web browser window showing the rendered HTML form. The title bar says "Formularios". The form contains two text input fields labeled "Usuarie:" and "Contraseña:", each with a placeholder message. Below the inputs is a "Ingresar" button.

A continuación,  
conoceremos etiquetas y  
atributos para construir  
formularios con HTML.

# Etiquetas y atributos



Hay muchos tipos de `<input>`, y también elementos como `<textarea>` o `<select>` que utilizamos en los formularios.

Además, vamos a emplear varios atributos en estas etiquetas, algunos son opcionales y otros requeridos.

# Etiquetas

**<form>** Define el formulario. Dentro se anidarán los demás elementos. Es el punto de partida.

**<label>** Se usan para identificar cada campo del formulario. Se vinculan con los inputs utilizando el atributo **for**, cuyo valor será el **id** que le pongamos al input. También colocando el input adentro del label.

**<input>** Se usan para crear los campos donde de ingresa la información. El tipo de dato que puede recibir depende de su atributo **type**.

Hay muchos tipos de input: text, email, password, date, number, range, checkbox, radio, color, file...

formularios.html

```
<form action="">
  <label for="email">
    Email:
    <input type="email" name="email" id="email">
  </label>
  <label for="country">País:
    <select name="country" id="country">
      <option value="AR">Argentina</option>
      <option value="UR">Uruguay</option>
      <option value="CH">Chile</option>
    </select>
  </label>
  <label for="message">Mensaje:
    <textarea
      name="message"
      id="message"
      cols="30"
      rows="10">
    </textarea>
  </label>
  <button type="submit">
    ¡Enviar!
  </button>
</form>
```

# Etiquetas

**<select>** Crea una lista desplegable de opciones. Dentro van los **<option>** que contienen las opciones.

**<textarea>** Se usa para ingresar texto más extenso.

**<button>** Genera un botón. Al agregarle un atributo **type="submit"**, cuando es clickeado envía el formulario y se recarga la página.

Si no está implementado el envío del formulario, es decir, si *no tenemos conectado Front End con Back End* o algún servicio para el envío de formularios, no se enviará a ningún lugar.

formularios.html

```
<form action="">
    <label for="email">
        Email:
        <input type="email" name="email" id="email">
    </label>
    <label for="country">País:
        <select name="country" id="country">
            <option value="AR">Argentina</option>
            <option value="UR">Uruguay</option>
            <option value="CH">Chile</option>
        </select>
    </label>
    <label for="message">Mensaje:
        <textarea
            name="message"
            id="message"
            cols="30"
            rows="10">
        </textarea>
    </label>
    <button type="submit">
        ¡Enviar!
    </button>
</form>
```

## Atributos

**action** Aquí se puede colocar una ruta (url) donde se enviarán los datos para ser procesados (al Back End). Los formularios también pueden enviarse de otras formas utilizando JavaScript, pero por ahora solo nos preocupa el maquetado 😊

**type** Define el tipo de dato que acepta el input. Algunos son: text, email, url, number, file, date, checkbox, radio.

**name** Identifica el dato ingresado en el campo, es necesario para luego poder procesar ese dato en el Back End.

Cuando usamos **inputs de tipo "radio"**, les colocamos a todos el **mismo valor en name** para que funcionen correctamente, ya que es un "multiple choice" y debe enviarse *una sola respuesta*.

formularios.html

```
<form action="">
  <label for="email">
    Email:
    <input type="email" name="email" id="email">
  </label>
  <label for="country">País:
    <select name="country" id="country">
      <option value="AR">Argentina</option>
      <option value="UR">Uruguay</option>
      <option value="CH">Chile</option>
    </select>
  </label>
  <label for="message">Mensaje:
    <textarea
      name="message"
      id="message"
      cols="30"
      rows="10">
    </textarea>
  </label>
  <button type="submit">
    ¡Enviar!
  </button>
</form>
```

# Atributos

**id** Colocaremos id a los campos del formulario para identificarlos.

**placeholder** Sugerencia para completar el campo.

**value** Es el valor del campo. En los inputs de texto, los usuarios completan este valor, pero **en inputs de tipo radio o checkbox y en los <options>** debemos usarlo para contener el valor que enviará ese campo.

**required** Hace que sea obligatorio completar el campo.

**rows y cols** Alto y ancho del textarea medido en filas y columnas de caracteres. Esto lo podemos manejar con CSS también.

## formularios.html

```
<form action="">
  <label for="email">
    Email:
    <input type="email" name="email" id="email">
  </label>
  <label for="country">País:
    <select name="country" id="country">
      <option value="AR">Argentina</option>
      <option value="UR">Uruguay</option>
      <option value="CH">Chile</option>
    </select>
  </label>
  <label for="message">Mensaje:
    <textarea
      name="message"
      id="message"
      cols="30"
      rows="10">
    </textarea>
  </label>
  <button type="submit">
    ¡Enviar!
  </button>
</form>
```

# Tipos de input

¡Existen muchos tipos de input! En este enlace pueden conocerlos y probarlos:

👉 [Tipos de inputs](#)

## INPUT TYPES IN HTML

<input type="text">

xyz

<input type="password">

\*\*\*\*\*

<input type="radio">

No  Yes

<input type="checkbox">

<input type="button">

Button

<input type="color">

Blue

<input type="email">

xyz@gmail.com

<input type="file">

Choose File image.jpg

<input type="hidden">

<input type="image">

Submit Image

<input type="number">

898

# Aclaración

Utilizaremos nuestros conocimientos sobre HTML para maquetar un formulario, pero será *solo la fachada*.

Para validar los datos ingresados y realizar el envío se necesita aplicar JavaScript, además no contamos con un Back End donde enviar los datos para ser procesados. Por esto, al clickear "enviar" no sucederá nada... por ahora 😊

# Muchas gracias

CÓDIGO  
PA' LANTE

Organizan



Socio estratégico



Con el apoyo de

