



Universidad
Nacional
de Quilmes

HTML

CSS

BOOTSTRAP

CONSTRUCCIÓN DE INTERFACES DE USUARIO



HTML



- ▶ **H**yper **T**ext **M**arkup **L**anguage
 - ▷ Lenguaje de Marcado de HiperTexto
 - ▷ Hipertexto: texto con enlaces a otros textos
- ▶ Propuesto por Tim Berners-Lee en 1980 para intercambiar documentos científicos con referencias.
- ▶ Describe la estructura de un documento mediante *marcas* (etiquetas)
- ▶ Permite agregar información extra a un documento
- ▶ Otros Lenguajes de Marcado:
 - ▷ XML, YAML, TeX (LaTeX), ODF, RTF

Breve repaso de siglas



- ▶ **URL** → *Uniform Resource Locator*
 - ▷ Localizador Uniforme de Recursos
- ▶ **HTTP** → *HyperText Transfer Protocol*
 - ▷ Protocolo de Transferencia de HiperTexto
- ▶ **HTML** → *HyperText Markup Language*
 - ▷ Lenguaje de Marcado de HiperTexto
- ▶ **XML** → *Extensible Markup Language*
 - ▷ Lenguaje de Marcado Extensible

HTML



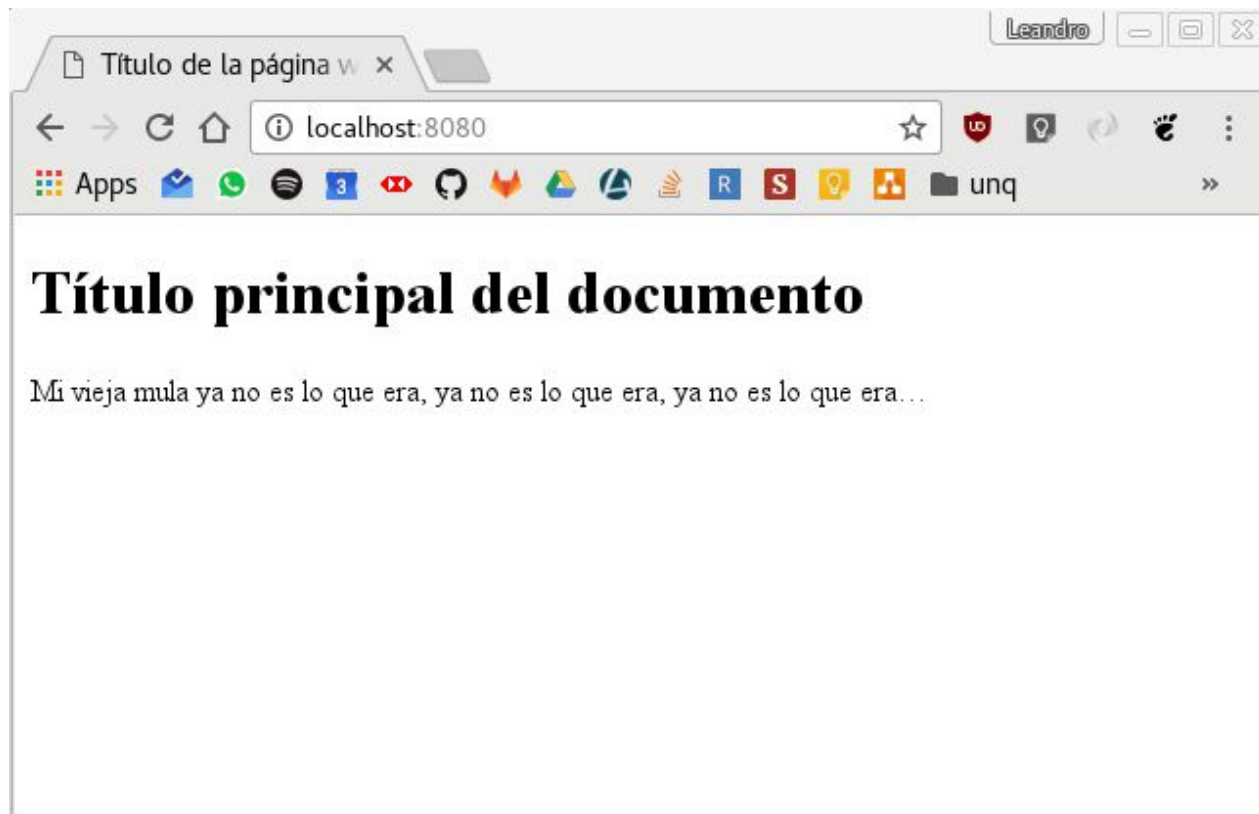
- ▶ Cada HTML contiene **elementos**
 - ▷ *Header, Body, Title, Paragraph, ...*
- ▶ Se representan como **etiquetas**
 - ▷ `<header>`, `<body>`, `<h1>`, `<p>`, ...
- ▶ Las cuales agrupan **contenido**
 - ▷ `<h1>Soy un Título</h1>`
 - ▷ `<p>Soy un Párrafo</p>`
- ▶ Los navegadores procesan las etiquetas y muestra el texto enriquecido

HTML >> Código



```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Título de la página web</title>
  </head>
  <body>
    <h1>Título principal del documento</h1>
    <p>Mi vieja mula ya no es lo que era,
    ya no es lo que era, ya no es lo que era...</p>
  </body>
</html>
```

HTML >> Renderizado



HTML >> Estructura



```
<html>
```

```
<head>
```

```
<title>Título de la Página</title>
```

```
</head>
```

```
<body>
```

```
<h1>Título principal del Documento</h1>
```

```
<p>Mi vieja mula ya no es lo que era, ya no es lo que era, ya no es  
lo que era...</p>
```

```
</body>
```

```
</html>
```

*Solo el
contenido
del elemento
<body> es lo
que se
muestra
en un
navegador.*

HTML >> Tags Básicos



- ▶ **<!DOCTYPE>** Tipo de Documento
- ▶ **<html>** Contenido HTML
- ▶ **<head>** Información del documento
- ▶ **<title>** Título de la página
- ▶ **<body>** Cuerpo principal
- ▶ **<h1>..<h6>** Títulos
- ▶ **<p>** Párrafo
- ▶ **
** Salto de línea
- ▶ **<hr>** Separación temática
- ▶ **<!-- Comentario -->**

HTML >> Más Elementos



- | | | |
|-------------------------------|-------------------------------|-----------------|
| ▶ <code><div></code> | ▶ <code></code> | unordered list |
| ▶ <code></code> | ▶ <code></code> | ordered list |
| ▶ <code></code> | ▶ <code></code> | list item |
| ▶ <code></code> | ▶ <code><table></code> | |
| ▶ <code><code></code> | ▶ <code><th></code> | table header |
| ▶ <code></code> | ▶ <code><tr></code> | table row |
| ▶ <code><audio></code> | ▶ <code><td></code> | table cell |
| ▶ <code><video></code> | ▶ <code><style></code> | |
| ▶ <code><a></code> | ▶ <code><script></code> | |
| ▶ <code><nav></code> | | navegación |
| ▶ <code><link></code> | | linkear recurso |

Referencia exhaustiva: https://www.w3schools.com/tags/ref_byfunc.asp

Elementos *Block* & *Inline*



Cada elemento HTML tiene una visualización por defecto, dependiendo del tipo de elemento que sea. Para la mayoría de los elementos la visualización default es *block* o *inline*:

- ▶ **Block:** Comienzan siempre en una nueva línea y ocupan todo el ancho disponible.
- ▶ **Inline:** Comienzan donde dejó el anterior y sólo ocupan el espacio necesario según su contenido.

Elementos *Block* & *Inline*



```
<!DOCTYPE html>
<html>
<body>
<div>Los div son</div>
<div>elementos block</div>
<div>
  En cambio los
  <span>span</span>
  son elementos
  <span>inline</span>
</div>
</body>
</html>
```

HTML >> Atributos



- ▶ Cada elemento HTML puede contener atributos
- ▶ Los atributos proveen información adicional
- ▶ Se definen en el *tag* de apertura
- ▶ Usualmente son del tipo **clave="valor"**

Referencia exhaustiva: https://www.w3schools.com/tags/ref_attributes.asp

HTML >> Atributos



```
<html lang="es-AR">
<a href="https://www.w3schools.com">This is a link</a>

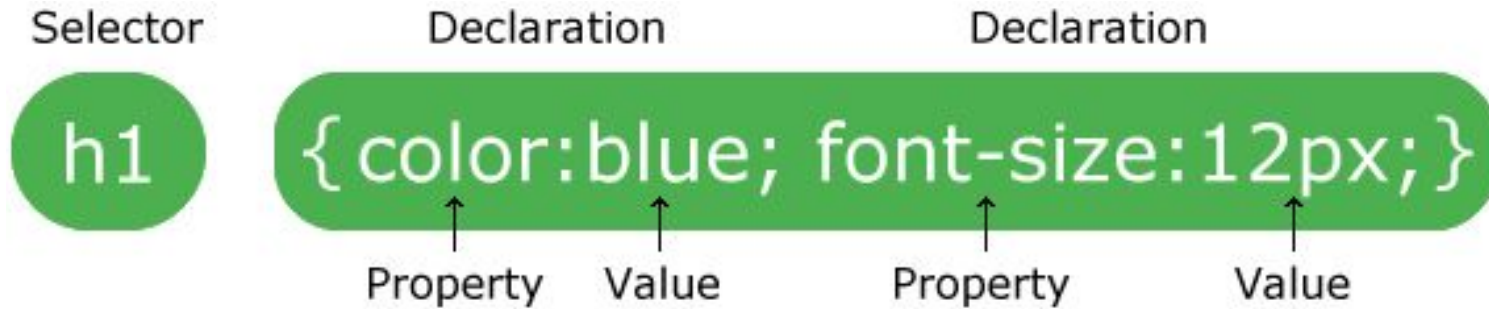
<p style="color:red"
  id="p1"
  class="mi-parrafo"
  title="I'm a tooltip">I am a paragraph</p>
<div data-custom="foo">
  Hola <span data-bar="baz">Jon</span>!
</div>
<button type="submit" name="enviar">Enviar</button>
```

CSS



- ▶ **C**ascading **S**tyle **S**heets
 - ▷ Hojas de estilo en cascada
- ▶ Definen el diseño de los elementos de HTML
 - ▷ En pantalla
 - ▷ En papel (al mandar a imprimir)
- ▶ Permiten separar el diseño del contenido
- ▶ Se pueden combinar múltiples hojas de estilo

CSS >> Sintaxis



/ Este es un comentario
multilínea */*

/ Este es un comentario de línea única */*

CSS >> Selectores



- ▶ Tag `div {}`
- ▶ Cualquier tag con ID `#miId {}`
- ▶ Cualquier tag con Clase `.miClase {}`
- ▶ Tag con ID `p#miId {}`
- ▶ Tag con Clase `p.miParrafo {}`
- ▶ Múltiples Selectores `h1,div,p.clase,a#id {}`

HTML con CSS



```
<!DOCTYPE html>
<html>
<head>
  <!-- Archivo externo (preferible siempre) -->
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  <!-- Definido dentro HTML -->
  <style>
    body { background-color: #FAFAFA; }
    h1 { color: darkBlue; margin-left: 40px; }
  </style>
</head>
<body>
  <!-- Definido dentro del tag (¡evitar!) -->
  <p style="color: red; font-size: 18px">Texto a destacar</p>
</body>
</html>
```

Colores



Hay varias maneras de definir colores:

- ▶ Por nombre (https://www.w3schools.com/colors/colors_names.asp)
 - ▷ Tomato
 - ▷ DodgerBlue
- ▶ En Hexa
 - ▷ #FF6347
 - ▷ #1E90FF
- ▶ RGB (Red, Green, Blue)
 - ▷ rgb(255, 99, 71)
 - ▷ rgb(30, 144, 255)
- ▶ HLS (Hue, Saturation, Lightness / Matiz, Saturación, Luminosidad)
 - ▷ hsl(9, 100%, 64%)
 - ▷ hsl(210, 100%, 56%)

CSS >> Box Model

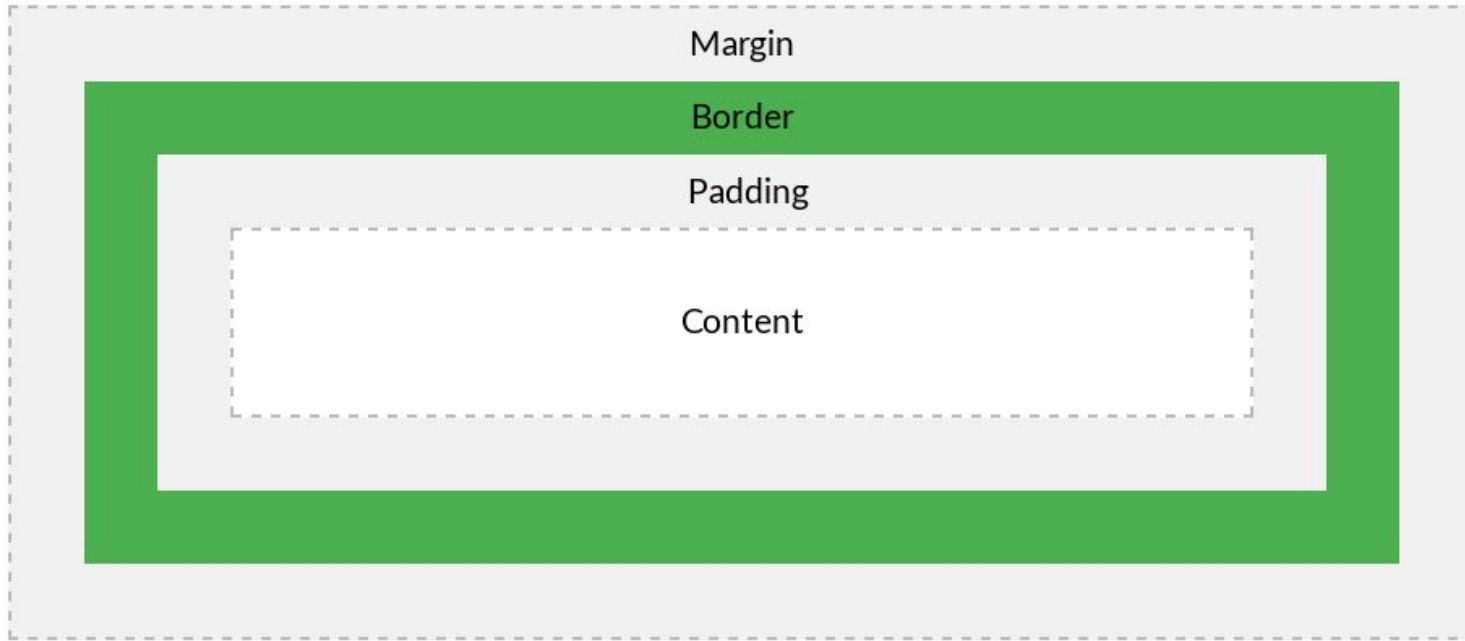


Cada elemento HTML puede ser considerado un **Box** (una “caja”). En CSS el término "box model" se usa cuando se habla del diseño de los elementos.

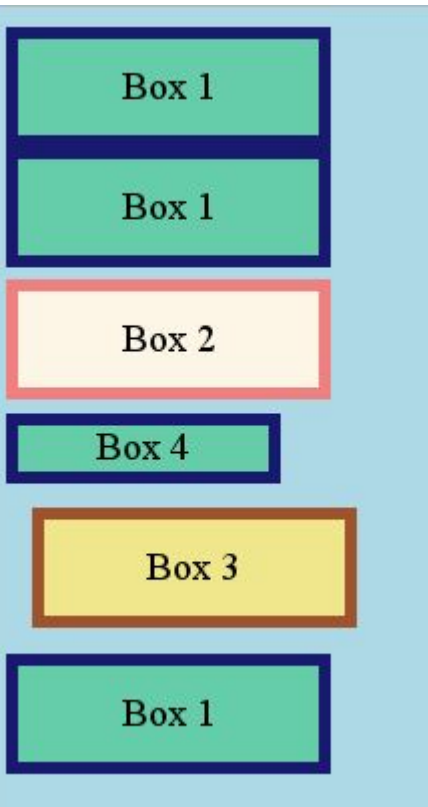
Un **CSS Box Model** es esencialmente una caja que envuelve al elemento HTML y debe definir:

- ▶ Margin (márgen)
- ▶ Border (borde)
- ▶ Padding (relleno)
- ▶ Content (contenido)

CSS >> Box Model



CSS >> Box Model



```
<div id="box1">
```

```
Box 1
```

```
</div>
```

```
<div id="box1">
```

```
Box 1
```

```
</div>
```

```
<div id="box2">
```

```
Box 2
```

```
</div>
```

```
<div id="box4">
```

```
Box 4
```

```
</div>
```

```
<div id="box3">
```

```
Box 3
```

```
</div>
```

```
<div id="box1">
```

```
Box 1
```

```
</div>
```

```
div {text-align:center;
```

```
width: 100px;
```

```
border: 5px solid;}
```

```
div#box1 {padding: 10px;
```

```
border-color: MidnightBlue;
```

```
background-color: MediumAquaMarine;}
```

```
div#box2 {padding: 10px;
```

```
margin-top: 5px;margin-bottom: 5px;
```

```
border-color: LightCoral;
```

```
background-color: OldLace;}
```

```
div#box3 {padding: 10px;
```

```
margin: 10px;
```

```
border-color:Sienna;
```

```
background-color: Khaki;}
```

```
div#box4 {padding: 0; margin: 0;
```

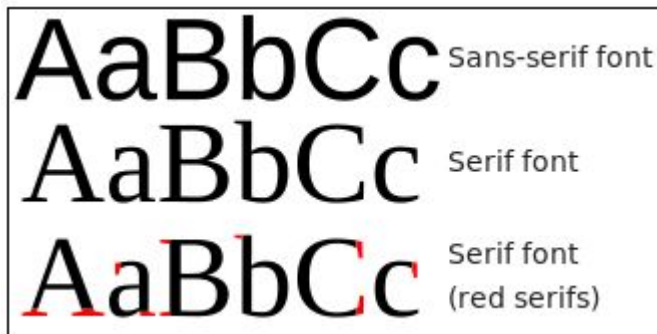
```
border-color: MidnightBlue;
```

```
background-color: MediumAquaMarine;}
```

CSS >> Fonts



- ▶ Familias
 - ▷ Serif
 - ▷ Sans-Serif
 - ▷ Monospace
 - ▷ Fantasy
- ▶ Específicas
 - ▷ Times New Roman
 - ▷ Arial
 - ▷ Courier New
 - ▷ Indie Flower



CSS >> Font Family



- ▶ Para que se vea la fuente elegida **tiene que estar instalada en el SO del usuario**
- ▶ En caso de que no exista debemos proveer alternativas “similares”
- ▶ Para poder hacer eso definimos:
 - a. Fuente concreta
 - b. Familia de fuente concreta
 - c. Fuente genérica

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

CSS >> Pseudo-Clases



Las *pseudo-clases* se usan para definir un estado especial de un elemento. Por ejemplo:

- ▶ Cuando el mouse está sobre el elemento
- ▶ Cuando un link fue visitado
- ▶ Cuando un elemento está en foco

```
/* Sintaxis */ selector:pseudo-class { property: value; }
```

```
a:link      { color: #FF0000; } /* sin clicar */
a:visited   { color: #00FF00; } /* clicado */
a:hover     { color: #FF00FF; } /* mouse encima */
a:active    { color: #0000FF; } /* clicando */
p:first-child { color: blue; } /* todos los primeros p */
p em:first-child { color: blue; } /* el primer em de los p */
p:first-child em { color: blue; } /* todos los em de los primeros p */
```


CSS >> Pseudo-Elementos



Los *pseudo-elementos* se usan para aplicar estilos a partes específicas de un elemento. Por ejemplo:

- ▶ Mayúscula a la primer letra
- ▶ Insertar contenido antes o después

```
/* Sintaxis */ selector::pseudo-element { property:value; }
```

```
p::first-line { color: #ff0000; font-variant: small-caps; }
```

```
p::first-letter { color: #ff0000; font-size: xx-large; }
```

```
h1::after { content: " ;-)" }
```

Bootstrap al rescate

Bootstrap es una *toolkit* (caja de herramientas) para diseñar aplicaciones web con HTML, CSS, y JS.

Permite prototipar ideas o construir un sitio web de forma rápida, eficiente y con un diseño atractivo.

Provee:

- ▶ Sistema de grillas responsive
- ▶ Componentes pre-construidos (muchos)
- ▶ Plugins javascript (con jQuery)
- ▶ Componentes orientados a web-semántica
- ▶ Estilo agradable a la vista



Bootstrap >> Started

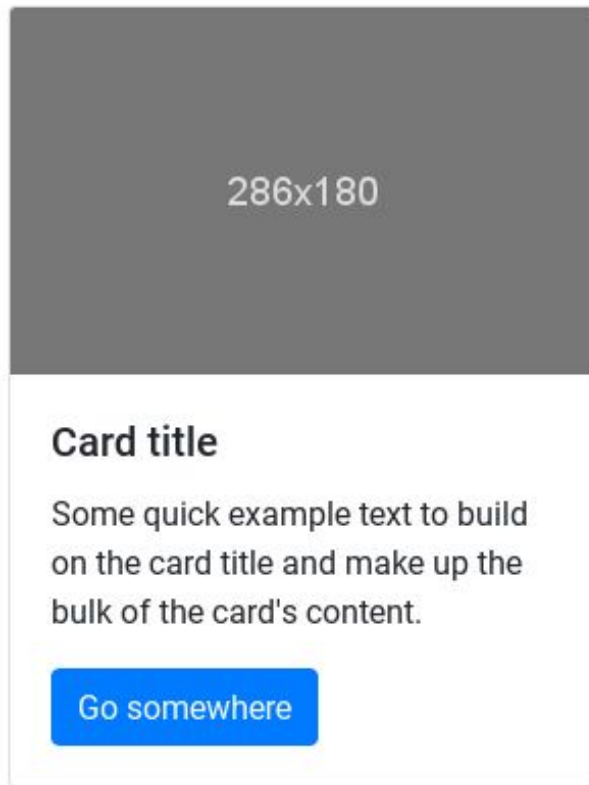


```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap
.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJ
wJ8ERdknLPMO"
crossorigin="anonymous">
    <title>Bootstrap</title>
  </head>
  <body>
    <div class="container">
      <h1>Hello, world!</h1>
    </div>
  </body>
</html>
```

Bootstrap >> Card



```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">
      Some quick example text to build on
      the card title and make up the bulk
      of the card's content.
    </p>
    <a href="#" class="btn btn-primary">
      Go somewhere
    </a>
  </div>
</div>
```



Bootstrap >> Layout & Grid



```
<div class="container">
  <div class="row"><!-- cada row puede tener máximo "12" columnas -->
    <div class="col">1 of 3 (3)</div>
    <div class="col-6">2 of 3 (6)</div>
    <div class="col">3 of 3 (3)</div>
  </div>
  <!-- si se usan menos, se "expanden" hasta cubrir todo el ancho -->
  <div class="row">
    <div class="col">1 of 3 (3.5)</div>
    <div class="col-5">2 of 3 (5)</div>
    <div class="col">3 of 3 (3.5)</div>
  </div>
```

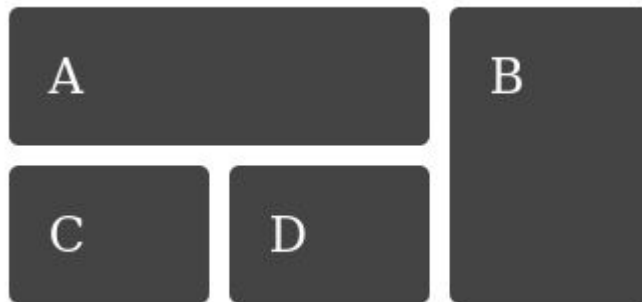
1 of 3 (3)	2 of 3 (6)	3 of 3 (3)
------------	------------	------------

1 of 3 (3.5)	2 of 3 (5)	3 of 3 (3.5)
--------------	------------	--------------

CSS >> Grid Layout

```
body { margin: 40px; }  
.wrapper {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: 100px 100px 100px;  
  background-color: #fff;  
  color: #444;  
}  
.box {  
  background-color: #444;  
  color: #fff;  
  border-radius: 5px;  
  padding: 20px;  
  font-size: 150%;  
}  
.a { grid-column: 1 / 3; grid-row: 1; }  
.b { grid-column: 3; grid-row: 1 / 3; }  
.c { grid-column: 1; grid-row: 2; }  
.d { grid-column: 2; grid-row: 2; }
```

```
<div class="wrapper">  
  <div class="box a">A</div>  
  <div class="box b">B</div>  
  <div class="box c">C</div>  
  <div class="box d">D</div>  
</div>
```



Grid Layouts

- ▶ https://www.w3schools.com/css/css_grid.asp
- ▶ <https://gridbyexample.com/examples/>

Recursos útiles

- ▶ <https://www.w3schools.com>
- ▶ <https://getbootstrap.com/>
- ▶ <https://gridbyexample.com/>
- ▶ dev tools (F12 en el browser)