

DIPLOMADO

Front End

Introducción

UVM

# Objetivo

Conocer los fundamentos de HTML y CSS para construir páginas web, selectores CSS, posicionamiento y flexbox así como el diseño web responsive

# Temario

01

Introducción  
a HTML

03

Introducción  
a FlexBox

02

Introducción  
a CSS

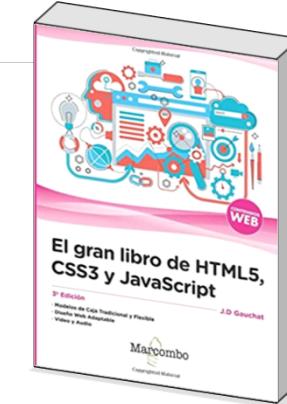
04

Layouts  
Responsivos

# Bibliografía del Módulo

Gauchat Juan Diego. (2017)

**El gran libro de HTML5, CSS y Javascript. 3<sup>a</sup> Edición.**  
Barcelona, España: Macombo.

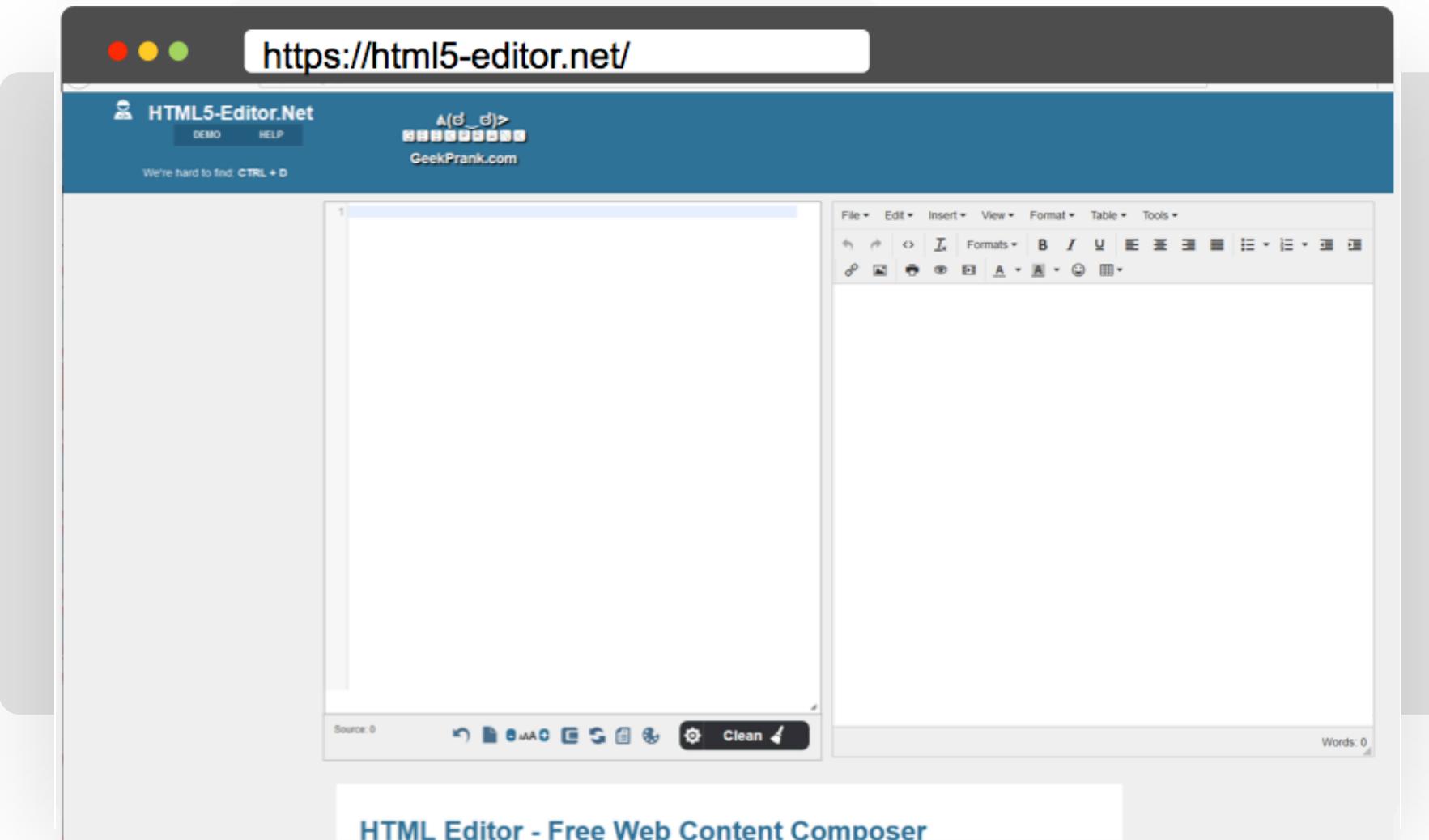


Lujan Castillo José Dimas. (2016)

**HTML5, CSS y Javascript. Crea tu web y apps con el estándar de desarrollo.** San Fernando de Henares, España: RC Libros.



# Herramientas de Trabajo en Clase



# Herramientas de Trabajo en Clase

The screenshot shows a web browser window displaying the [collabedit.com](http://collabedit.com/) website. The URL is visible in the address bar. The page has a dark header with three colored dots (red, yellow, green) and a navigation menu with links: new, download, history, invite, and Log In. Below the header, the site's logo "collabedit" and the tagline "simple collaborative text" are displayed. A user identifier "9ayr2" is shown in a box. The main content area contains a text editor with two lines of code:  
1 <b> Esto es un texto en negrilla </b>  
2  
A vertical scroll bar is visible on the right side of the text editor. To the right of the editor, there are several sidebar components: "Programming Language" set to "html", "Collaborators" listing "Javier T", "Chat" showing a message from Javier T about changing the language, and a "Chat Here" input field. At the bottom of the editor area, there are position indicators: "Position: Ln 2, Ch 1" and "Total: Ln 2, Ch 37".

DIPLOMADO

Front End

HTML

Introducción al HTML

UVM

TEMA 1

# ¿Qué es HTML?

**HTML es una tecnología** que se complementa con otras formando así el conjunto que constituye una base sólida para cualquier proyecto web: HTML5, CSS3 y JavaScript.



- **HTML es un conjunto de normas** que debemos de seguir.
- **Su objetivo es desplegar un sitio web de forma correcta e indicada.**
- **El significado de HTML** es HyperText Markup Language.
- Quiere decir: lenguaje de marcas de hipertexto. O de una forma sencilla "**lenguaje de etiquetado**".

# ¿Qué es HTML?

- **HTML no es propiamente un lenguaje de programación** (Luján Castillo, 2016).
- Un lenguaje de programación tiene variables, ciclos, tipos de datos y otras propiedades exclusivas.
- HTML en cambio, tiene como base las etiquetas, los atributos y los estilos.



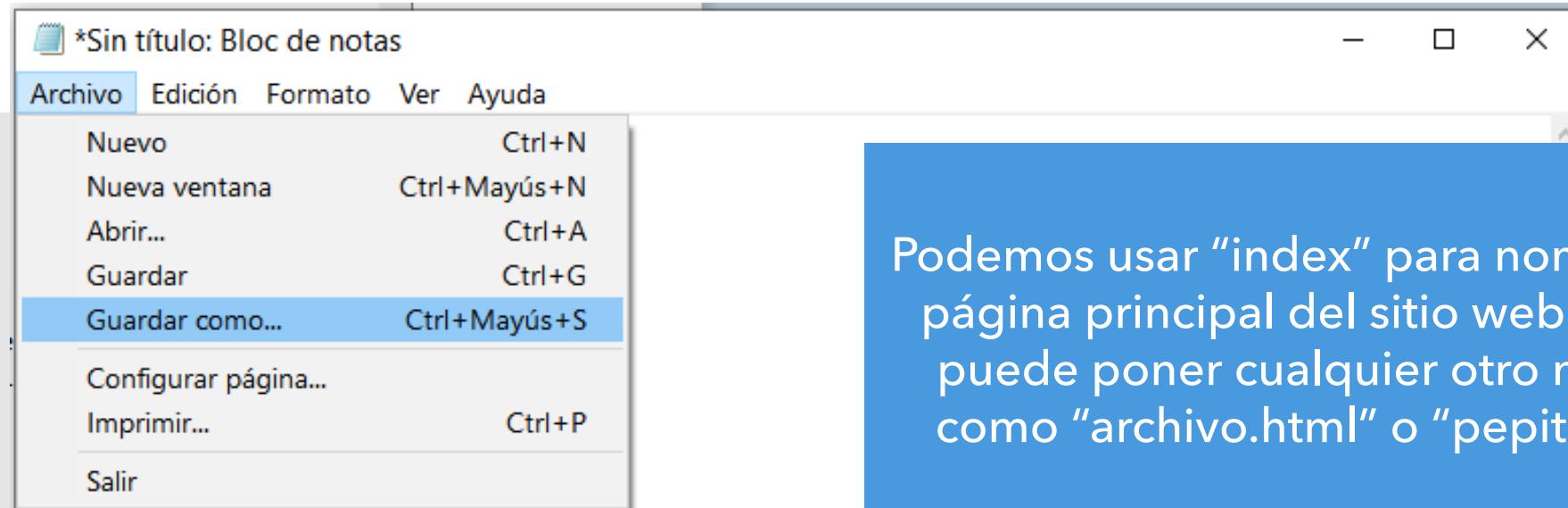
# ¿Qué es HTML?

- HTML5 provee estructura, estilo y funcionalidad.
- HTML5 es considerado la combinación de HTML, CSS y Javascript y actúan como una sola unidad.
- HTML está a cargo de la estructura
- CSS presenta esa estructura y su contenido en la pantalla y
- Javascript hace el resto que es extremadamente significativo.



(Gauchat, 2012)

# Conceptos



Podemos usar "index" para nombrar a la página principal del sitio web pero se puede poner cualquier otro nombre como "archivo.html" o "pepito.html"

PARA CREAR UN ARCHIVO HTML5 o HTML, solo es necesario abrir un editor de texto, seleccionar guardar como y asignar la extensión del archivo como ".html"

# Conceptos

- Uno de los conceptos básicos de HTML **son las etiquetas**.
- La mayoría de ellas se debe cerrar con el símbolo de "/".
- No todas las etiquetas se cierran colocando la misma etiqueta con la diagonal.
- Estructura general de una etiqueta en HTML:

**<apertura> Contenido </cierre>**

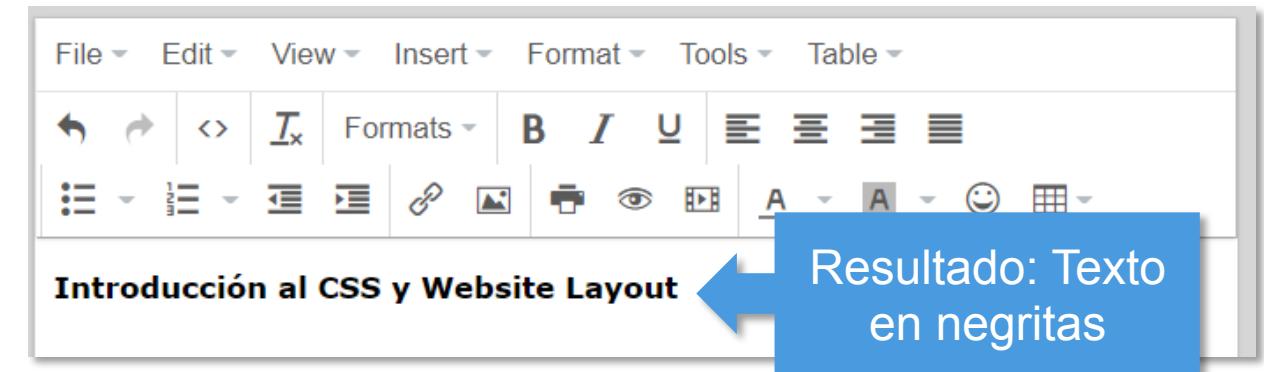
**<apertura atributo="valor"> Contenido </cierre>**

# Conceptos

- Ejemplo: Etiqueta “**”(significa bold o negrita).**
- Escriba lo siguiente:

A screenshot of a code editor window. The title bar says "Source: 51". The code area contains two lines of HTML: "1 <b>Introducción al CSS y Website Layout</b>" and "2". Two blue arrows point upwards from blue boxes containing the text "Apertura <b>" and "Cierre de la etiqueta con </b>".

```
<b>Introducción al CSS y Website Layout</b>
```



# Estructura Básica de HTML

## Ejemplo

```
1. <!DOCTYPE html>
2. <html lang="es">
3.   <head>
4.     <title>Mi web en HTML5</title>
5.   </head>
6.   <body>
7.     </body>
8. </html>
```

- Este es un ejemplo de la estructura básica de un sitio web e HTML5
- La página aparecerá aun cuando hacen falta algunas etiquetas básicas pero no estrictamente necesarias para crear un sitio web.
- Abra un nuevo archivo de texto, escriba este código sin los número de renglón) y guárdelo con la extensión .html
- ¿Cómo se ve al abrirlo con un navegador?

# Estructura Básica de HTML

## Ejemplo

```
1. <!DOCTYPE html>
2. <html lang="es">
3.   <head>
4.     <title>Mi web en HTML5</title>
5.   </head>
6.   <body>
7.     </body>
8. </html/>
```

Todos los elementos tiene un cierre con `</head>` /`</body>` y `</html>`

El “DOCTYPE” es lo primero que se escribe en HTML5 para declarar el tipo de página. Debe ser la primera línea del archivo.

Etiqueta “html” con un atributo asignado “=es” que significa que esta página está en ESPAÑOL.

La etiqueta `<head>` identifica la primera parte de la página y lo que contiene el encabezado.

Identificamos de manera global al “título” de la página con la etiqueta `<title>`  
**NO SE DESPLIEGA EN EL NAVEGADOR** pero si aparece en las listas de “favoritos”

Body es la etiqueta que marca la parte “visible” de la página.

# Estructura Básica de HTML

## Etiquetas básicas

Algunas de las etiquetas que se consideran elementales para que un documento de HTML funcione correctamente.

Las etiquetas `<meta>` son consideradas como información específica para navegadores y buscadores en internet.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<meta charset="utf-8">`
5. `<meta name="author" content="Juan Pérez">`
6. `<meta name="description" content="Ejemplo">`
7. `<meta="keywords" content="HTML5">`
8. `<link rel="stylesheet" href="estilo.css"`
9. `</head>`

# Estructura Básica de HTML

## Etiquetas básicas

<meta charset="utf-8">

Identifica el tipo de caracteres de la página. Si no se coloca el correcto, no se podrían mostrar caracteres como los acentos

<meta name="author" content="juan">

Esta etiqueta permite colocar info. sobre el autor de la página que estamos viendo.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. <meta name="author" content="Juan Pérez">
6. <meta name="description" content="Ejemplo">
7. <meta="keywords" content="HTML5">
8. <link rel="stylesheet" href="estilo.css"
9. </head>

# Estructura Básica de HTML

## Etiquetas básicas

<meta name= <“description“>

Contiene la descripción del contenido de la página. Uso:  
<meta name=“description” content=“Ejemplo básico”>

<meta=“keywords” content=“HTML5”>

Contiene las palabras clave que relacionan a la página con lo que buscan los usuarios en internet.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=“utf-8”>
5. <meta name=“author” content=“Juan Pérez”>
6. <meta name=“description” content=“Ejemplo”>
7. <meta=“keywords” content=“HTML5”>
8. <link rel=“stylesheet” href=“estilo.css”>
9. </head>

# Estructura Básica de HTML

## Etiquetas básicas

```
<link rel="stylesheet" href="estilo.css"
```

<link> permite usar archivos externos, en la mayoría de los casos enlaza con aquellos que tienen estilos, conocidos como archivos CSS o de “Cascada de estilos”

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. <meta name="author" content="Juan Pérez">
6. <meta name="description" content="Ejemplo">
7. <meta="keywords" content="HTML5">
8. <link rel="stylesheet" href="estilo.css"
9. </head>

# Estructura Básica de HTML

## Atributos

- Algunas etiquetas HTML pueden múltiples atributos .
- Estos sirven para especificar un valor a la etiqueta que lo contiene.
- Sintaxis:

```
<etiqueta atributo="valor">
```

Y cuando hay mas de un atributo:

```
<etiqueta atributo1="valor1" atributo2="valor2">
```

# Estructura Básica de HTML

## Atributo "id"

- Permite asignar un identificador único.
- Este atributo "id" puede ser utilizado desde CSS o Javascript.

Ejemplo: asignamos a cada párrafo (con la etiqueta "parrafo de texto" <p>) un identificador:

1. <p id="nombre"> Javier </p>
2. <p id="twitter"> @xterres </p>
3. <p id="email"> xterres@gmail.com </p>

- Los atributos "id" NO ACEPTAN caracteres especiales
- Deben ser únicos
- Se debe utilizar únicamente letras y números

# Estructura Básica de HTML

## Atributo "dir"

- Habilita la alineación (dirección) de los elementos
- Puede tener 2 valores
  - **RTL** Coloca el elemento de derecha a izquierda
  - **LTR** Coloca el elemento de izquierda a derecha
- Ejemplo

The screenshot shows a web-based HTML editor interface. On the left, there's a toolbar with standard editing tools like 'Archivo', 'Formatos', and various text styling buttons. Below the toolbar is a text area containing two paragraphs of text: 'Segundo Texto' and 'Primer texto'. A large red arrow points upwards from the word 'Segundo' towards the code editor. Another red arrow points upwards from the word 'Primer' towards the code editor. To the right of the text area is a code editor window titled 'LIMPIAR'. It displays the following HTML code:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title> Ejemplo de atributo dir </title>
</head>
<body>
<p dir="RTL"> Primer texto </p>
<p dir="LTR"> Segundo Texto </p>
</body>
</html>
```

# Estructura Básica de HTML

## Atributo "class"

- Permite asignar el nombre de una clase a un elemento seleccionado
- Es muy similar a lo que sucede con la programación orientada a objetos
  - Una clase puede tener muchos elementos
- Ejemplo:

The screenshot shows a web editor interface. On the left, there's a preview area with two paragraphs: "Esto es rojo" in red and "Esto es azul" in blue. A large red arrow points upwards from the preview area towards the code editor. The code editor on the right displays the following HTML and CSS code:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <style>
5     .claserrojo {color:red}
6     .claseazul{color:blue}
7   </style>
8   <title> Ejemplo de clases</title>
9 </head>
10 <body>
11   <p class="claserrojo"> Esto es rojo </p>
12   <p class= "claseazul"> Esto es azul </p>
13 </body>
14 </html>
```

The code editor has a toolbar at the top with various icons for file operations, editing, and styling. A red button labeled "LIMPIAR" is visible in the toolbar.

# Estructura Básica de HTML

## Atributo "style"

- Habilita la opción de colocar estilo de forma directa sobre el elemento que lo contenga
- Lo ideal es tener el estilo separado en otro archivo.
- Ejemplo

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

<h1>Este es un encabezado </h1>
<p>Este es un párrafo.</p>

</body>
</html>
```

**Este es un encabezado**

Este es un párrafo.



# Estructura Básica de HTML

## Organización de la página

En el Encabezado se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del sitio web o la página.

En la Barra de Navegación casi todos los desarrolladores ofrecen un menú o lista de enlaces para facilitar la navegación a través del sitio.



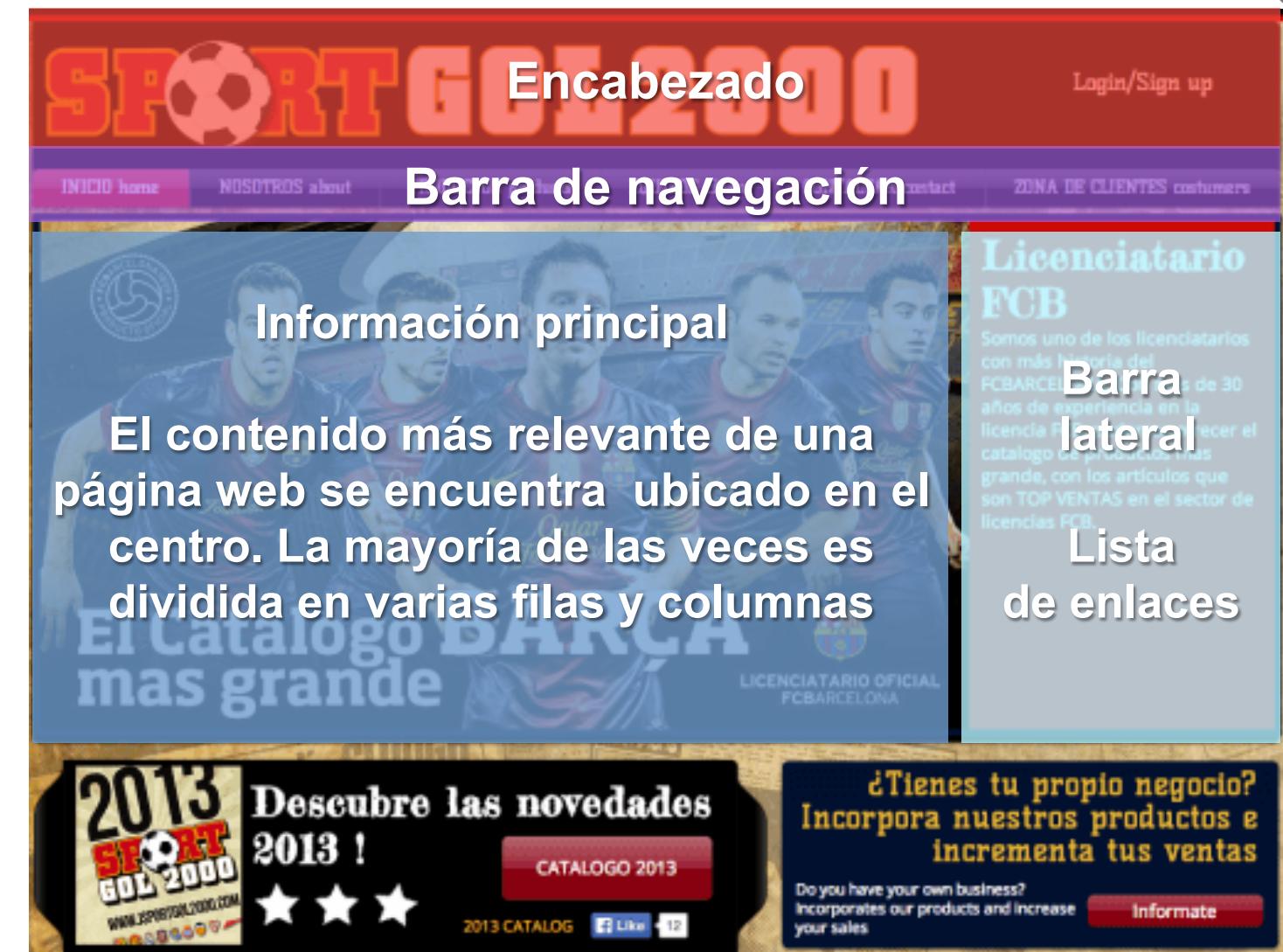
# Estructura Básica de HTML

## Organización de la página

### Información Principal y Barra Lateral

Esta sección es flexible y normalmente se adapta de acuerdo a las necesidades del diseñador insertando más columnas, dividiendo cada columna en bloques más pequeños o generando diferentes distribuciones.

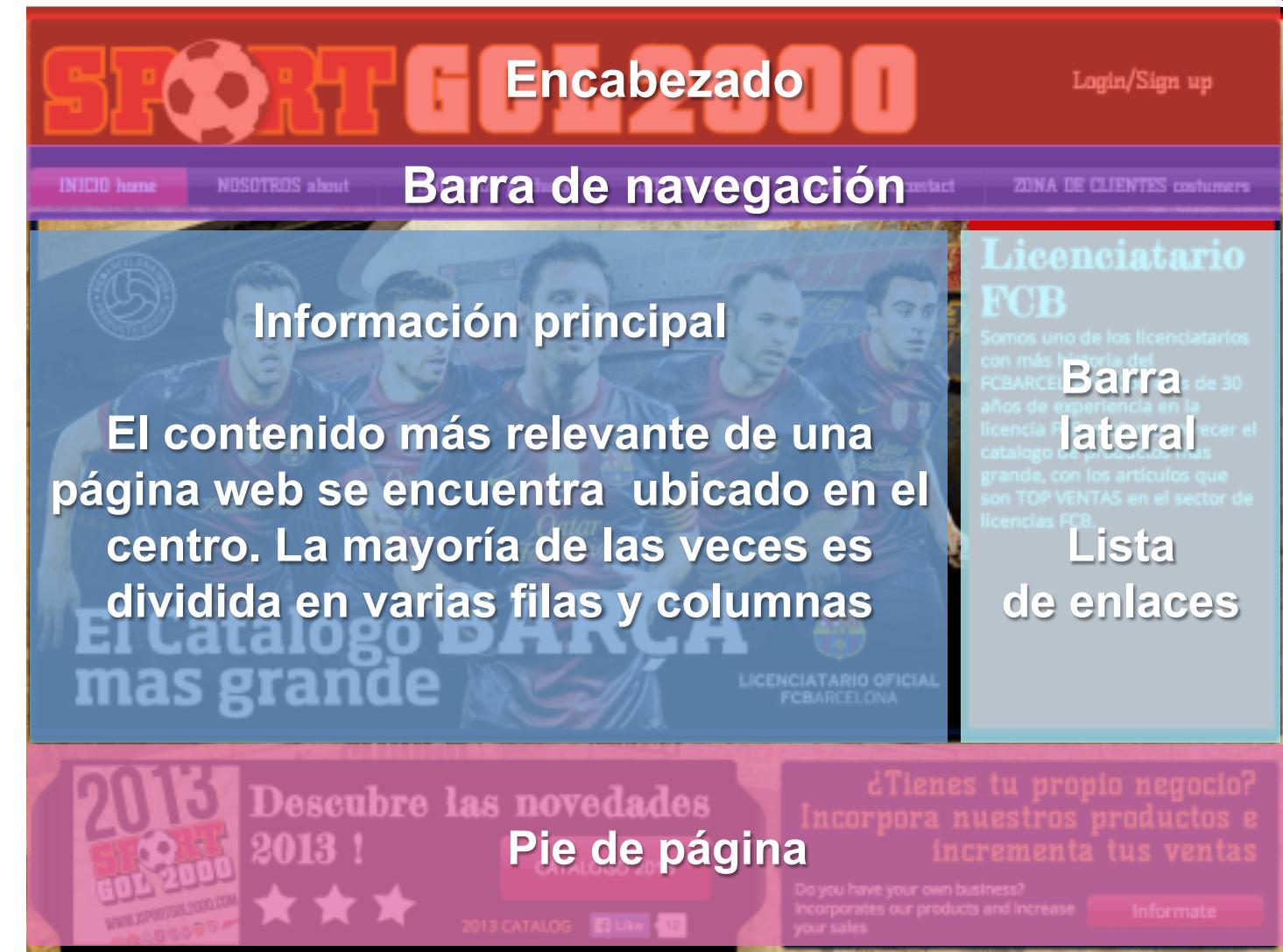
La Barra Lateral podría mostrar una lista de enlaces apuntando hacia cada uno de esos ítems.



# Estructura Básica de HTML

## Organización de la página

En el pie de página normalmente se muestra información acerca del sitio web, el autor o la empresa, además de algunos enlaces con respecto a reglas, términos y condiciones y toda información adicional que el desarrollador considere importante compartir.



# Estructura Básica de HTML

## Organización de la página

Esta simple representación ayuda a entender que cada sección definida en un sitio web tiene un propósito.

HTML5 considera esta estructura básica y provee nuevos elementos para diferenciar y declarar cada una de sus partes. Ahora es posible decir al navegador para qué es cada sección:



# Tipo de Programación para la Clase

Crea un grupo de archivos con los elementos básicos de HTML para que puedas ir modificando únicamente los elementos que vamos revisando en los ejercicios.

```
1. <!DOCTYPE html>
2. <html lang="es">
3. <head>
4. <meta charset="iso-8859-1">
5. <meta name="description" content="Ejemplo de HTML5">
6. <meta name="keywords" content="HTML5, CSS3, JavaScript">
7. <title>Este texto es el título del documento</title>
8. <link rel="stylesheet" href="misestilos.css">
9. </head>
10. <body>
11. <header>
12. <h1>Este es el título principal del sitio web</h1>
13. <p>Empezamos a aprender HTML</p>
14. </header>
15. </body>
16. </html>
```

# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

<html> y </html> indican el comienzo del código HTML y su final.

```
<!DOCTYPE html>
<html lang="es"> ←
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
  </body>
</html> ←
```

**Este es el título principal del sitio web**

# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

`<head>` representa la primera sección. En su interior se realizan diversas definiciones del sitio web.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
</body>
</html>
```



**Este es el título principal del sitio web**



# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

**<body>** representa la parte visible de todo documento.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
</body>
</html>
```



**Este es el título principal del sitio web**



# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

<nav> Genera la Barra de navegación. Va dentro de las etiquetas <body> y después </header>

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
<nav>
<ul>
<li>principal</li>
<li>fotos</li>
<li>videos</li>
<li>contacto</li>
</ul>
</nav>
</body>
</html>
```

Este es el título principal del sitio web

- principal
- fotos
- videos
- contacto



# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

<section> es el elemento que especifica las columnas Información Principal y Barra Lateral

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
<nav>
<ul>
<li>principal</li>
<li>fotos</li>
<li>videos</li>
<li>contacto</li>
</ul>
</nav>
<section>
----- [sección] -----
</section>
</body>
</html>
```



**Este es el título principal del sitio web**

- principal
- fotos
- videos
- contacto

----- [sección] -----



# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

**<aside>** define el espacio para enlaces y pequeñas reseñas que son solo una ayuda para la navegación.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
<nav>
<ul>
<li>principal</li>
<li>fotos</li>
<li>videos</li>
<li>contacto</li>
</ul>
</nav>
<section>
----- [sección] -----
</section>
<aside>
<blockquote>Mensaje número uno</blockquote>
<blockquote>Mensaje número dos</blockquote>
</aside>
</body>
</html>
```

**Este es el título principal del sitio web**

- principal
- fotos
- videos
- contacto

----- [sección] -----

Mensaje número uno

Mensaje número dos



**<aside>** podría estar ubicado del lado derecho o izquierdo pero la etiqueta no tiene una posición predefinida. Solo describe la información que contiene, no el lugar dentro de la estructura

# Estructura Básica de HTML

Escribe el siguiente código, incluyendo los elementos estructurales. Corrobora el resultado.

<footer>es el final del cuerpo. También se usa para representar el final de diferentes secciones

```
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
<nav>
<ul>
<li>principal</li>
<li>fotos</li>
<li>videos</li>
<li>contacto</li>
</ul>
</nav>
<section>
----- [sección] -----
</section>
<aside>
<blockquote>Mensaje número uno</blockquote>
<blockquote>Mensaje número dos</blockquote>
</aside>
<footer>
Derechos Reservados &copy; 2010-2011
</footer>
</body>
</html>
```

**Este es el título principal del sitio web**

- principal
- fotos
- videos
- contacto

----- [sección] -----

Mensaje número uno

Mensaje número dos

Derechos Reservados © 2010-2011



# Estructura Básica de HTML

## Estructura del cuerpo

- Las etiquetas `<article>` se ubican dentro `<section>`.
- `<article>` son hijos, al igual que cada elemento dentro de las etiquetas `<body>` es su hijo.
- Del mismo modo que cada elemento hijo del cuerpo, las etiquetas `<article>` son ubicadas una sobre otra.



**<article>** no está limitado por su nombre. Agrupa porciones de información que están relacionadas entre sí mas allá de su naturaleza.

Como una parte independiente del documento, el contenido de cada **<article>** podrá tener su propia estructura delimitada con **<header>** y **<footer>**.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
<nav>
<ul>
<li>principal</li>
<li>fotos</li>
<li>videos</li>
<li>contacto</li>
</ul>
</nav>
<section>
<article>
<header>
<h1>Título del mensaje uno</h1>
</header>
Este es el texto de mi primer mensaje
<footer>
<p>comentarios (0)</p>
</footer>
</article>
<article>
<header>
<h1>Título del mensaje dos</h1>
</header>
Este es el texto de mi segundo mensaje
<footer>
<p>comentarios (0)</p>
</footer>
</article>
</section>
<aside>
<blockquote>Mensaje número uno</blockquote>
<blockquote>Mensaje número dos</blockquote>
</aside>
<footer>
Derechos Reservados &copy; 2010-2011
</footer>
</body>
</html>
```



# Estructura Básica de HTML

Este es el título principal del sitio web

- principal
- fotos
- videos
- contacto

Título del mensaje uno

Este es el texto de mi primer mensaje  
comentarios (0)



Título del mensaje dos

Este es el texto de mi segundo mensaje  
comentarios (0)



Mensaje número uno

Mensaje número dos

Derechos Reservados © 2010-2011

# Estructura Básica de HTML

**<hgroup>** agrupa las etiquetas de los encabezados ya que **<header>** fue creado para contener también otros elementos como tablas de contenido o textos cortos y logos.

Para construir este tipo de cabeceras, podemos aprovechar el resto de las etiquetas “H”, como **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>**.

Por propósitos de procesamiento interno y evitar generar múltiples secciones durante la interpretación del documento por parte del navegador, estas etiquetas deben ser agrupadas juntas por **<hgroup>**

```
<header>
<hgroup>
<h1>Título del mensaje dos</h1>
<h2>Subtítulo del mensaje dos</h2>
</hgroup>
<p>publicado 31-05-2021</p>
</header>
```

**Título del mensaje dos** ←  
**Subtítulo del mensaje dos** ←  
publicado 31-05-2021

- ▶ **<hgroup>** es necesario cuando tenemos un título y subtítulo o más etiquetas “H” juntas en la misma cabecera. Si solo dispone de una etiqueta **<h1>** o la etiqueta **<h1>** junto con datos adicionales, **no tiene que agrupar estos elementos juntos**.

## <hgroup>

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
<nav>
<ul>
<li>principal</li>
<li>fotos</li>
<li>videos</li>
<li>contacto</li>
</ul>
</nav>
<section>
<article>
<header>
<hgroup>
<h1>Título del mensaje uno</h1>
<h2>Subtítulo del mensaje uno</h2>
</hgroup>
<p>publicado 10-05-2021</p>
</header>
Este es el texto de mi primer mensaje
<footer>
<p>comentarios (0)</p>
</footer>
</article>

```

```

<article>
<header>
<hgroup>
<h1>Título del mensaje dos</h1>
<h2>Subtítulo del mensaje dos</h2>
</hgroup>
<p>publicado 11-10-2021</p>
</header>
Este es el texto de mi segundo mensaje
<footer>
<p>comentarios (0)</p>
</footer>
</article>
</section>
<aside>
<blockquote>Mensaje número uno</blockquote>
<blockquote>Mensaje número dos</blockquote>
</aside>
<footer>
Derechos Reservados &copy; 2020-2021
</footer>
</body>
</html>

```

# Estructura Básica de HTML

**Este es el título principal del sitio web**

- principal
- fotos
- videos
- contacto

**Título del mensaje uno**

**Subtítulo del mensaje uno**

publicado 10-05-2021

Este es el texto de mi primer mensaje

comentarios (0)

**Título del mensaje dos**

**Subtítulo del mensaje dos**

publicado 11-10-2021

Este es el texto de mi segundo mensaje

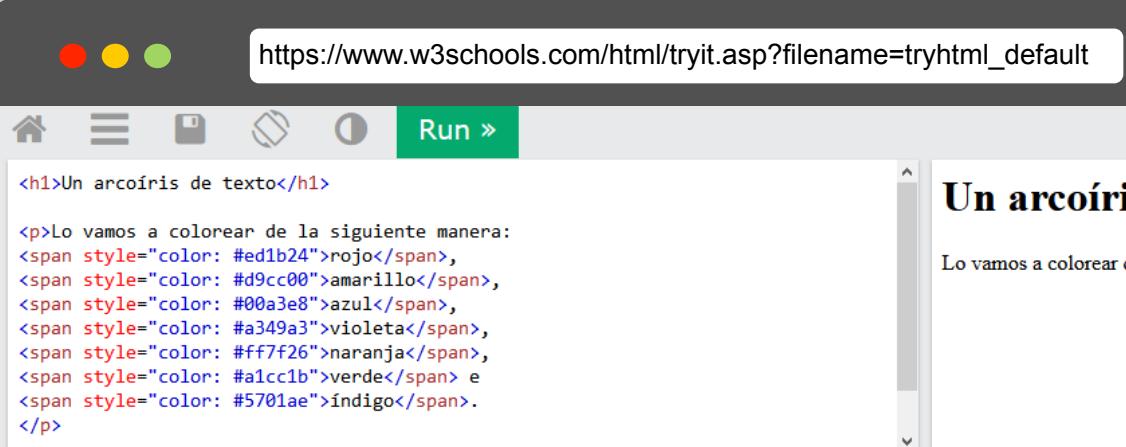
comentarios (0)

Mensaje número uno

Mensaje número dos

Derechos Reservados © 2020-2021

# Otras Etiquetas



A screenshot of a web browser window. The address bar shows the URL [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default). The browser interface includes standard buttons for back, forward, and refresh, along with a "Run" button. The left panel displays the HTML code:

```
<h1>Un arcoíris de texto</h1>
<p>Lo vamos a colorear de la siguiente manera:<br>
<span style="color: #ed1b24">rojo</span>,<br>
<span style="color: #d9cc00">amarillo</span>,<br>
<span style="color: #00a3e8">azul</span>,<br>
<span style="color: #a349a3">violeta</span>,<br>
<span style="color: #ff7f26">naranja</span>,<br>
<span style="color: #a1cc1b">verde</span> e<br>
<span style="color: #5701ae">índigo</span>.</p>
```

The right panel shows the rendered output: "Un arcoíris de texto". Below it, a descriptive text reads: "Lo vamos a colorear de la siguiente manera: rojo, amarillo, azul, violeta, naranja, verde e índigo."

`<span>` es un contenedor genérico sin un significado semántico en particular. Comúnmente usado en la creación de páginas con propósitos de estilo.



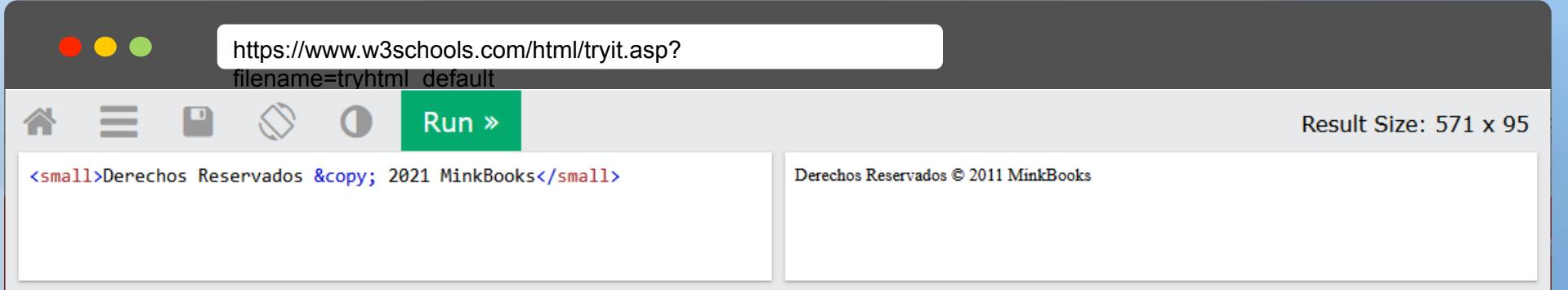
A screenshot of a browser window. The address bar shows the URL [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default). The browser interface includes standard buttons for back, forward, and refresh, along with a "Run" button. The left panel displays the HTML code:

```
<span> Mi <mark> coche </mark> es rojo </span>
```

The right panel shows the rendered output: "Mi coche es rojo".

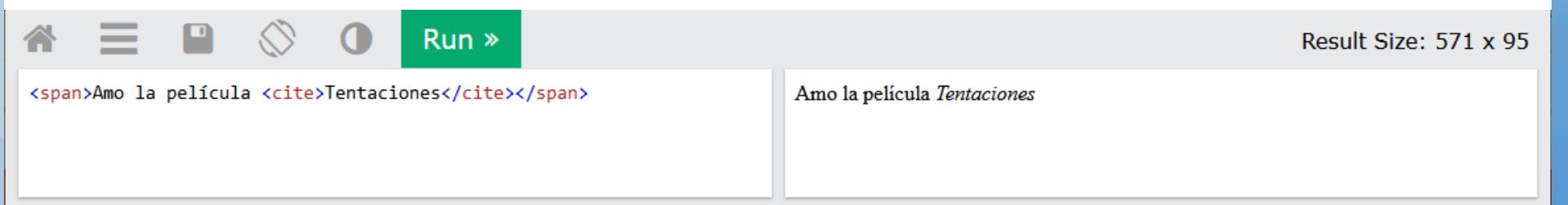
`<mark>` resalta parte de un texto. En algunos navegadores, esta palabra será resaltada con un fondo amarillo por defecto, pero siempre podemos sobrescribir estos estilos con los nuestros utilizando CSS.

# Otras Etiquetas



A screenshot of a web browser window. The address bar shows the URL [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default). Below the address bar is a toolbar with icons for home, menu, file, and refresh, followed by a green "Run" button. To the right of the toolbar, it says "Result Size: 571 x 95". The main content area displays two snippets of HTML code and their rendered results. On the left, the code is: <small>Derechos Reservados © 2021 MinkBooks</small>. On the right, the result is: Derechos Reservados © 2021 MinkBooks. The rendered result is smaller than the original code.

<small> es utilizado con la intención de presentar cualquier texto con letra pequeña.



A screenshot of a web browser window, similar to the one above. The address bar and toolbar are identical. The main content area displays two snippets of HTML code and their rendered results. On the left, the code is: <span>Amo la película <cite>Tentaciones</cite></span>. On the right, the result is: Amo la película *Tentaciones*. The word "Tentaciones" is italicized, indicating it is a citation.

<cite> se utilizan para realizar una cita y encierran el título de un trabajo, como un libro, una película, una canción, etc.

# Otras Etiquetas



The screenshot shows a browser window with the URL [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default). The browser interface includes a toolbar with icons for home, refresh, save, and run, and a status bar indicating a result size of 588 x 145. The left pane displays the HTML code:

```
<figure>

Esta es la imagen del primer mensaje
</figcaption>
</figure>
```

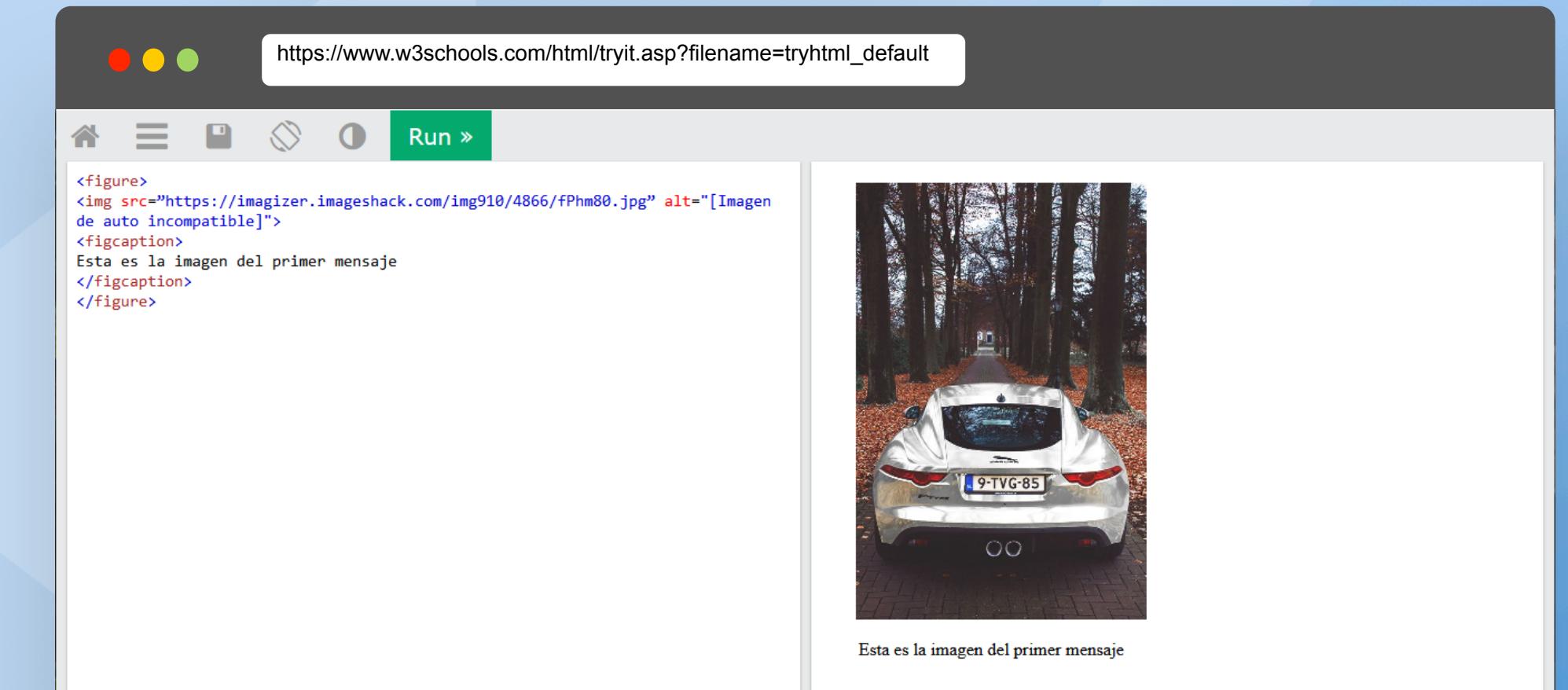
The right pane shows the rendered output: a placeholder text "[Imagen de auto incompatible]" followed by the caption "Esta es la imagen del primer mensaje".

`<figure>` representa una porción independiente de contenido (imágenes, diagramas o videos) referenciadas desde el contenido principal.

`<figcaption>` utilizado para mostrar una leyenda o pequeño texto relacionado con el contenido de un elemento `<figure>`, como la descripción de una imagen.

`<img>` representa una imagen en el documento. No siempre se muestra la imagen a la que el elemento hace referencia: si el usuario elige no mostrar la imagen, o el navegador es incapaz de mostrarla porque no es válida o soportada. En ese caso, el navegador la reemplazará con el texto definido en el atributo alt.

# Otras Etiquetas



The screenshot shows a web browser window with the URL [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default). The browser interface includes a title bar with three colored dots (red, yellow, green), a toolbar with icons for home, refresh, and save, and a green "Run" button. The main content area displays the following HTML code:

```
<figure>

<figcaption>
Esta es la imagen del primer mensaje
</figcaption>
</figure>
```

To the right of the code, there is a caption and an image. The caption reads "Esta es la imagen del primer mensaje". Below the caption is a silver Jaguar car parked on a paved path through a forest with fallen leaves. The car's license plate is 9-TVG-85.

Resultado cuando <img> si puede representar una imagen en el documento.

# Otras Etiquetas



The screenshot shows a web browser window with the URL [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_default](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default). The browser interface includes a toolbar with icons for home, menu, file, edit, and run, along with a green "Run »" button. The result size is listed as 658 x 298. On the left, the HTML code is displayed:

```
<article>
<header>
<hgroup>
<h1>Título</h1>
<h2>Subtítulo</h2>
</hgroup>
<p>Primer párrafo</p>
<hr/>
<p>Segundo párrafo </p>
<p>Mostramos el <em> texto </em>por<q>Javier Terrés</q></p>
<p>La fecha del día de hoy es: </p>
<time datetime="2021-06-01"> 1 de junio de 2021 </time>
</header>
</article>
```

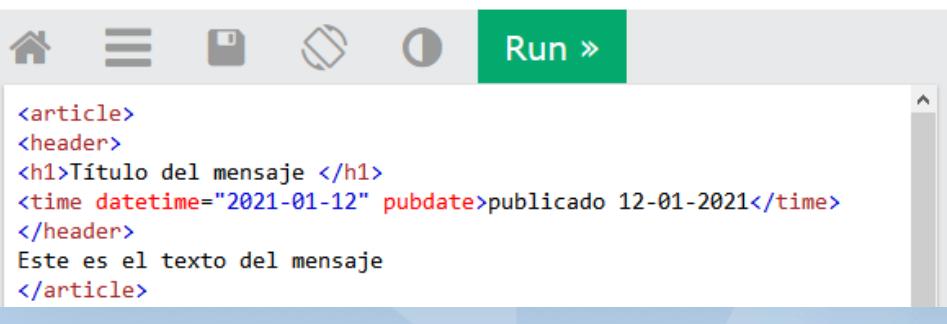
The right side shows the rendered HTML with styling applied:

**Título**  
**Subtítulo**  
Primer párrafo

---

Segundo párrafo  
Mostramos el *texto* por «Javier Terrés»  
La fecha del día de hoy es:  
1 de junio de 2021

`<time>` permite declarar un texto comprensible para humanos, navegadores y bots, que representa fecha y hora.



The screenshot shows a web browser window with the same URL as the previous one. The browser interface includes a toolbar with icons for home, menu, file, edit, and run, along with a green "Run »" button. The result size is listed as 571 x 176. On the left, the HTML code is displayed:

```
<article>
<header>
<h1>Título del mensaje </h1>
<time datetime="2021-01-12" pubdate>publicado 12-01-2021</time>
</header>
Este es el texto del mensaje
</article>
```

The right side shows the rendered HTML with styling applied:

**Título del mensaje**  
publicado 12-01-2021  
Este es el texto del mensaje

# Estructura Básica de HTML

`<span>` es un contenedor genérico sin un significado semántico en particular.

- Se usa en la creación de páginas de estilo junto con `<style>` y `<class>`
- También puede proveer atributos.
- Está diseñado para contener líneas simples de texto únicamente. Para bloques más grandes se usa `<div>`.

```
<h1>Pintando el arcoíris</h1>
```

```
<p>Esta vez usaremos los siguientes colores: <span style="color: #ed1b24">rojo</span>, <span style="color: #d9cc00">amarillo</span>, <span style="color: #00a3e8">azul</span>, <span style="color: #a349a3">violeta</span>, <span style="color: #ff7f26">naranja</span>, <span style="color: #a1cc1b">verde</span> e <span style="color: #5701ae">índigo</span>.</p>
```

## Pintando el arcoíris

Esta vez usaremos los siguientes colores: rojo, amarillo, azul, violeta, naranja, verde e índigo.

# Estructura Básica de HTML

<div> es un contenedor genérico sin un significado semántico en particular.

- Comúnmente usado en la creación de páginas de estilo junto con los atributos <style> y <class>
- También puede resultar útil para proveer atributos comunes a los elementos en su interior, como por ejemplo lang o title.
- <div> se usa en los casos donde los elementos con mayor significado, como <main>, <header>, <footer>, <nav> o <article> no son aplicables.

```
<h1>Tim Berners-Lee</h1>
```

```
<div style="color: #040; font-style: italic">
```

```
<p>Timothy "Tim" John Berners-Lee es un científico de la computación británico, conocido por ser el padre de la Web. Estableció la primera comunicación entre un cliente y un servidor usando el protocolo HTTP en noviembre de 1989.</p>
```

```
<p>En octubre de 1994 fundó el Consorcio de la World Wide Web (W3C) con sede en el MIT, para supervisar y estandarizar el desarrollo de las tecnologías sobre las que se fundamenta la Web y que permiten el funcionamiento de Internet.</p>
```

```
</div>
```

## Tim Berners-Lee

Timothy "Tim" John Berners-Lee es un científico de la computación británico, conocido por ser el padre de la Web. Estableció la primera comunicación entre un cliente y un servidor usando el protocolo HTTP en noviembre de 1989.

En octubre de 1994 fundó el Consorcio de la World Wide Web (W3C) con sede en el MIT, para supervisar y estandarizar el desarrollo de las tecnologías sobre las que se fundamenta la Web y que permiten el funcionamiento de Internet.

# Estructura Básica de HTML

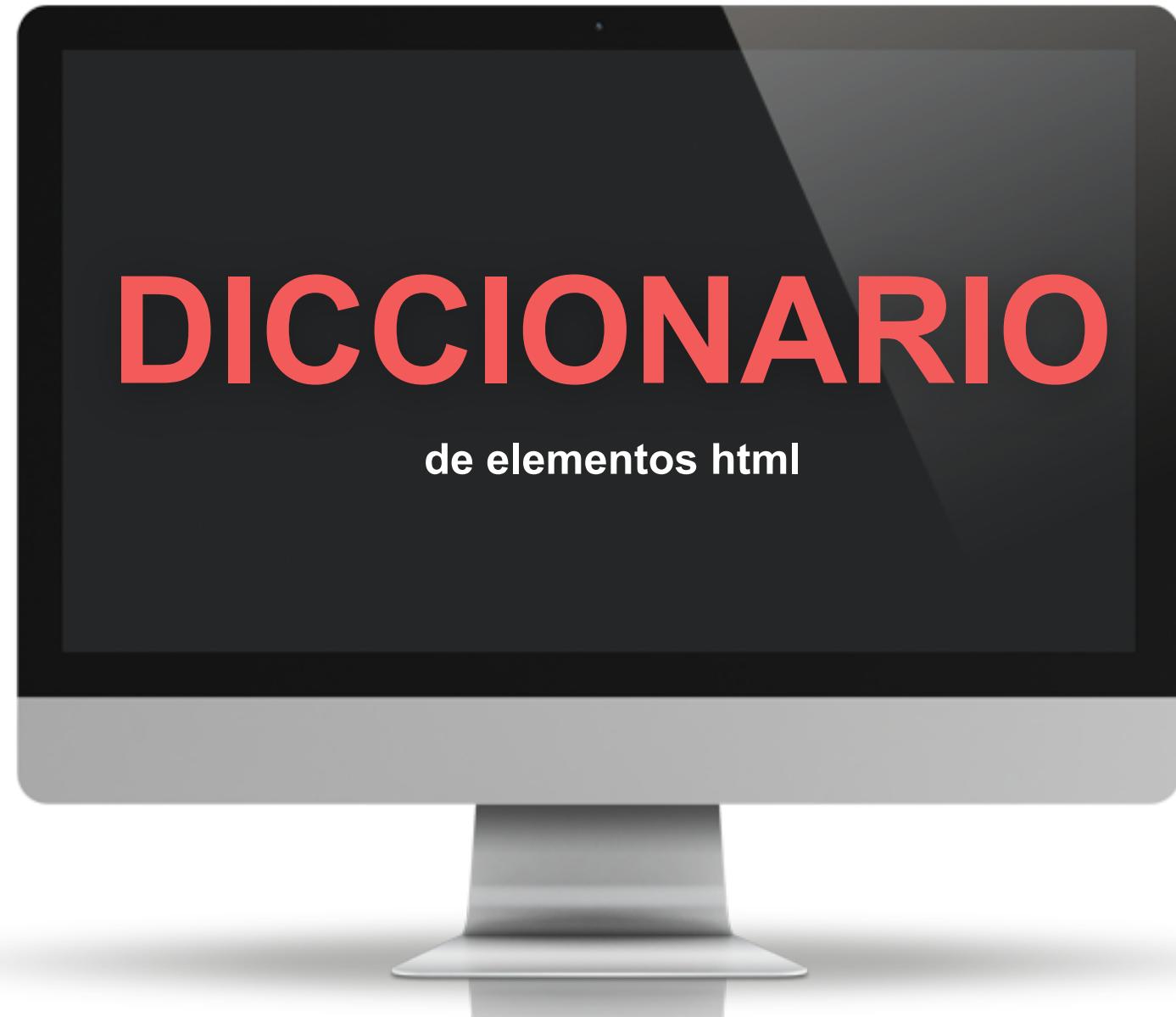
- El siguiente ejemplo muestra un par de párrafos `<p>` dentro por un elemento `<div>`.
- El propósito de `<div>` en este ejemplo, es el de aplicar un conjunto de estilos a ambos párrafos en una sola declaración.

```
<h1>La red informática mundial</h1>  
<div title="descripción inicial de la WWW">  
    <p>En informática, la World Wide Web (WWW) o red informática mundial  
    es un sistema de distribución de documentos de hipertexto o hipermedios  
    interconectados y accesibles vía Internet.</p>  
    <p>Con un navegador web, un usuario visualiza sitios web compuestos de  
    páginas web que pueden contener texto, imágenes, videos u otros  
    contenidos multimedia, y navega a través de esas páginas usando  
    hiperenlaces.</p>  
</div>
```

## La red informática mundial

En informática, la World Wide Web (WWW) o red informática mundial es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet.

Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.



## ELEMENTO RAÍZ

<html> Contenedor principal

## METADATOS DEL DOCUMENTO

<head> Sección de encabezado del documento

<title> Título del documento

<base> URI base para resolver URIs relativos

<link> Información relacional para el documento

<meta> Variable para el documento

<style> Atributos presentacionales

## SECCIONES

<body> El cuerpo del documento

<article> Contenido redistribuible

<section> Define una sección

<nav> Sección de navegación

<aside> Contenido relacionado

<h1> <h2> <h3> Encabezados de nivel 1 a 3

<h4> <h5> <h6> Encabezado de nivel 4 a 6

<hgroup> Agrupa encabezados consecutivos

<header> El encabezado de una sección

<footer> El pie de una sección

<address> Información de contacto del autor

## AGRUPACIÓN DE CONTENIDO

< p > Párrafo

< hr > Separador de contenido

< pre > Bloque de texto preformatado

< blockquote > Cita en forma de bloque

< ol > Lista ordenada

< ul > Lista no ordenada

< menu > Inserta una barra de menú

< li > Ítem de una lista

< dl > Lista de descripciones

< dt > Término en una lista de descripciones

< dd > Descripción en una lista de descripciones

< figure > Información autocontenido

< figcaption > Título para una figura

< main > Contenido principal de una sección

< div > Contenedor genérico para bloques de texto

## SEMÁNTICA DE TEXTO

`<a>` Hypervínculo

`<em>` Texto con énfasis

`<strong>` Texto con énfasis fuerte

`<small>` Comentario secundario

`<s>` Contenido que ya no es preciso o relevante

`<cite>` Cita o referencia

`<q>` Cita en forma de línea

`<dfn>` Término definido en el texto circundante

`<abbr>` Término abreviado

`<data>` Información apta para programas

`<time>` Fecha y/u hora

`<code>` Código de computadora

`<var>` Instancia de una variable

`<samp>` Ejemplo de salida del programa

`<kbd>` Texto ingresado por el usuario

`<sub>` Texto en subíndice

`<sup>` Texto en superíndice

`<i>` Texto desplazado de la prosa normal

`<b>` Texto desplazado del contenido circundante

`<u>` Anotaciones no textuales

`<mark>` Marca texto en otro documento

`<bdi>` Aisla texto para el formateo bidireccional

`<bdo>` Anula el algoritmo bidireccional

`<span>` Contenedor genérico de líneas de texto

`<br>` Quiebre de línea

`<wbr>` Oportunidad de quiebre de línea

## INFORMACIÓN TABULADA

`<table>` Tabla

`<caption>` El título de una tabla

`<colgroup>` Grupo de columnas

`<col>` Establece atributos para columnas

`<tbody>` Cuerpo de la tabla

`<thead>` Encabezado de la tabla

`<tfoot>` Pie de la tabla

`<tr>` Fila

`<td>` Celda común

`<th>` Celda de encabezado

## CONTENIDO INCRUSTADO

`<picture>` Imagen con múltiples orígenes

`<source>` Recurso de medios alternativo

`<img>` Imagen

`<iframe>` Contexto de navegación anidado

`<embed>` Inserta aplicaciones externas

`<object>` Inserta aplicaciones externas

`<param>` Parámetro para una aplicación externa

`<video>` Video

`<audio>` Audio

`<track>` Pista de texto para videos

`<map>` Mapa de imagen del lado cliente

`<area>` Sector en un mapa de imagen

## EDICIÓN

<ins> Texto agregado

<del> Texto eliminado

## FORMULARIOS

<form> Formulario

<label>Etiqueta de un control

<input>Control de entrada

<button>Botón

<select>Lista de opciones

<datalist>Sugerencias para los controles

## FORMULARIOS (Continua...)

<optgroup>Grupo de opciones en una lista

<option>Una opción en una lista

<textarea>Campo de texto multilínea

<output>La salida de un proceso

<progress>El progreso de una tarea

<meter>Una medida

<fieldset>Grupo de controles

<legend>El título de un grupo de controles

## ELEMENTOS INTERACTIVOS

<details> Contenido desplegable

<summary> Un resumen del contenido desplegable

<dialog> Cuadro de diálogo

## PROGRAMAS

<script> Incrusta programas

<noscript> Contenido alternativo para programas

<template> Plantilla para contenido a ser agregado

<slot> Espacio en componentes para insertar contenido

<canvas> Contenedor para gráficos dinámicos

DIPLOMADO

Front End

Introducción a CSS

UVM

TEMA 2

# ¿Qué es CSS?



**CSS es un lenguaje** que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc.

- **El significado de CSS** es Cascading Style Sheets.
- La traducción es “**hojas de estilo en cascada**”.
- CSS es entonces un lenguaje **DE REGLAS** que describe hojas de estilo y que trabajan sobre HTML.
- Este lenguaje es un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML.

# ¿Qué es CSS?

- Una regla CSS puede indicar como modificar la apariencia o posicionamiento de un elemento o de un grupo de elementos en HTML.
- Las reglas CSS se componen de tres partes: selector, propiedades y valores asignados.
- Cada regla puede tener varios valores asignados.

Regla

The diagram illustrates the structure of a CSS rule. It consists of a horizontal line with a bracket underneath it. The top part of the line is labeled "Regla". Below the line, the CSS code `h1 { color: #990000; font-size: 25px; }` is shown. Brackets below the code identify its components: the first bracket groups the selector `h1`, the second bracket groups the first property `color: #990000;`, the third bracket groups its value `#990000`, the fourth bracket groups the second property `font-size: 25px;`, and the fifth bracket groups its value `25px`. Below these labels are the terms: "Selector", "Propiedad 1", "Valor asignado 1", "Propiedad 2", and "Valor asignado 2".

```
h1 { color: #990000; font-size: 25px; }
```

Selector      Propiedad 1      Valor asignado 1      Propiedad 2      Valor asignado 2

Declaración

# ¿Qué es CSS?

- Es posible incorporar estilos en HTML dentro de las etiquetas por medio del atributo "style".

Ejemplo: `<p style="font-size: 20px">Mi texto</p>`

- Esto permite probar estilos y obtener una vista rápida de sus efectos, pero no es recomendado para aplicar estilos a todo el documento: implica repetir cada estilo en cada uno de los elementos, haciendo difícil su mantenimiento. Realiza la siguiente actividad:

The image shows a comparison between a plain text editor and a rich text editor (WYSIWYG). On the left, a text editor displays the following HTML code:

```
1 <h2> Muestra de colores </h2>
2 <p style="color: #ed1b24">rojo</p> rojo
3 <p style="color: #d9cc00">amarillo</p> amarillo
4 <p style="color: #00a3e8">azul</p> azul
5 <p style="color: #a349a3">violeta</p> violeta
6 <p style="color: #ff7f26">naranja</p> naranja
7 <p style="color: #a1cc1b">verde</p> verde
8 <p style="color: #5701ae">índigo</p> índigo
9
```

A blue callout box with an upward arrow points to the text editor, containing the instruction: "1. Escribe este código completo".

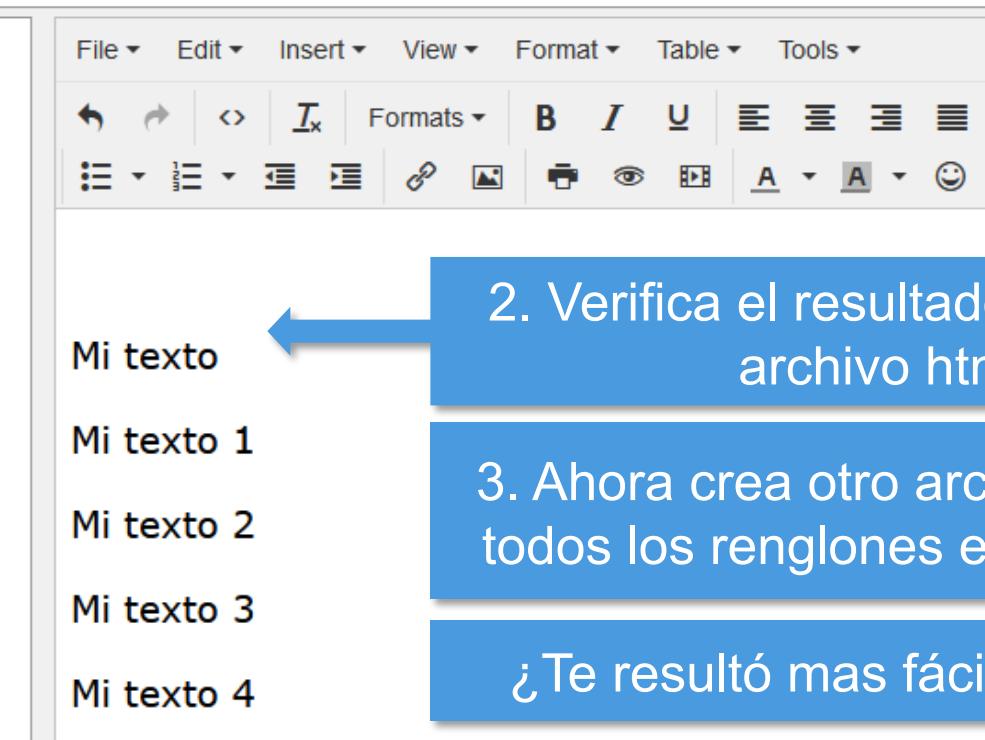
On the right, a WYSIWYG editor shows the same content. It has a toolbar at the top with various styling options like bold, italic, underline, etc. Below the toolbar, the heading "Muestra de colores" is displayed. To its right is a color palette with colored squares corresponding to the words: rojo (red), amarillo (yellow), azul (blue), violeta (purple), naranja (orange), verde (green), and índigo (indigo). A blue callout box with a leftward arrow points to the WYSIWYG editor, containing the instruction: "2. Verifica el resultado y guarda el archivo html.". Another blue callout box to the right contains the instruction: "3. Ahora crea otro archivo html con todos los renglones en color azul". At the bottom, another blue callout box contains the question: "¿Te resultó fácil y rápido realizar el cambio?".

# ¿Qué es CSS?

- Una alternativa es insertar los estilos en la cabecera del documento con `<style>` y luego usar referencias para afectar los elementos HTML como en el ejemplo. Realiza la actividad:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <title>Este texto es el título </title>
5 <style>
6   p { font-size: 20px }
7 </style>
8 </head>
9 <body>
10 <p>Mi texto</p>
11 <p>Mi texto 1</p>
12 <p>Mi texto 2</p>
13 <p>Mi texto 3</p>
14 <p>Mi texto 4</p>
15 </body>
16 </html>
```

1. Escribe este código completo



2. Verifica el resultado y guarda el archivo html.

3. Ahora crea otro archivo html con todos los renglones en altura 10px

¿Te resultó mas fácil el cambio?

- Sin embargo, es mejor mover todos los estilos a un archivo externo y luego utilizar `<link>` para insertar este archivo dentro de cada página que necesite esos estilos (CSS externo).

# CSS Extremo

## Ventajas

- Permite cambiar los estilos por completo simplemente incluyendo un archivo diferente y modificar o adaptar nuestros documentos a cada caso o dispositivo.
- Los archivos CSS son archivos de texto comunes.

## misestilos.css

```
1. body {  
2.     background-color:  
3.         powderblue;  
4. }  
5. h1 {  
6.     color: #00a3e8;  
7.     font-size: 25px;  
8. }  
9. p {  
10.    font-size: 20px;  
11. }
```

## Ejemplo.html

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3. <head>  
4. <title>Este texto es el título </title>  
5. <<link href="misestilos.css" rel="stylesheet">  
6. </head>  
7. <body>  
8. <h1>Ejemplo</h1>  
9. <p>Mi texto</p>  
10. </body>  
11. </html>
```

Realiza la prueba escribiendo ambos archivos (.css y .html). Guárdalos en el mismo directorio

- De ahora en adelante agregaremos estilos CSS al archivo llamado **misestilos.css**. Debe crear este archivo en el mismo directorio (carpeta) donde se encuentra el archivo HTML y copiar los estilos CSS en su interior para comprobar cómo trabajan.

# Resultado:

Activa el depurador en el menú [ ••• ] en las siguientes opciones:

- Mas herramientas
  - Herramientas de desarrollo

Navegador: Microsoft Edge

Ejemplo

Mi texto

Código

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Este es el titular del sitio</title>
    <link href="misestilos.css" rel="stylesheet">
  </head>
  <body> == $0
    <h1>Ejemplo</h1>
    <p>Mi texto</p>
  </body>
</html>
```

html body

Estilos Calculado Diseño Oyentes de eventos

Filtro :hover .cls +

```
element.style { }
body {
  background-color: powderblue;
}
body { hoja de estilo del agente de usuario
  display: block;
  margin: 8px;
}
```

# Conceptos básicos sobre estilos

## Referenciando con palabra clave

- Al declarar las reglas CSS utilizando la palabra clave del elemento afectamos cada elemento de la misma clase en el documento. Por ejemplo, la siguiente regla cambiará los estilos de todos los elementos <p>:

```
1. p{ font-size: 20px; }CSS
```

- Todos los textos envueltos en <p> etiquetas tendrán el tamaño de 20 pixeles.
- ¿Pero qué ocurre si solo necesitamos referenciar una etiqueta específica?

Respuesta: Se debe utilizar los atributos “id” y “class”.

# Conceptos básicos sobre estilos

## Referenciando con el atributo "id"

- El atributo "id" es el nombre que identifica al elemento de manera única en todo el documento.
- Para referenciar un elemento en particular usando "id" desde nuestro archivo CSS la regla debe iniciar con el símbolo "#" al frente del valor que usamos para identificar el elemento:

```
1. #texto1 { font-size: 20px; }
```

```
1. <!DOCTYPE html>
2. <html lang="es">
3. <head>
4. <title>Este texto es el título del documento</title>
5. <<link href="misestilos.css" rel="stylesheet">
6. </head>
7. <body>
8. <p id="texto1">Mi texto</p>
9. </body>
10. </html>
```

Nota: Eliminamos de nuestro archivo css la regla: p { font-size: 20px }

# Resultado:

The screenshot shows a Microsoft Edge browser window displaying a simple HTML page. The page title is "Este es el titular del sitio" and the document title is "Este texto es el título del documento". The main content area contains the heading "Ejemplo de asignación con id" and a paragraph "Mi texto 10px". The browser's developer tools are open, specifically the Elements and Styles panels. The Elements panel shows the HTML structure, including a 

element with the ID "texto1". The Styles panel shows the CSS rules applied to the elements. A blue bracket labeled "Código" points from the right side of the developer tools back towards the browser window, highlighting the connection between the code and its visual representation.

Navegador: Microsoft Edge

Ejemplo de asignación con id

Mi texto 10px

Este es el titular del sitio

Este texto es el título del documento

Archivo | D:/MEGA%20sync/1%20UVM/ejemplo%20css%20id.html

Elementos Consola ...

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Este texto es el título del documento</title>
    <link href="misestilos.css" rel="stylesheet">
  </head>
  <body> == $0
    <h1>Ejemplo de asignación con id</h1>
    <p id=""texto1"">Mi texto 10px</p>
  </body>
</html>
```

html body

Estilos Calculado Diseño Oyentes de eventos ...

Filtro :hov .cls +

```
element.style { }
body {
  background-color: powderblue;
}
body { hoja de estilo del agente de usuario
  display: block;
  margin: 8px;
}
```

Código

**misestilos.css:1**

# Conceptos básicos sobre estilos

## Referenciando con el atributo “class”

- “Class” Funciona mejor para definir estilos
- Puede ser asignado a cada elemento HTML en páginas con diseños similares:

1. .texto1 { font-size: 10px }

- Lleva un punto delante ya que es posible construir referencias más complejas. El ejemplo de abajo solo afecta a los elementos 

con class="texto1":

1. p.texto1 { font-size: 20px }

```
1. <!DOCTYPE html>
2. <html lang="es">
3. <head>
4. <title>Este texto es el título del documento</title>
5. <link href="misestilos.css" rel="stylesheet">
6. </head>
7. <body>
8. <h1 class="texto1">Atributo Class</h1>
9. <p>texto X sin class</p>
10. <p class="texto1">Mi texto 1</p>
11. </body>
12. </html>
```

Nota: solo conservamos el formato para `<body>` y las clases mencionadas

# Resultado:

The screenshot shows a Microsoft Edge browser window displaying a local HTML file. The page content includes an orange header "Asignacion con Class:", a black text "texto X sin class", and a pink text "Mi texto 1". The browser's developer tools are open, specifically the Elementos (Elements) tab, which shows the HTML structure. A blue bracket on the right side of the Elements tab covers the entire code area. A blue circle highlights the file path "misestilos.css:1" in the Styles panel, indicating the source of the applied styles.

Navegador: Microsoft Edge

Este texto es el título del documento

Archivo | D:/MEGA%20sync/1%20UVM/ejemplo%20css%20class.html

Asignacion con Class:

texto X sin class

Mi texto 1

Elementos Consola

Este texto es el título del documento

link href="misestilos.css" rel="stylesheet">

body

h1 class="texto1">Asignacion con Class:

p class="texto1">Mi texto 1

html body

Estilos Calculado Diseño Oyentes de eventos

Filtro :hover .cls + [ ]

element.style { }

body { background-color: powderblue; }

body { hoja de estilo del agente de usuario display: block; }

Código

misestilos.css:1

# Conceptos básicos sobre estilos

## Referenciando con cualquier atributo

Con este método, CSS3 permite referenciar elementos HTML a través de cualquier otro atributo.

1. p [name] { font-size: 20px }

CSS

Esta regla afecta a elementos <p> que tienen un atributo llamado “name”

Se puede ser mas específico condicionando el atributo a un valor:

1. p [name="mitexto"] { font-size: 20px }

CSS

Referenciamos elementos <p> que tienen un atributo “name” igual al valor “mitexto”.

Al utilizar este tipo referencia, se puede combinar con las siguientes expresiones:

- ^= “x” Asigna el estilo a todo elemento que el atributo “name” empiece con el valor x
- \$= “x” Asigna el estilo a todo elemento que el atributo “name” termine con el valor x
- \*= “x” Asigna el estilo a todo elemento que el atributo “name” que contenga con el valor x

# Conceptos Básicos sobre Estilos

## misestilos.css

```
1. body { background-color:  
    powderblue; }  
  
2. h1.texto1 {font-size: 10em;  
3.     color:#00ff00;}  
  
4. p[name^="rojo"]{color:#FF0000;}  
  
5. p[name$="grande"]  
6. {color:#0000FF;}  
  
7. p[name*="nia"]{color:#0000FF;  
    font-size:0.5em;}
```

Realiza la prueba escribiendo ambos archivos (.css y .html). Guárdalos en el mismo directorio

## Ejemplo.html

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3. <head>  
4. <title>Este texto es el título del documento</title>  
5. <link href="misestilos.css" rel="stylesheet">  
6. </head>  
7. <body>  
8. <h1 class="texto1">Mi Web</h1>  
9. <p name="texto1">Esto es un párrafo</p>  
10. <p name="rojo">Esto es un párrafo rojo</p>  
11. <p name="rojomini">Esto es un párrafo rojomini</p>  
12. <p name="azulgrande">Esto es un párrafo azul grande</p>  
13. <p name="miniazul">Esto es un párrafo miniazul</p>  
14. </body>  
15. </html>
```

# Resultado:

Este texto es el título del documento

Este texto es el título del documento

Archivo | D:/MEGA%20sync/1%20UVM/ejemplo%20css%20name.html

Mi Web

Esto es un párrafo

Esto es un párrafo rojo

Esto es un párrafo rojomini

Esto es un párrafo azul grande

Esto es un párrafo miniazul

Elementos Consola Orígenes

html lang="es">Este texto es el título del documento

# Mi Web

Esto es un párrafo

Esto es un párrafo rojo

Esto es un párrafo rojomini

Esto es un párrafo azul grande

Esto es un párrafo miniazul

Estilos Calculado Diseño Oyentes de eventos

Filtro :hov .cls +

element.style {  
}  
body {  
background-color: powderblue;  
}  
body {  
display: block;  
margin: 8px;  
}



DIPLOMADO

# Front End

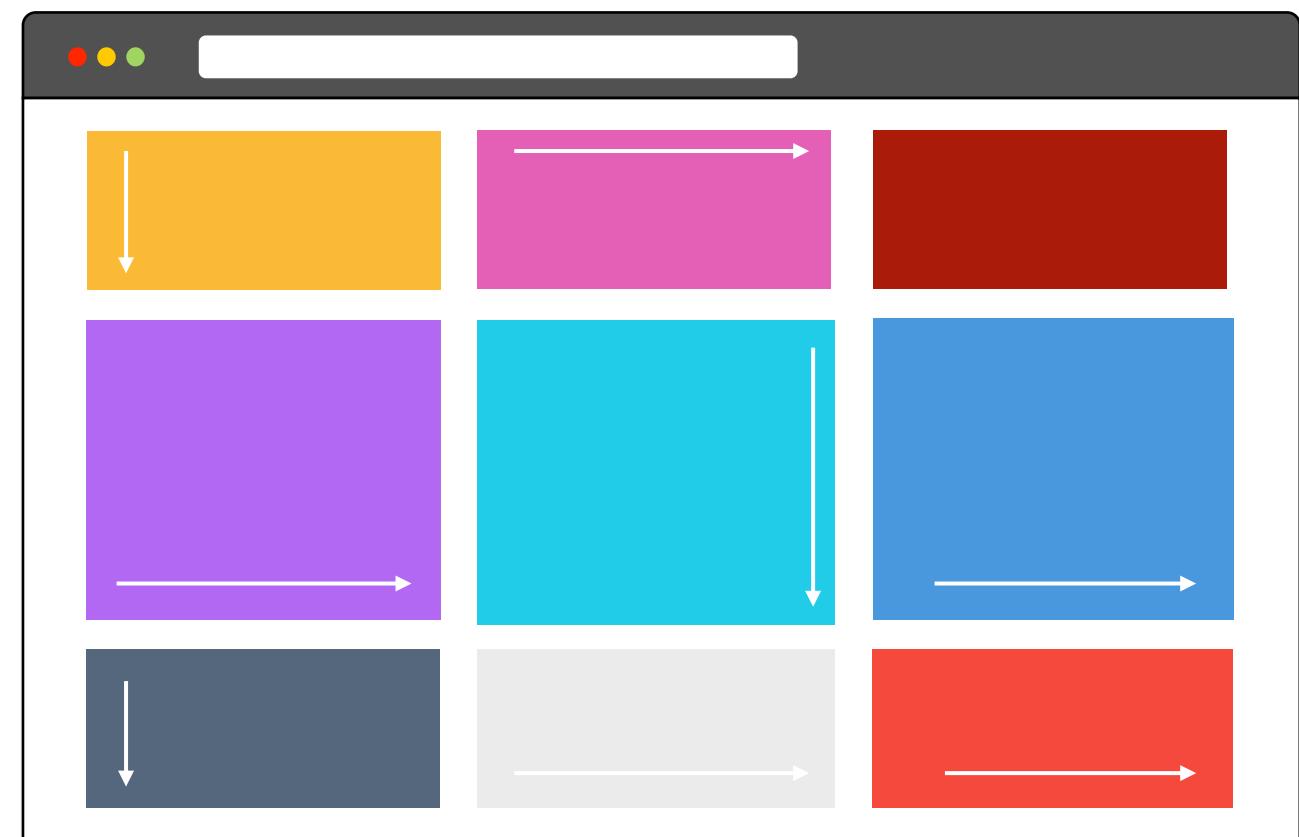
UVM

## TEMA 3

Introducción  
a FlexBox

# Conceptos

- El módulo de diseño Flexbox proporciona una forma eficiente de diseñar, alinear y distribuir el espacio entre los elementos de un contenedor.
- Da al contenedor la capacidad de alterar el ancho / alto (y el orden) de sus elementos para llenar mejor el espacio.



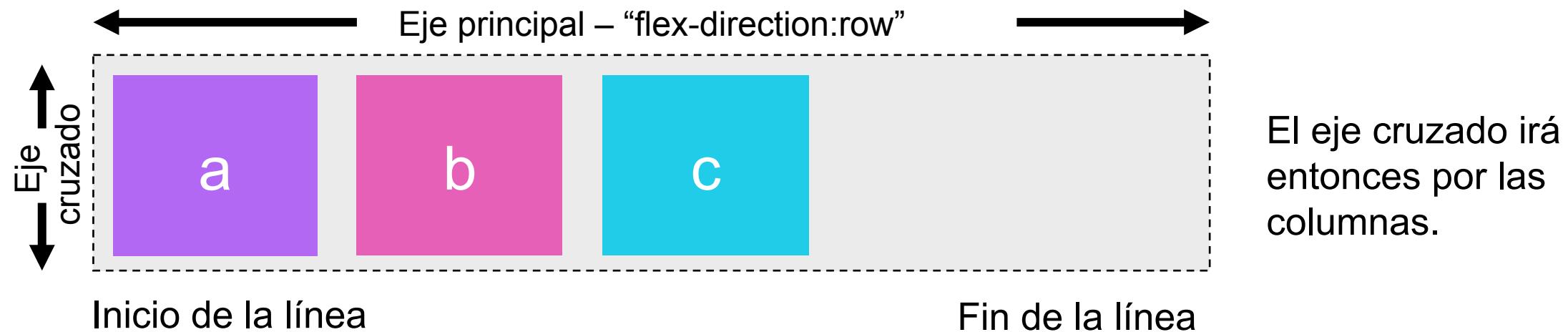


# Conceptos

- Flexbox es independiente de la dirección de la pantalla.
- El diseño de Flexbox es más apropiado para los componentes de una aplicación y diseños de pequeña escala.
- Flexbox es considerado como unidimensional ya que maneja el diseño o "layout" en una sola dimensión a la vez, ya sea como fila o como columna.

# Conceptos

- Todo se maneja en términos de dos ejes: el eje principal y el eje cruzado.
- El Eje Principal se define por la propiedad “flex-direction” y el Eje cruzado es perpendicular a este.
- flex-direction, cuenta con cuatro posibles valores: row, row-reverse, column, column-reverse
- Si elegimos row o row-reverse, el Eje principal correrá a lo largo de la fila en dirección de la línea



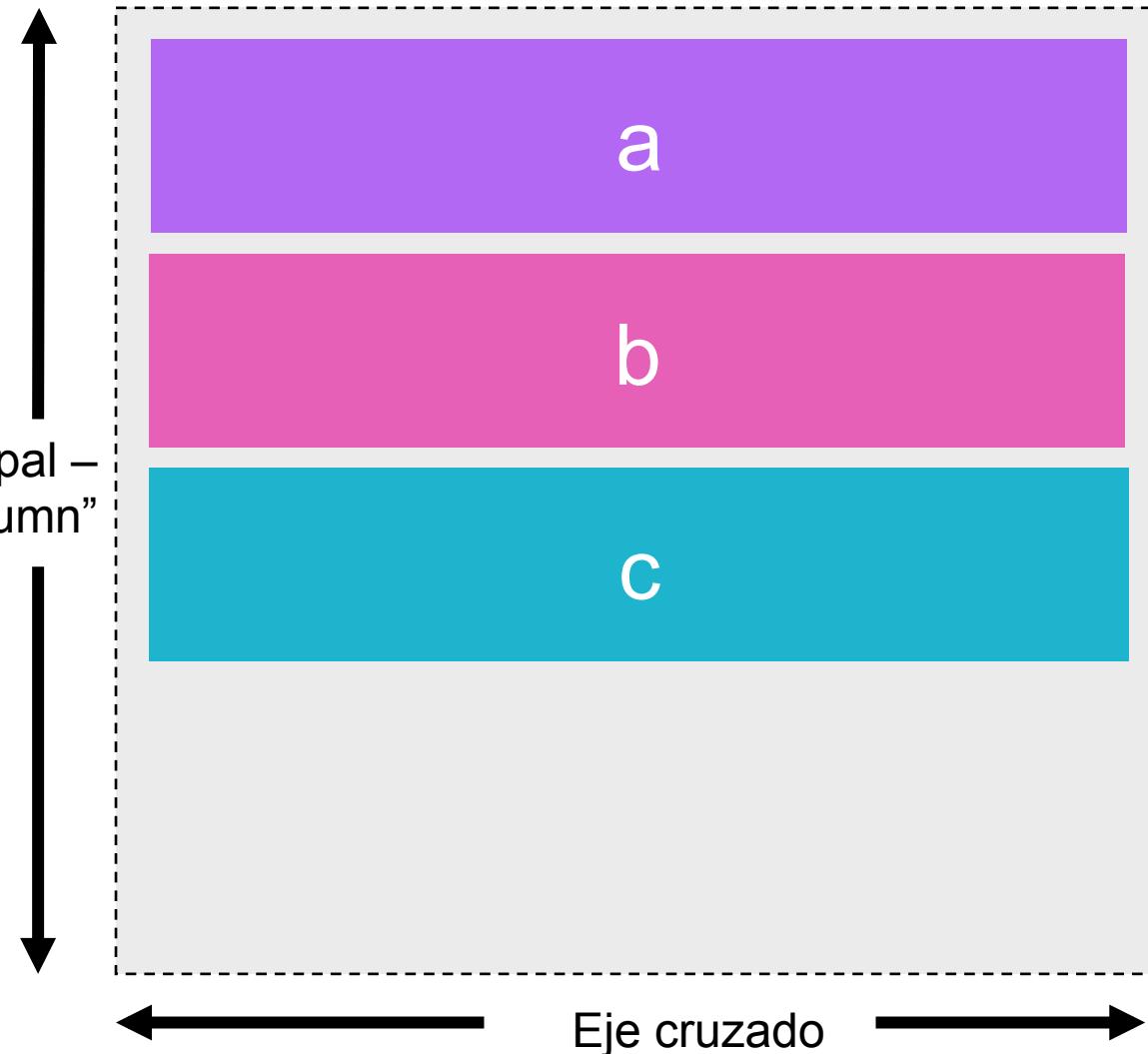
Debido a que trabajamos en español, el margen inicial del eje principal quedará a la izquierda, y el margen final a la derecha.

# Conceptos

- Al elegir column o column-reverse, el eje principal correrá desde el borde superior de la página hasta el final – según la dirección del bloque.

Eje principal –  
“flex-direction:column”

- El eje cruzado corre a lo largo de las filas.



# Conceptos

- Un área del documento que contiene un flexbox es llamada **contenedor flex**.
- Para crear un contenedor flex, establecemos la propiedad del área del contenedor **display como flex o inline-flex**.
- Tan pronto como hacemos esto, **los hijos directos de este contenedor se vuelven ítems flex**.
- Como con todas las propiedades de CSS, se definen algunos valores iniciales, así que cuando se crea un contenedor flex, todos los ítems flex contenidos se comportarán de la siguiente manera:
  - Los ítems se despliegan sobre una fila (la propiedad **flex-direction** por defecto es row).
  - Los ítems **empiezan desde el margen inicial sobre el eje principal**.
  - Los ítems **no se ajustan en la dimensión principal, pero se pueden contraer**.
  - Los ítems **se ajustarán para llenar el tamaño del eje cruzado**.
  - La propiedad **flex-basis** es definida como auto.
  - La propiedad **flex-wrap** es definida como nowrap.

# Conceptos Básicos sobre Estilos

## misestilos.css

```
1. .box {  
2.     display: flex;  
3. }  
4.
```

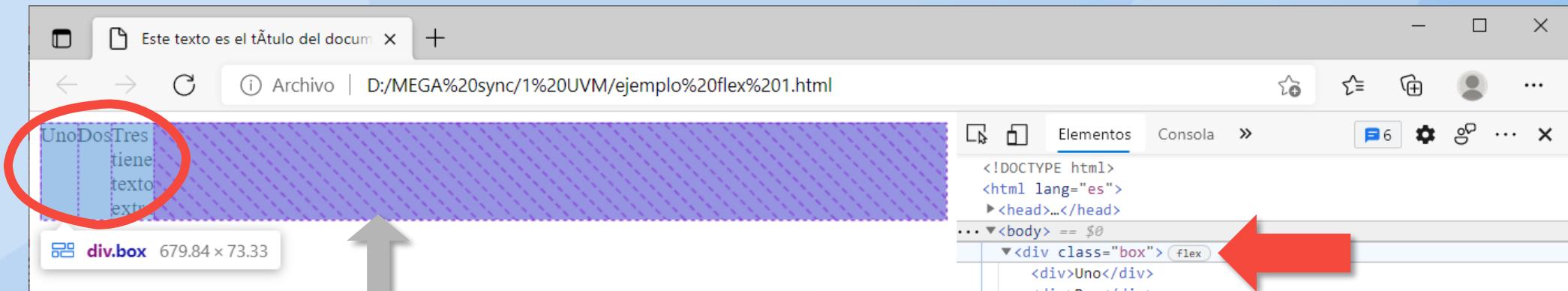
# CSS

## Ejemplo.html

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3. <head>  
4. <title>Este texto es el título del documento</title>  
5. <link href="misestilos.css" rel="stylesheet">  
6. </head> <body>  
7. <div class="box">  
8.     <div>Uno</div>  
9.     <div>Dos</div>  
10.    <div>Tres  
11.        <br>tiene  
12.        <br>texto  
13.        <br>extra  
14.    </div> </div>  
15. </body>
```

Realiza la prueba escribiendo ambos archivos (.css y .html). Guárdalos en el mismo directorio

# Resultado:



El resultado es que todos los ítems se alinearán en una sola fila, usando el tamaño del contenedor como su tamaño en el eje principal.

Si hay más ítems de los que caben en el contenedor, estos no pasarán más abajo si no que sobrepasarán el margen.

Si hay ítems más altos que otros, todos los ítems serán ajustados en el eje cruzado para alcanzar al mayor.

This part of the screenshot shows the Microsoft Edge developer tools. The element inspector on the left shows a tree of HTML elements. The styles panel on the right shows the 'Estilos' tab with a list of CSS rules, including 'element.style { }' and 'body { }'. A red arrow points from the browser's element inspector to the 'Estilos' tab here.

- La propiedad `flex-direction` en el contenedor `flex` nos permite cambiar la dirección de cómo los ítems son desplegados.
- Colocando `flex-direction: row-reverse` se mantendrá el despliegue a lo largo de la fila, sin embargo el inicio y final quedarán al revés
- Si cambiamos `flex-direction` a `column` el eje principal se cambiará y los ítems aparecerán en una columna.
- Colocando `column-reverse` las líneas de inicio y fin serán nuevamente puestas al revés.

# Conceptos

```
1. .box {  
2.     display: flex;  
3.     flex-direction: row-reverse;  
4. }
```

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3. <head>  
4.   <title>Este texto es el título del documento</title>  
5.   <link href="misestilos.css" rel="stylesheet">  
6. </head> <body>  
7.   <div class="box">  
8.     <div>uno</div>  
9.     <div>dos</div>  
10.    <div>tres</div>  
11.  </div> </body> </html>
```

# Conceptos

- Aun cuando Flexbox es un modelo unidimensional, es posible lograr que nuestros ítems flex sean repartidos en varias líneas.
- Haciendo esto, se deberá considerar cada línea como un nuevo contenedor flex.
- Cualquier distribución del espacio solo sucederá dentro de esa línea
- Para lograr repartirse en varias líneas

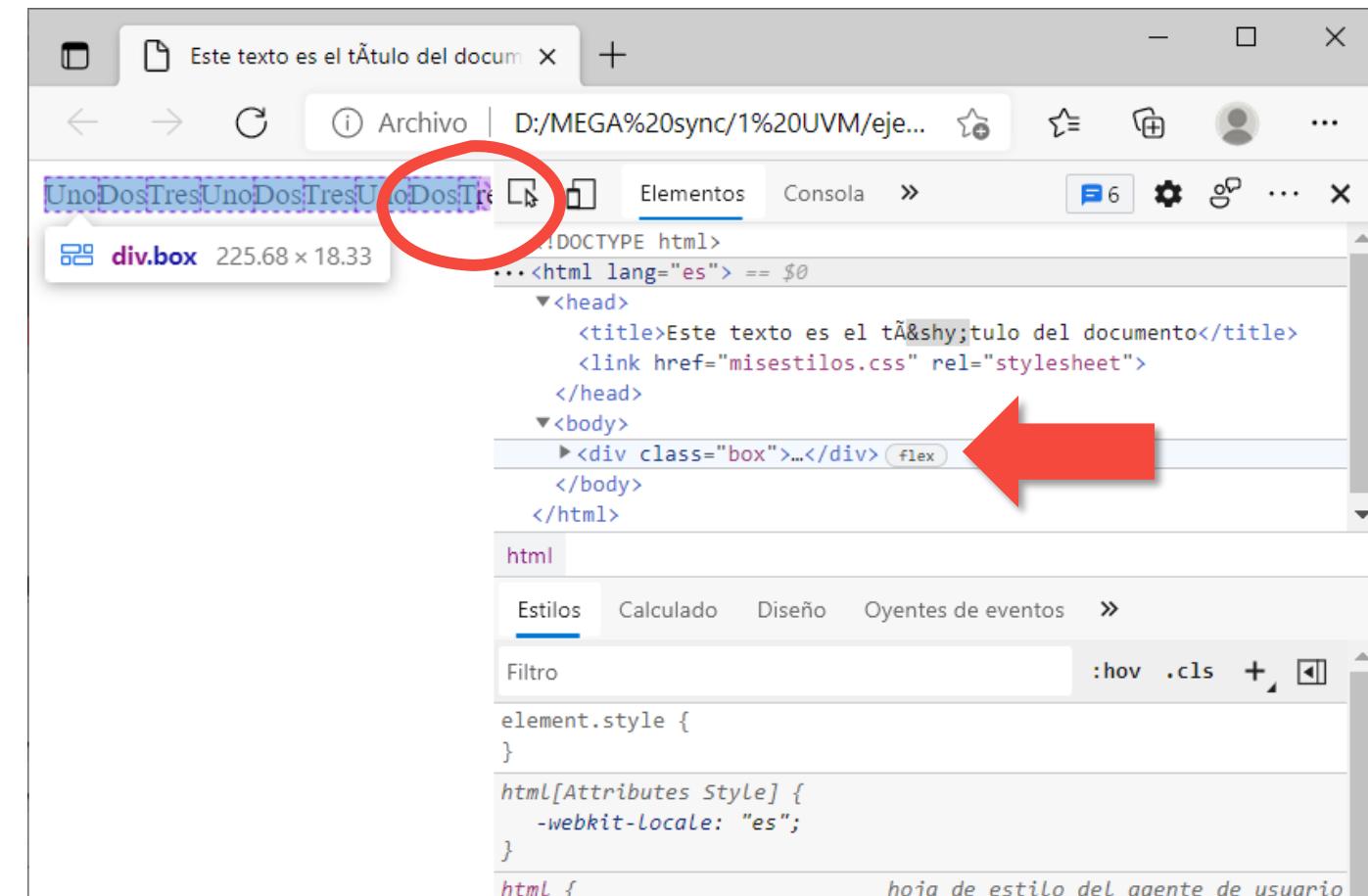
```
1. .box {  
2.     display: flex;  
3.     flex-wrap: wrap;  
4. }
```

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3. <head>  
4. <title>Este texto es el título del documento</title>  
5. <link href="misestilos.css" rel="stylesheet">  
6. </head> <body>  
7. <div class="box">  
8.     <div>uno</div> <div>dos</div> <div>tres</div>  
9.     <div>uno</div> <div>dos</div> <div>tres</div>  
10.    <div>uno</div> <div>dos</div> <div>tres</div>  
11.    <div>uno</div> <div>dos</div> <div>tres</div>  
12. </div> </body> </html>
```

# Conceptos

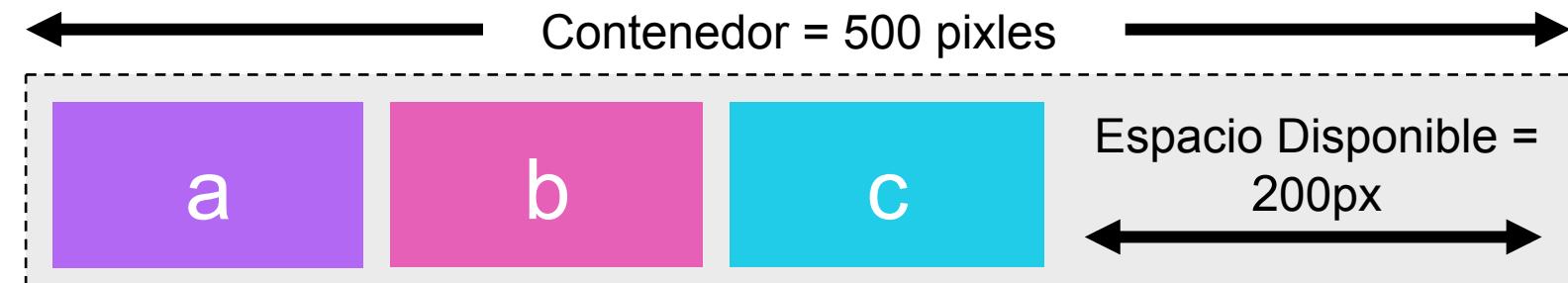
- Se pueden combinar las propiedades flex-direction y flex-wrap en la abreviatura flex-flow. El primer valor especificado es flex-direction y el segundo valor es flex-wrap.
- En el ejemplo intente cambiar el primer valor por uno de los valores permitidos para flex-direction - row, row-reverse, column o column-reverse, y cambie también el segundo valor por wrap y nowrap.

```
1. .box {  
2.     display: flex;  
3.     flex-flow: row wrap;  
4. }
```



# Conceptos

- Para obtener más control sobre los ítems flex podemos apuntarlos directamente por medio de tres propiedades:
  - 1. flex-grow
  - 2. flex-shrink
  - 3. flex-basis
- Si hay tres ítems con ancho de 100 píxeles en un contenedor de 500 píxeles de ancho, entonces quedan 200 píxeles disponibles. Si no cambiamos los valores iniciales entonces flexbox colocará ese espacio después del último ítem.



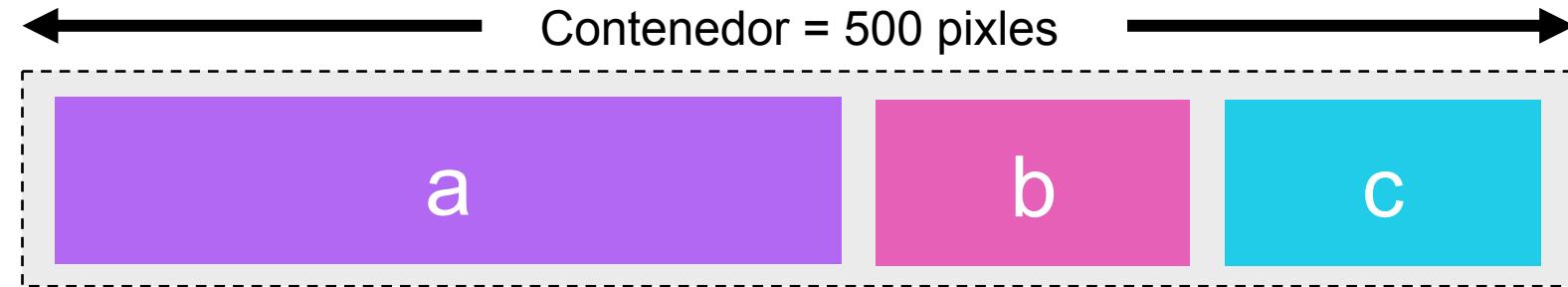
- Si queremos que los objetos crezcan para llenar el espacio, se deben utilizar estas propiedades flex

# Conceptos



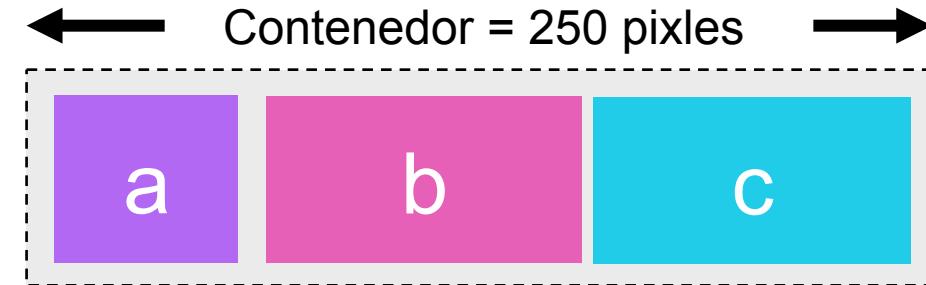
- **flex-basis** define el tamaño de un objeto en términos del espacio que deja como espacio disponible. El valor inicial de esta propiedad es `auto`, en este caso el navegador revisa si los objetos definen un tamaño.
- Si los objetos no tienen un tamaño, entonces el tamaño de su contenido es usado como **flex-basis**.
- **flex-grow** definida como un entero positivo, los objetos flex pueden crecer en el eje principal a partir de **flex-basis**. Esto hará que el objeto se ajuste y tome todo el espacio disponible del eje, o una proporción del espacio disponible si otro ítem también puede crecer.

# Conceptos



- Si le damos a todos los objetos del ejemplo anterior un valor `flex-grow` de 1 entonces el espacio disponible en el contenedor flex será compartido igualitariamente entre estos objetos y se ajustarán para llenar el contenedor sobre el eje principal.
- Podemos usar `flex-grow` apropiadamente para distribuir el espacio en proporciones. Si otorgamos al primer ítem un valor `flex-grow` de 2 y a los otros un valor de 1, entonces 2 partes serán dadas al primer objeto y 1 parte para cada uno de los restantes.

# Conceptos



- Así como la propiedad `flex-grow` se encarga de añadir espacio sobre el eje principal, la propiedad `flex-shrink` controla como se contrae.
- Si no contamos con suficiente espacio en el contenedor para colocar los objetos y `flex-shrink` posee un valor entero positivo, el objeto puede contraerse a partir de `flex-basis`.
- Así como podemos asignar diferentes valores de `flex-grow` con el fin que un ítem se expanda más rápido que otros, un objeto con un valor más alto de `flex-shrink` se contraerá más rápido que sus hermanos que poseen valores menores.

# Ejemplo de Asignación

## misestilos.css

```
1. .box {  
2.     display: flex;  
3. }  
4. .uno {  
5.     flex: 1 1 auto;  
6. }  
7. .dos {  
8.     flex: 1 1 auto;  
9. }  
10. .tres {  
11.     flex: 1 1 auto;  
12. }
```



## Ejemplo.html

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3. <head>  
4. <title>Este texto es el título del documento</title>  
5. <link href="misestilos.css" rel="stylesheet">  
6. </head> <body>  
7. <div class="box">  
8.     <div class="uno">Uno</div>  
9.     <div class="dos">Dos</div>  
10.    <div class="tres">Tres</div>  
11. </div>  
12. </body>  
13. </html>
```

El orden de asignación de la abreviación “flex” es: 1 flex-grow, 2 flex-shrink y 3 flex-basis

# Conceptos

Hay además algunas abreviaturas de valores que cubren la mayoría de los casos de uso. Los valores predefinidos son los siguientes:

- flex: initial
- flex: auto
- flex: none
- flex: número positivo

Con flex: initial el objeto se restablece con los valores iniciales de Flexbox: flex: 0 1 auto.

- En este caso el valor de flex-grow es 0, así que los ítems no crecerán más de su tamaño flex-basis.
- El valor flex-shrink es 1, así que los ítems pueden contraerse si es necesario en vez de salirse de los márgenes.
- El valor de flex-basis es auto. Los ítems pueden definir un tamaño en la dimensión del eje principal, o bien obtener su tamaño por el contenido de los mismos.
- Al usar flex: none se crearán objetos flex totalmente inflexibles. Es como escribir flex: 0 0 auto. Los objetos no pueden ni crecer ni encoger pero serán colocados usando flexbox con flex-basis en auto.

- La propiedad justify-content es usada para alinear los objetos en el eje principal, cuyo flex-direction define la dirección del flujo.
- El valor inicial es flex-start que alineará los objetos al inicio del margen del contenedor, pero también se podría definir como flex-end para alinearlos al final, o center para alinearlos al centro.
- space-between toma el espacio sobrante después de colocar los objetos y distribuye de forma uniforme los objetos para que haya un espacio igual entre cada uno.
- space-around crea un espacio equitativo a la derecha e izquierda de cada objeto.

# Conceptos

```
1. .box {  
2.     display: flex;  
3.     justify-content: flex-start;  
4. }
```

```
1. <!DOCTYPE html>  
2. <html lang="es">  
3.   <head>  
4.     <title>Este texto es el título del documento</title>  
5.     <link href="misestilos.css" rel="stylesheet">  
6.   </head> <body>  
7.     <div class="box">  
8.       <div>uno</div>  
9.       <div>dos</div>  
10.      <div>tres</div>  
11.    </div> </body> </html>
```

DIPLOMADO

Front End



Layouts Responsivos

UVM

TEMA 4

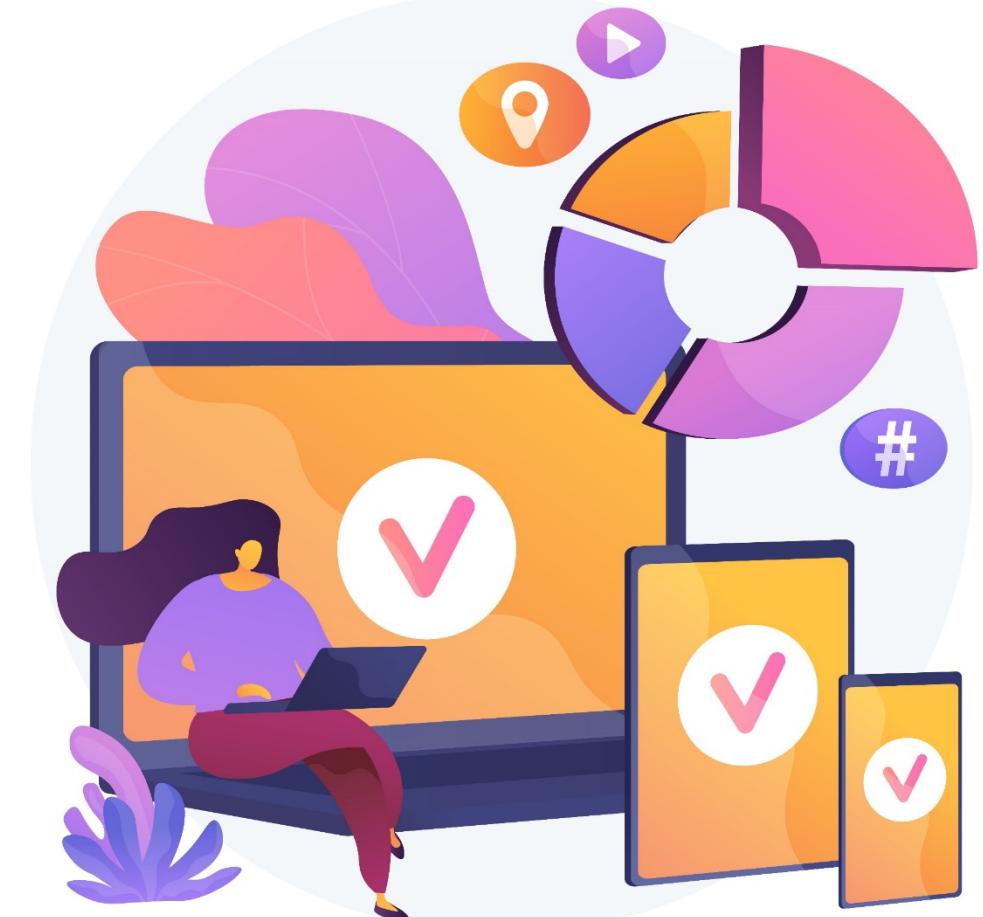
# Conceptos Básicos Diseños Responsivos

El objetivo del diseño web responsive es crear distribuciones que se adapten a la pantalla en la que aparecerán.

Ethan Marcotte, en el artículo que sentó las bases del diseño web responsive\* señala las tres técnicas que deben aplicarse para crear sitios responsivos:

1. Cuadrículas fluidas o responsivas,
2. Imágenes fluidas o responsivas y
3. Media queries.

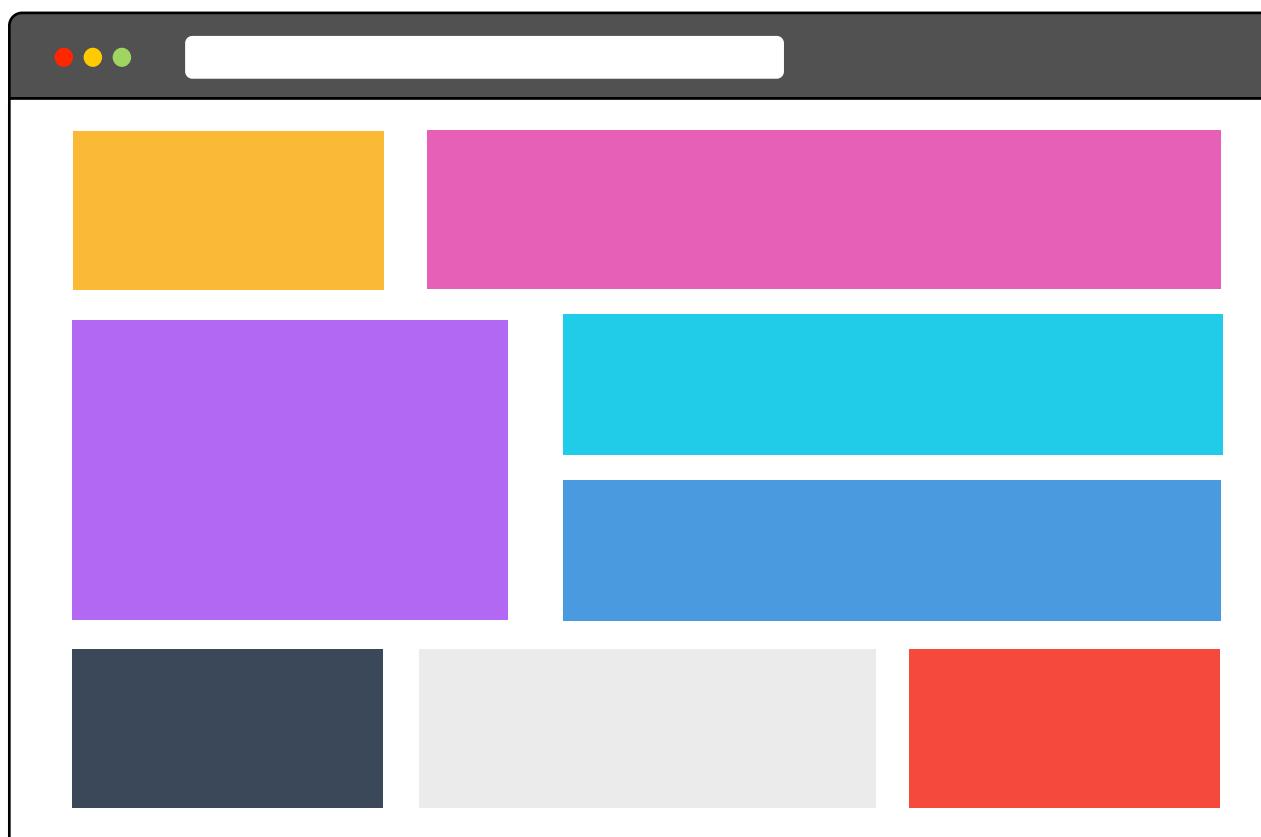
En este apartado nos centraremos en las cuadrículas e responsivas.



\*<http://alistapart.com/article/responsive-web-design>

# Conceptos Básicos Diseños Responsivos

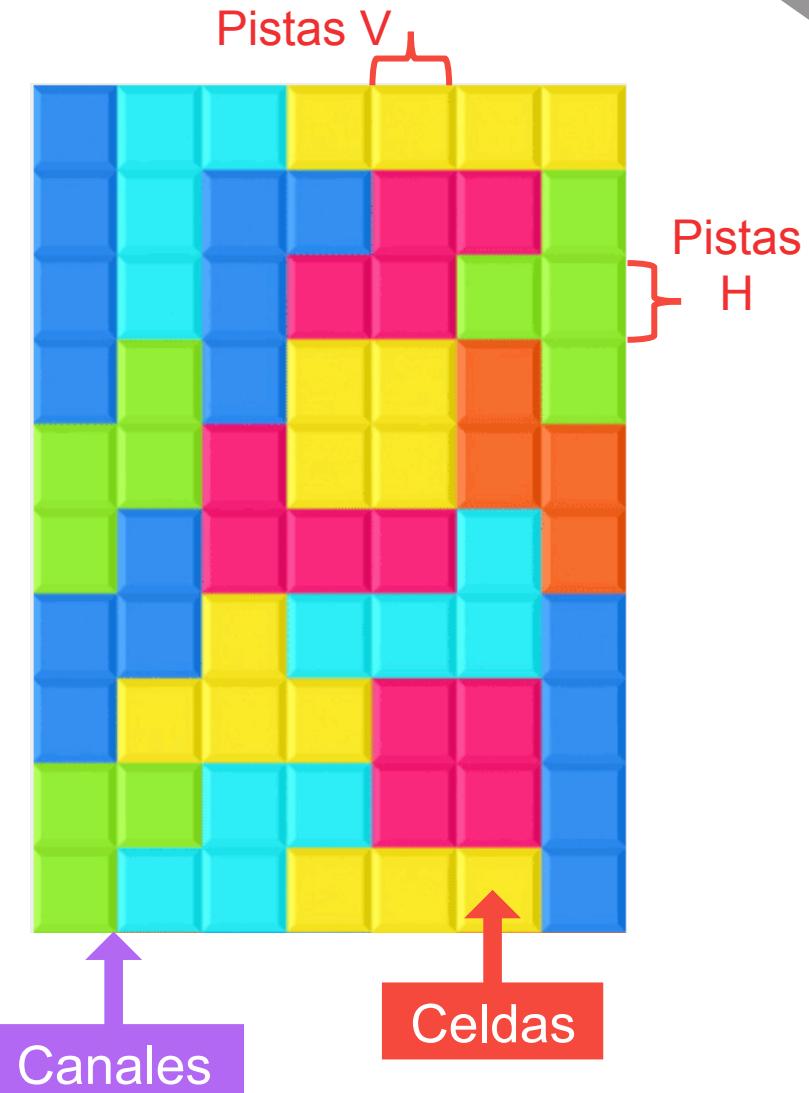
## Cuadrículas responsivas



- Las cuadrículas responsivas crean distribuciones basadas en una cuadrícula tipográfica que se adapta en función del ancho de la pantalla.
- Normalmente tenemos una cuadrícula con 12 columnas para las pantallas más grandes que acaba quedándose en 1 columna para los smartphones.
- CSS Grid Layout ("Grid") aprovecha un soporte sólido de la mayoría de los navegadores.

# Conceptos Básicos Diseños Responsivos

- Grid nos permite organizar elementos en una página, de acuerdo con las regiones creadas por las líneas de guía.
- Estas guías, o líneas de cuadrícula, enmarcan “pistas” de cuadrícula horizontales y verticales.
- Las pistas de cuadrícula sirven como filas y columnas, con canales que corren entre ellas.
- Donde las pistas de cuadrícula horizontal y vertical se cruzan, están las celdas.



# Conceptos Básicos Diseños Responsivos

## Creación de la Cuadrícula CSS

```
1. .grid-1 {  
2.   display: grid;  
3.   grid-template-columns: 150px 150px 150px;  
4.   grid-template-rows: auto auto auto;  
5.   grid-gap: 20px;  
6. }
```



Cada valor de columnas y renglones podría tener auto, tomando altura del contenido, pero establecimos los 150px como ejemplo para visualizar lo que sucede.

- Primero se debe declarar al contenedor como grid al asignarlo a la propiedad display.
- Se define la cuadrícula indicando cuánto renglones y columnas tendrá con las propiedades
  - grid-template-columns y
  - grid-template-rows
- Los valores en grid-template-columns indican que las tres columnas deben tener 150 píxeles de ancho y grid-template-rows indican dimensiones automáticas para los renglones.
- Agregamos un ancho de canales divisores de 20px con grid-gap

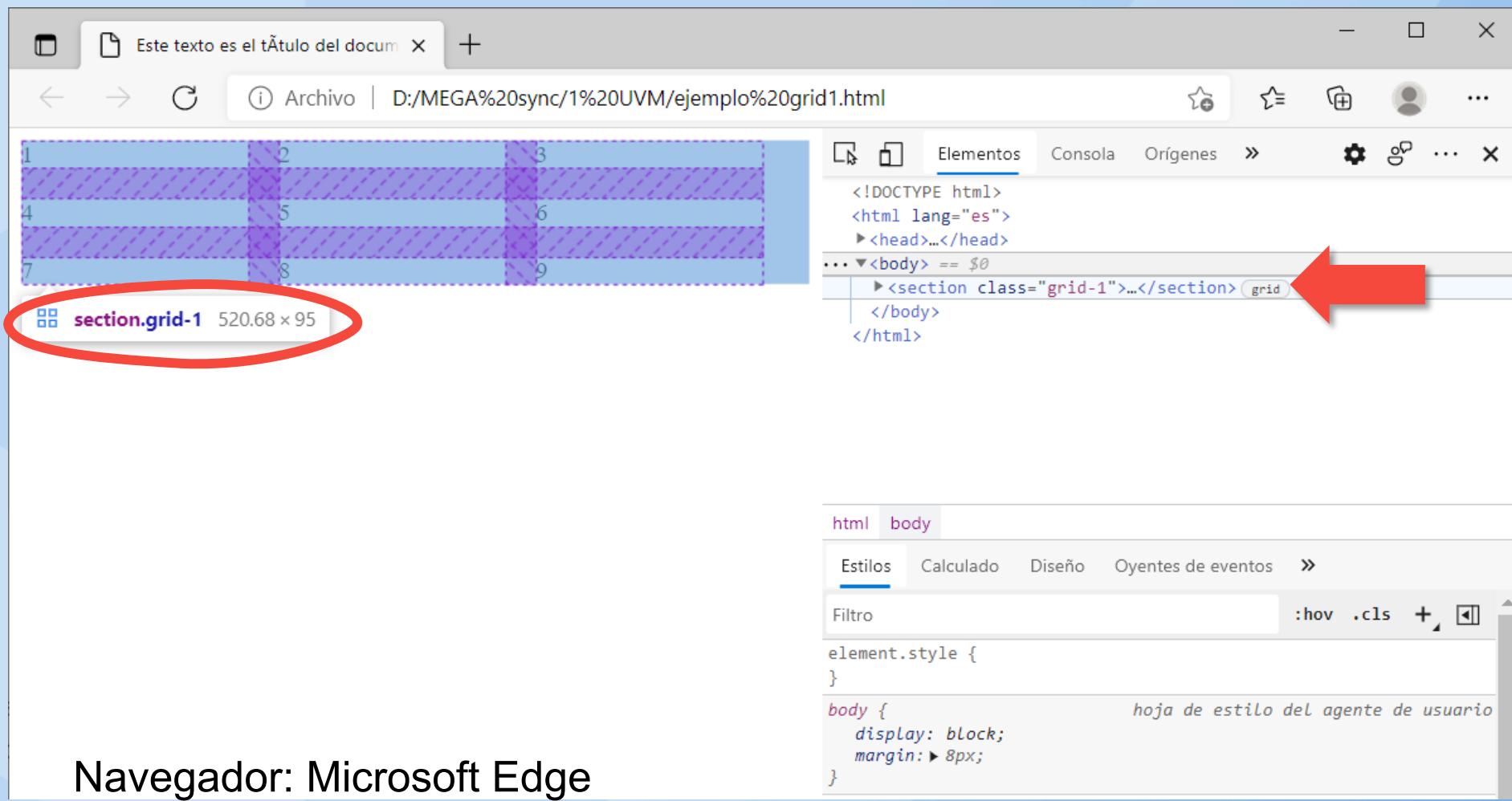
# Conceptos Básicos Diseños Responsivos

## Creación de la Cuadrícula CSS

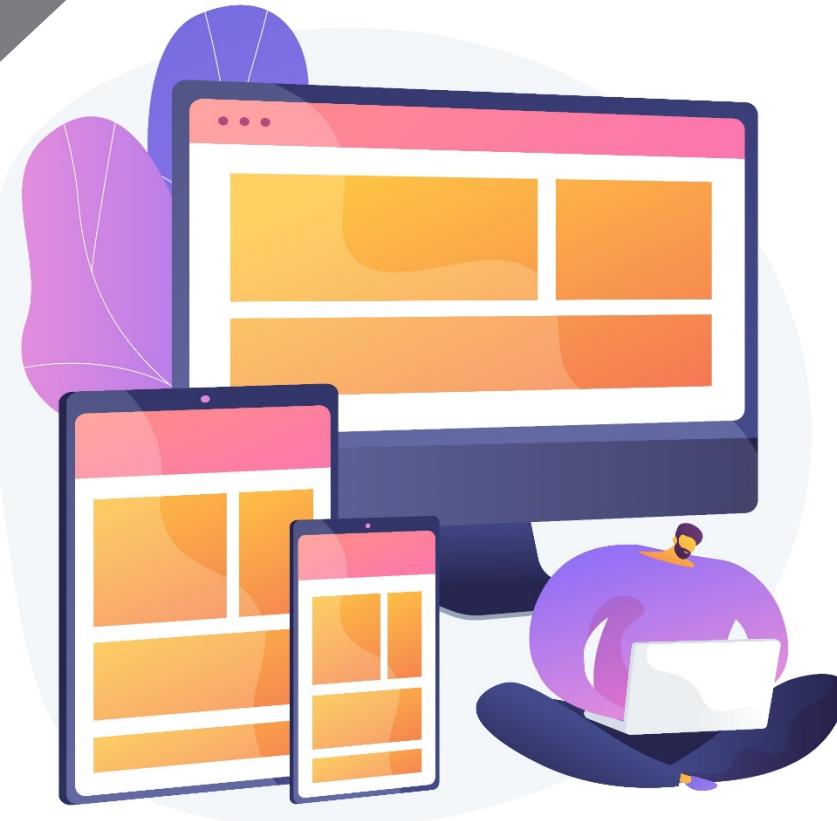
- Para recrear la cuadrícula, necesitamos un elemento contenedor `<section>`
- Posteriormente colocaremos 9 celdas dividiendo el contenido con `<div>` y referenciando a cada uno de ellos con `class`

```
1.  <!DOCTYPE html>
2.  <html lang="es">
3.  <head><title>Este texto es el título del documento</title>
4.  <link href="misestilos.css" rel="stylesheet">
5.  </head> <body><section class="grid-1">
6.      <div class="item-1">1</div>
7.      <div class="item-2">2</div>
8.      <div class="item-3">3</div>
9.      <div class="item-4">4</div>
10.     <div class="item-5">5</div>
11.     <div class="item-6">6</div>
12.     <div class="item-7">7</div>
13.     <div class="item-8">8</div>
14.     <div class="item-9">9</div>
15. </section> </body> </html>
```

# Resultado:



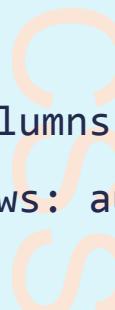
# Conceptos Básicos Diseños Responsivos



- El objetivo de grid es permitirnos controlar adecuadamente el diseño en la web, así que se debe hacer que la cuadrícula que acabamos de definir sea fluida
- Para lograrlo, es necesario cambiar las unidades de píxeles por .grid-1 {

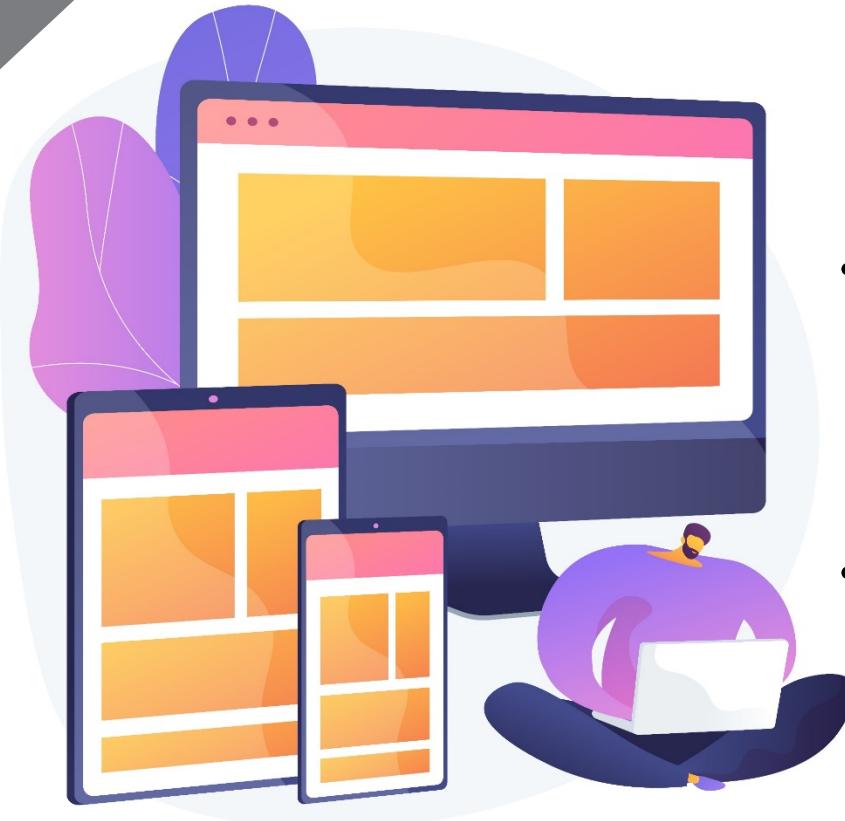
```
1. display: grid;
2. grid-template-columns: 30% 30% 30%;
3. grid-template-rows: auto auto auto;
4. grid-gap: 10%;
```

6. }



- Podríamos eliminar el renglón 4 con grid-template-rows ya que los valores "auto" se asignan por defecto.

# Conceptos Básicos Diseños Responsivos



- Es posible hacer una mejora importante a nuestra cuadrícula y resolver el problema de ancho que acabamos de mencionar con fr o unidad de fracción. Esta se puede combinar con las unidades y % anteriores.
- Una sola unidad fr describe una pieza de las muchas piezas en las que se dividen las columnas.
  1. `grid-template-columns: 1fr 1fr 1fr ;`
- En el ejemplo hay un total de tres unidades fr, por lo que cada columna tendría un tercio de ancho.
  1. `grid-template-columns: 2fr 1fr 1fr;`
- Ahora hay un total de cuatro unidades fr, por lo que la primera columna ocuparía la mitad del ancho disponible, y las otras dos columnas tomarían un cuarto cada una.

# Conceptos Básicos Diseños Responsivos

- Es posible agregar un estilo de color a cada elemento del grid agregando un color a cada elemento en nuestro archivo CSS:

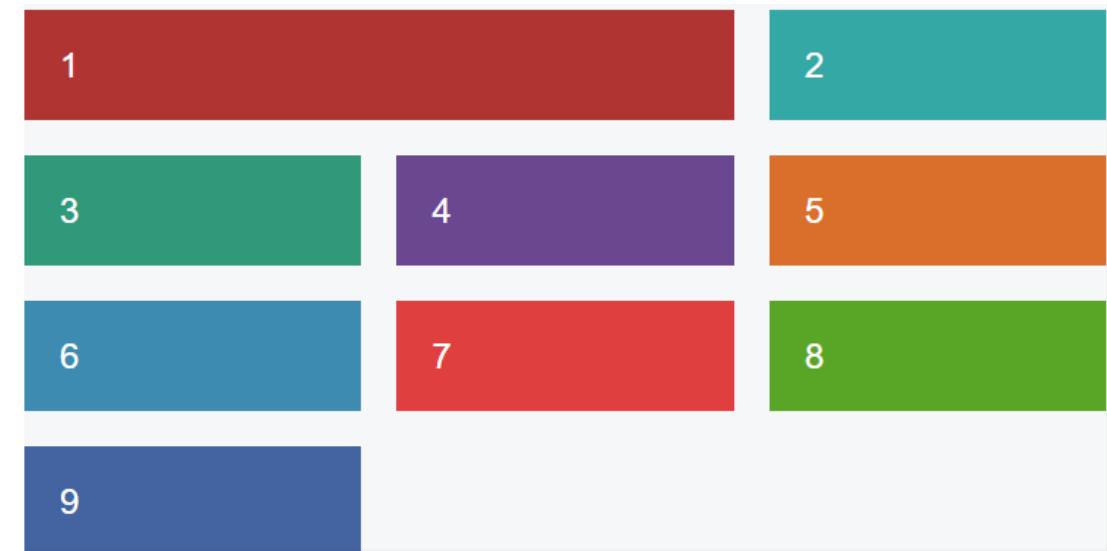
```
1. .item-1 {  
2.   background: #b03532;  
3. }
```



# Conceptos Básicos Diseños Responsivos

- Hasta ahora, cada uno de nuestros elementos ha estado contenido dentro de una sola celda, pero podemos lograr diseños más útiles al romper esos límites con las reglas de estilo grid-column y grid-row que indican a los elementos en donde deben iniciar y terminar:

```
1. .item-1 {  
2.   background: #b03532;  
3.   grid-column: 1 / 3;  
4.   grid-row: 1;  
5. }
```



- En esta declaración abreviada estamos diciendo al elemento que comience en la línea 1 columna 1 de la cuadrícula y termine en la línea 1, columna 3.

# Conceptos Básicos Diseños Responsivos

- Lo mismo se puede hacer con las filas, lo que nos daría un área mucho más grande en la parte superior izquierda de nuestra cuadrícula:

```
1. .item-1 {  
2.   background: #b03532;  
3.   grid-column: 1 / 3;  
4.   grid-row: 1 / 3;  
5. }
```



- Al ir haciendo este tipo de asignaciones, logramos un grid que puede atender a las necesidades del contenido.