

# Unidad 1: Introducción

## ¿Qué es Internet?

Internet es una **red de redes** que permite la **interconexión descentralizada** de computadoras a través de un conjunto de protocolos denominado **TCP/IP**.

Red de redes se refiere a que es una red que no sólo interconecta computadoras, sino que interconecta redes de computadoras entre sí.

Permite la interconexión **descentralizada**, conectando computadoras ubicadas en diferentes lugares del mundo, sin la existencia de una computadora central.

Para poder interconectar estas computadoras, de diferentes características, sistemas operativos, etc, fue necesario crear un protocolo para la comunicación denominado TCP/IP.

El origen de Internet se debe gracias a un proyecto militar del ministerio de defensa de los estados Unidos denominado ARPANET (Advanced Research Projects Agency Networks) que tenía como fin crear una red de computadoras que uniera a los centros de investigación de defensa en caso de ataques, de modo de mantener contacto remotamente y que no se interrumpiese su funcionamiento si alguno de sus nodos fuera destruido.

## WWW (World Wide Web)

A diferencia de lo que suele pensarse, Internet y la World Wide Web no son sinónimos. La WWW es un sistema de información global desarrollado en 1989 por Tim Berners

Lee.

Mediante este sistema, un usuario puede acceder a documentos HTML almacenados en diferentes servidores, pudiendo saltar de un servidor a otro de forma instantánea, mediante los enlaces de hipertexto. Un hipertexto es un enlace entre diferentes documentos.

Este servicio permite el acceso a información mediante el protocolo HTTP (Protocolo de transferencia de hipertexto).

La www, es uno de los servicios disponibles en internet, pero no es el único.

Otros servicios y protocolos disponibles en la red de redes son el acceso remoto a computadoras conocido como Telnet, el sistema de transferencia de archivos FTP, el correo electrónico (POP y SMTP), y las conversaciones online o chats, entre otros..

En la web, las páginas se almacenan en computadoras llamadas Servidores Web y se transfieren mediante un protocolo de comunicación llamado HTTP.

Cuando un usuario se conecta a Internet puede acceder a las páginas Web a través de un programa llamado navegador.

Cuando el usuario, al que vamos a denominar “cliente”, ingresa una url en la barra de direcciones del navegador, realiza un pedido por el protocolo HTTP (HTTP request), al servidor web donde se encuentra alojado el sitio.

El servidor web, procesa el pedido y devuelve un archivo en formato HTML, que es el único lenguaje que interpretan los navegadores. Esta respuesta, la hace también vía el protocolo HTTP, en este caso HTTP response.

Los navegadores interpretan el documento HTML que reciben del servidor y lo muestran en pantalla.

## ¿Qué es un servidor?

Un servidor es una aplicación instalada en una computadora remota, capaz de atender las

peticiones de un cliente y devolverle una respuesta en concordancia.

Trabaja con lenguajes de servidor (como PHP, .NET o ruby) que permiten, por ejemplo, procesar y acceder a datos almacenados en una base de datos.

Si bien puede utilizar en el procesamiento, distintos lenguajes, al momento de devolver, devuelve siempre un documento escrito en lenguaje HTML.

## HTML

HTML significa HyperText Markup Language.

Es un lenguaje basado en marcas conocidas como “tags” o etiquetas.

Estas etiquetas marcan la manera en la que se debe entender y mostrar la información.

Por ejemplo, la etiqueta `p`, es una etiqueta que marca que un bloque de texto es un párrafo.

Entonces es `<p>` sería la etiqueta que abre y `</p>` la etiqueta que cierra.

Todo el texto que se encuentre entre un `<p>` y un `</p>` va ser interpretado por los navegadores como un párrafo.

Un archivo html, es un archivo de texto que es reconocido por el navegador, ya que tiene la extensión `.html` o `.htm`. Cualquiera de las dos extensiones son válidas.

Cuando incorpora algún recurso, como una imagen, no la incorpora dentro del mismo archivo, sino que le indica al navegador de qué dirección obtenerla. Y es el navegador que al momento de mostrarla lee la imagen de la ubicación previamente indicada.

Al ser un archivo de texto, puede ser editado por cualquier editor de texto, como por ejemplo el block de notas que viene por defecto instalado en nuestra computadora.

Existen a su vez, programas especializados, para escribir código HTML, que son conocidos como IDEs, ya que no solo son editores de código, sino que a su vez nos brinda una serie de herramientas, que nos ayudan a hacer nuestro trabajo más simple y eficiente.

Entre los IDE más conocidos, podemos mencionar Notepad ++, Sublime, Brackets, Visual

Studio Code, Atom.

Te sugerimos que los pruebes y encuentres el editor con el que te sientas más a gusto.

## Versiones de HTML

El HTML fue desarrollado originalmente por Tim Bernes-Lee pero debido al rápido crecimiento de la web, surgió la necesidad de crear un estándar para que tanto los programadores como los navegadores pudieran basarse en unas mismas normas para escribir HTML. Cada versión de HTML establece unas normas respecto a cuáles son las etiquetas válidas y cómo se deben escribir.

El organismo encargado de establecer estos estándares es el W3C (World Wide Web Consortium).

El HTML 5 es la última especificación oficial, fue liberado por la W3C a finales de octubre de 2014 y se espera que continúe evolucionando a lo largo de los próximos años.

HTML5 , entre las mejoras que incorpora podemos mencionar:

- Etiquetas Semánticas. Nuevas etiquetas de marcado semántico, que permiten organizar más eficientemente el contenido de nuestros archivos.
- Reproductor de Audio y video incorporado. Nuevas etiquetas para integrar directamente en el HTML archivos de video o de audio que antes necesitaban de algún componente externo.
- Acceso a los dispositivos (geo localización)
- Nuevas etiquetas para los formularios y atributos.
- Canvas HTML5. Posibilidad de incorporar dibujos y animaciones.
- Drag and Drop (arrastrar y soltar objetos al navegador)

- Local Storage. Capacidad de Almacenamiento Offline

Entre otras características.

## HTML y CSS

Inicialmente, el formato visual estaba dentro del mismo html.

A partir de 1996 surge el lenguaje CSS que permite dar estilos a los textos en forma independiente del contenido en HTML.

CSS significa, Hojas de Estilo en Cascada (Cascading Style Sheets).

El estándar actual, es la versión CSS3, que incorporó entre otras características:

Animaciones y transiciones

- Transformaciones de elementos (rotación, sesgado, traslación y escala, y transformaciones 3D)
- Fuentes Embebidas(@Font Face) poder utilizar tipografías sin necesidad de que estén instaladas en las computadoras de los usuarios.
- Bordes redondeados
- Degradés
- Sombras (tanto para las fuentes como para los elementos contenedores)
- Transparencias
- Medias Queries.  
Condicionales que permiten mostrar en forma diferente el contenido visual, en función del ancho o del alto de la ventana del navegador.  
Esto se refiere específicamente a poder hacer que nuestro sitio se adapte a los diferentes dispositivos tanto móviles, como tablets o pcs de escritorio y adaptar el

contenido en función del formato y tamaño de pantalla.

## Direcciones IP y nombres de dominio

Cada ordenador que se conecta a Internet se identifica por medio de una dirección IP que se compone de 4 grupos de números comprendidos entre el 0 y el 255, ambos inclusive, y separados por puntos, que permiten identificar una computadora en forma unívoca. Así, por ejemplo un dirección IP podría ser: 125.110.13.45.

Esta distribución jerárquica de la Red, permite enviar y recibir rápidamente paquetes de información entre dos ordenadores conectados a Internet en cualquier parte del Mundo, y desde cualquier sub-red a la que pertenezcan.

Un usuario de Internet, no necesita conocer ninguna dirección IP, ya que éstas son manejadas por los equipos en sus comunicaciones por medio del Protocolo TCP/IP, de manera transparente para el usuario.

Para nombrar los ordenadores dentro de la Red se utilizan los Nombres de Dominio, que son la traducción para "los humanos" de las direcciones IP que usan los ordenadores. Así por ejemplo, yahoo.com, nerja.net son nombres de dominio.

Los nombres de domino son palabras separadas por puntos.

Para ver una página se debe conocer su dirección o nombre de dominio. La dirección de una página o sitio Web recibe el nombre de **URL** (Universal Resource Location – Localización Universal de Recursos) y se compone de la siguiente manera:

Extensiones de dominio según la orientación del sitio:

- **com:** empresas
- **edu:** instituciones de carácter educativo, sobre todo Universidades.
- **org:** organizaciones.
- **gov:** instituciones gubernamentales.
- **mil:** centros militares.

Extensiones de dominio según el país al que pertenece:

- **es** España
- **fr** Francia
- **uk** Reino Unido
- **it** Italia
- **ar** Argentina
- **uy** Uruguay

Los Servidores de Nombres de Dominio (DNS), son equipos de la Red cuya misión es averiguar la dirección IP de un ordenador a partir de su nombre de dominio.

## Qué elementos necesitamos para poner un sitio en internet?

1. Tener registrado un dominio
2. Contratar un servidor de hosting para alojarlo
3. Contar con un programa ftp para poder subir los archivos al servidor.

## Registro de Dominio

La entidad encargada de registrar dominios con extensión .ar es el NIC Argentina.

Para registrar un dominio, tenemos que ingresar a [www.nic.ar](http://www.nic.ar)

Actualmente para registrar y administrar un dominio .com.ar necesitamos contar con número de CUIT o CUIL y clave fiscal.

No es necesario contar previamente con el hosting contratado.

Una vez que queremos habilitar nuestro sitio en internet, tenemos que delegar el dominio, apuntando el nombre del dominio a los DNS del servidor, para que al momento de ingresar la URL en el navegador, este se dirija hacia nuestro sitio.

## Hosting

El hosting es el servicio de alojamiento de un sitio en internet.

Las empresas de hosting, alquilan un espacio en un servidor que es una computadora conectada a internet las 24hs, que tiene instalados una serie de servicios.

Los hostings nos brindan dos servicios básicos, el del alojamiento del sitio en internet y el de correo, es decir la posibilidad de tener casillas de correo con la extensión de nuestro dominio.

## Protocolo FTP

El protocolo FTP (file transfer protocol), es el protocolo de transferencia de archivos, que nos permite tanto subir archivos a un servidor, como descargarlos del servidor a una carpeta local.

Uno de los programas más utilizados para hacer ftp, es el filezilla, que se puede descargar en forma gratuita.

El Dreamweaver, tiene el ftp, como una funcionalidad incorporada y puede sincronizar automáticamente, nuestro sitio local con el sitio online. (Es decir detectar las diferencias y mantener actualizado el servidor, sin que le tengamos que indicar qué archivos hay que actualizar).

## Bibliografía



- “World Wide Web Consortium (W3C) “  
<http://www.w3c.org>
- Brackets  
<http://brackets.io/>
- Notepad++  
<https://notepad-plus-plus.org/>
- Sublime Text  
<https://www.sublimetext.com/>

# Unidad 2: HTML 5

HTML	3
Sintaxis	3
Atributos de etiquetas	3
Etiquetas que no tienen cierre	4
Comentarios en html	4
Estructura Básica de html	5
Etiqueta Title (<title>)	6
Encabezados (etiquetas <h1>, <h2>, <h3>, <h4>, <h5>, <h6>)	6
Párrafos (etiqueta <p>)	7
Saltos de Línea (etiqueta  )	7
Listas Numeradas (etiqueta <ol>)	8
Listas No Numeradas (etiqueta <ul>)	8
Imágenes (etiqueta <img>)	9
Vínculos (etiqueta <a>)	9
Podemos establecer vínculos:	10
Formateo de Texto	11
Bloques (etiqueta <div>)	12
Tablas (etiqueta <table>)	12
HTML5 DOCTYPE	13
Etiquetas Semánticas:	13
Header	14



Nav	<b>15</b>
Main	<b>15</b>
Section	<b>16</b>
Article	<b>16</b>
Aside	<b>16</b>
Footer	<b>17</b>
Resumen Etiquetas HTML Básicas	<b>18</b>
Resumen Etiquetas Semánticas HTML5	<b>20</b>

# HTML

HTML (HyperText Markup Language), es un lenguaje basado en marcas conocidas como etiquetas.

Estas etiquetas marcan la manera en la que se debe entender y mostrar la información.

Por ejemplo, la etiqueta `p`, es una etiqueta que marca que un bloque de texto es un párrafo.

Entonces es `<p>` sería la etiqueta que abre y `</p>` la etiqueta que cierra.

Todo el texto que se encuentre entre un `<p>` y un `</p>` va ser interpretado por los navegadores como un párrafo.

Los archivos html, tienen la extensión `.htm` o `.html`

## Sintaxis

La estructura básica de las etiquetas HTML se caracteriza por ir encerradas entre los símbolos `<>` para las de apertura y `</>` para las de cierre.

Siguiendo este modelo:

```
<nombre_etiqueta> ... </nombre_etiqueta>
```

Por ejemplo:

```
<h1> Titulo </h1>
```

Es etiqueta que abre, contenido de texto a marcar, y etiqueta que cierra.

## Atributos de etiquetas

Las etiquetas html, tienen atributos que varían dependiendo la etiqueta.

Los atributos ,van dentro de la apertura de la etiqueta y siguen el siguiente formato:

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

Es decir nombre\_de\_atributo seguido por el signo igual (=), y el valor va encerrado entre comillas dobles (" ").

Por ejemplo:

```
<h1 id="inicio"> Titulo </h1>
```

## Etiquetas que no tienen cierre

Algunas etiquetas especiales llamadas “etiquetas vacías” no necesitan encerrar ningún texto. La etiqueta `<br>` por ejemplo, se utiliza para indicar el comienzo de una nueva línea. Por sus características, la etiqueta `<br>` nunca encierra ningún contenido de texto.

Por ende no se cierra.

Otra etiqueta que no se cierra es la etiqueta `img`, que sirve para vincular una imagen a nuestro html.

## Comentarios en html

Si quiero dejar comentarios dentro del código que me ayuden a identificar y entender ciertos fragmentos, el formato en html para los comentarios es el siguiente:

```
<!-- Comentarios -->
```

```
<!-- para abrir un comentario
```

```
--> para cerrar un comentario
```

Todo el texto que vaya entre `<!--` y `-->` quedará como comentario, es decir que no saldrá en el html resultante.

## Estructura Básica de html

La estructura básica de una página HTML es la siguiente:

```
<html>  
  
  <head>  
  
  </head>  
  
  <body>  
  
  </body>  
  
</html>
```

Las 3 etiquetas básicas de un documento html son la etiqueta `<html>`, la etiqueta `<head>` y la etiqueta `<body>`

- `<html>`: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>`.

En el interior de la etiqueta `<html>` se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta `<html>` se ignora.

- `<head>`: delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta `<title>`, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte

superior izquierda de la ventana del navegador.

- `<body>`: delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el `<body>` de un documento contiene cientos de etiquetas HTML, mientras que el `<head>` no contiene más que unas pocas.

## Etiqueta Title (`<title>`)

```
<html>
```

```
  <head>
```

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

```
  </head>
```

```
  <body>
```

Esta etiqueta se utiliza para indicar el título de un documento. Los navegadores lo visualizan en la solapa de la página web. Va dentro de la cabecera del documento (dentro de la etiqueta `<head>`)

## Encabezados (etiquetas `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`)

```
<h1> - <h6>
```

Las etiquetas h son utilizadas para marcar los encabezados de un documento HTML.

Son etiquetas jerárquicas que van del h1 al h6, siendo el h1 el encabezado más importante de nuestro documento y el h6 el menos importante.

Solo puede haber un h1 en nuestro documento HTML.

Los encabezados son textos destacados en nuestro documento.

**<h1> Textos de Encabezado 1 </h1>**

**<h2> Textos de Encabezado 2 </h2>**

**<h3> Textos de Encabezado 3 </h3>**

**<h4> Textos de Encabezado 4 </h4>**

**<h5> Textos de Encabezado 5 </h5>**

**<h6> Textos de Encabezado 6 </h6>**

## **Párrafos (etiqueta <p>)**

La etiqueta p, sirve para marcar párrafos comunes.

**<p> Un párrafo</p>**

## **Salto de Línea (etiqueta <br>)**

Para marcar saltos de línea, existe la etiqueta <br> que como vimos anteriormente, es una etiqueta que no tiene cierre.

Podemos establecerla para marcar saltos de línea dentro de un párrafo o dentro de cualquier etiqueta HTML.

Por cada <br> que insertemos obtendremos un salto de línea.

**<p>**

**Renglón 1 <br>**



Renglón 2

</p>

## Listas Numeradas (etiqueta <ol>)

La etiqueta que nos permite generar listas numeradas, es la etiqueta <ol>

- <ol></ol>, para marcar un contenido como lista
- y la etiqueta <li></li>, para marcar cada ítem en la lista

<ol>

<li>Primer Paso</li>

<li>Segundo Paso</li>

</ol>

Esto se vería como:

1. Primer Paso
2. Segundo Paso

## Listas No Numeradas (etiqueta <ul>)

La etiqueta que nos permite generar listas NO numeradas, es la etiqueta <ul>

- <ul></ul>, para marcar un contenido como lista
- y la etiqueta <li></li>, para marcar cada ítem en la lista

<ul>

<li>Blanco</li>

<li>Verde</li>

</ul>

Esto se vería como:

- Primer Paso
- Segundo Paso

## Imágenes (etiqueta <img>)

La etiqueta que nos permite vincular imágenes a nuestro documento html, es la etiqueta <img>. Al igual que la etiqueta br, es una de las etiqueta que NO tienen etiqueta de cierre.

Tiene dos atributos obligatorios:

1. El atributo src. En donde establecemos la ubicación de la imagen (ruta) y el nombre del archivo incluyendo su extensión  
``
2. El atributo alt. En donde escribimos un texto descriptivo acerca de la imagen que acabamos de insertar.

El atributo alt, es un texto alternativo a nuestra imagen.

```

```

Otros atributos de la etiqueta img son: width y height que nos permiten establecer un ancho o un alto para nuestras imágenes. Lo ideal es no usarlos, y darles el formato a través de css.

```

```

## Vínculos (etiqueta <a>)

La etiqueta <a></a> sirve para marcar un hipervínculo.

Su atributo más importante es el atributo href, que es donde se incluye el link de

destino.

Por ejemplo:

```
<a href="https://www.unlam.edu.ar/">Unlam </a>
```

Esto se vería así en un navegador:

Unlam

y al hacer clic en el texto Instituto Da Vinci, nos llevaría a su página (la url que incluimos en el atributo href).

Otro atributo que tiene esta etiqueta, es el atributo target, en donde podemos establecer en donde queremos que se abra el vínculo.

Por ejemplo, si queremos que la página del instituto se abra en página aparte, el valor que deberíamos usar en el atributo target, es "\_blank"

```
<a href="https://www.unlam.edu.ar/" target="_blank">
```

Unlam

```
</a>
```

### **Podemos establecer vínculos:**

1. A otro documento u página externa

Por ejemplo

```
<a href="http://www.google.com.ar"> Google </a>
```

2. Dentro del mismo documento

Si queremos generar un vínculo que nos lleve a una parte de nuestro html.

Por ejemplo queremos incluir al final de nuestro documento, un link que nos lleve al principio.

Para hacer esto necesitamos dos pasos:

1. Paso 1, agregar un id, a la etiqueta adonde queremos ir

```
<h1 id="inicio"> </h1>
```

2. Paso 2, agregar un vínculo cuyo valor del atributo href sea numeral (#)  
nombre\_de\_id

```
<a href="#inicio"> Ir al Inicio</a>
```

3. De tipo Email

mailto: el modificador mailto, dentro del href de una etiqueta a, genera un vínculo, que al hacer clic intenta abrir nuestro programa de correo y generar un correo nuevo hacia la casilla indicada.

```
<a href="mailto:ejemplo@ejemplo.com">ejemplo@ejemplo.com</a>
```

## Formateo de Texto

- **<strong></strong>**

Sirve para marcar un texto importante.

Por defecto muestra la tipografía en bold, pero se le puede dar cualquier formato en el css.

- **<em></em>**

Sirve para enfatizar un texto.

Por defecto muestra la tipografía en cursiva, pero se le puede dar cualquier formato en el css.

- **<small></small>**

Define un texto de tamaño chico.

## Bloques (etiqueta <div>)

Esta etiqueta sirve para marcar bloques o secciones dentro de un HTML.

Los divs son usados para darle formato en css, a un bloque de contenido.

Se les suele asignar un atributo id o un atributo class.

Por ejemplo:

```
<div class="noticia">  
  
    <h2>Titulo Noticia</h2>  
  
    <p>Párrafo descriptivo de la noticia.</p>  
  
</div>
```

## Tablas (etiqueta <table>)

Cuando necesitamos utilizar una tabla en nuestro documento, usamos la etiqueta <table></table>, un <tr> para cada fila y un <td> para cada columna.

```
<table>  
  
    <tr> Fila  
  
        <td> Columna </td>  
  
    </tr>  
  
</table>
```

# HTML5 DOCTYPE

La manera de indicar que un document html es html5, es el siguiente:

```
<!DOCTYPE html>

<html lang="es">

  <head>

    <meta charset="UTF-8">

    <title></title>

  </head>

  <body>

  </body>

</html>
```

La primera línea indica el doctype (esto indica que se trata de la versión HTML 5).

El atributo lang en la etiqueta de apertura <html> es el único atributo que hay que especificar en HTML5. Este atributo define el idioma del contenido del documento que estamos creando, en este caso es por español.

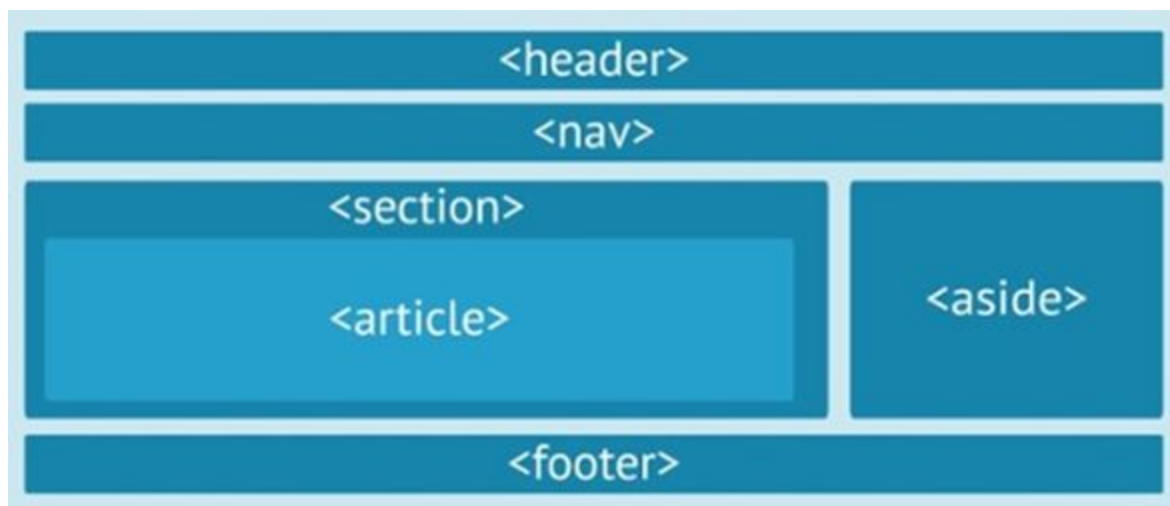
Adicionalmente, se puede agregar la etiqueta meta charset que define los tipos de caracteres que va a tener el documento. La opción UTF-8 nos permite utilizar acentos y ñ, y que estos se visualicen correctamente en el navegador.

## Etiquetas Semánticas:

Las etiquetas semánticas ayudan a definir la estructura del documento y permiten que las páginas web sean mejor indexadas por los buscadores.

Una etiqueta se califica como semántica si tiene que ver el significado, es decir, si nos informa sobre lo que trata su contenido.

El esquema de una página con etiquetas semánticas podría ser el siguiente:



Las etiquetas que vamos a ver a continuación, son las que definen la estructura general del sitio (HEADER, ARTICLE, SECTION, ASIDE, FOOTER, NAV), y vienen a reemplazar a los div con id encabezado, contenido y pie, entre otros.

## Header

La etiqueta header, marca el encabezado del contenido de nuestro documento.

`<header>` no debe ser confundido con la etiqueta `<head>` de html.

La etiqueta header va dentro de la etiqueta body y suele contener, el título principal de la página, el logo del sitio, y una breve descripción corta.

`<body>`

```
<header>
```

```
    <h1>Título de la página</h1>
```

```
</header>
```

```
</body>
```

## Nav

Esta etiqueta marca una barra de navegación con enlaces internos al sitio. Pueden haber una o varias etiquetas nav en un sitio web.

```
<nav>
```

```
    <ul>
```

```
        <li>Home</li>
```

```
        <li>Contacto</li>
```

```
    </ul>
```

```
</nav>
```

## Main

Se utiliza para marcar el contenido central de una página (Sin encabezado, menu, ni pie). Solo puede haber una etiqueta main por página.

```
<main>
```

```
    <section>
```

```
        ....
```

```
    </section>
```



</main>

## Section

Se utiliza para marcar secciones de contenido que están relacionados entre sí (un mismo tema). Pueden haber varios section por página.

```
<section>

    <article>

        ....

    </article>

</section>
```

## Article

Cada contenido independiente, completo, atómico. Puede haber uno, o varios por página. Son subsecciones dentro de un section.

```
<article>

    <h2>Titulo de la Noticia</h2>

    <p>Contenido.....</p>

</article>
```

## Aside

Sirve para indicar que se trata de información complementaria, relacionada con la página, pero que no pertenece al contenido principal del mismo.

Ejemplo: productos relacionados, comentarios de lectores

`<aside>`

`<h2>201x</h2>`

...

`</aside>`

## Footer

Sirve para marcar el pie de página. Por lo general, van los datos de contacto, redes sociales, copyright de la página.

`<footer>`

`<p>`

201x – Programación.....

`</p>`

`</footer>`

## Resumen Etiquetas HTML Básicas

Etiqueta	Descripción
html	Indica el comienzo y el final de un documento HTML
head	Delimita la parte de la cabecera del documento.
body	Delimita el cuerpo del documento HTML, solo las etiquetas contenidas dentro del body se visualizarán en el navegador.
title	Establece el título de un documento html. Va dentro de la etiqueta <head>
h1-h6	Las etiquetas h, marcan los encabezados de un documento.
p	Sirve para marcar párrafos
br	Establece un salto de línea. Esta etiqueta no tiene etiqueta de cierre.
ol	Listas Numeradas.
ul	Listas NO Numeradas.
li	Marca un ítem en una lista

img	Vincula una imagen a un documento HTML. Los dos atributos obligatorios son src (ubicación) y alt (texto alternativo)
a	Sirve para generar un vínculo. Su atributo más importante es href (se establece el link de destino)
div	Etiqueta que sirve para marcar bloques de contenido.

## Resumen Etiquetas Semánticas HTML5

Etiqueta	Descripción
header	La etiqueta header va dentro de la etiqueta body y suele contener, el título principal de la página, el logo del sitio, y una breve descripción corta.
nav	Este elemento sirve para marcar que el contenido del mismo es un menú y que contiene enlaces a secciones internas al sitio.
main	Se utiliza para marcar el contenido central de una página (Sin encabezado, menu, ni pie). Solo puede haber una etiqueta main por página.
section	Se utiliza para marcar secciones de contenido que están relacionados entre sí (un mismo tema). Pueden haber varios section por página.
article	Este elemento representa una porción independiente de información relevante (por ejemplo, cada noticia de un diario o cada entrada de un blog).
aside	Este elemento representa contenido que está relacionado con el contenido principal pero no es parte del mismo. Ejemplos pueden ser citas, información en barras laterales, publicidad, artículos relacionados, etc
footer	Esta etiqueta se utiliza para marcar el pie de página.



# HTML



# HTML 5

## Formularios

Programación Web I / Móvil I

**Comisión Miércoles Noche**

DIS. Alicia Rosenthal

Tec. Willian Dos Reis

**Comisión Viernes Mañana**

Ing. Gabriel Panik

# Etiqueta FORM

Los formularios constituyen uno de los elementos más importantes porque permiten el intercambio de información. HTML5 incorpora nuevos atributos y elementos.

**<form></form>**



- **action:** Define la acción que se realiza cuando se envía un formulario. Especifica una dirección URL dentro del servidor donde enviar el formulario.
- **method:** define de que manera se van a transmitir los datos hacia el archivo que se define en el action. Existen dos métodos :
  - get** ( por defecto) cuando la información a enviar no es sensible; los datos del formulario son visible en la dirección url de la página. Recomendado para poca cantidad de texto;
  - post**, cuando la información es sensible; ofrece mayor seguridad dado que los datos no están visibles en la dirección de la página.
- **enctype:** Especifica cómo los datos serán codificados el enviarlos al servidor. Sólo se puede aplicar si el método utilizado es post. Las opciones son : application/x-www-form-urlencoded ( por defecto), multipart/form-data y text/plain.

# Etiqueta FORM

## ATRIBUTOS OBLIGATORIOS

```
<form action="php/enviar.php"  
method="get" enctype="multipart/form-data">
```

...CONTENIDO DEL FORMULARIO...

```
</form>
```

# Etiqueta FORM

## NUEVOS ATRIBUTOS

### **autocomplete:**

Muestra las entradas previas como posibles valores. Puede aplicarse al form o a alguna de sus elementos en particular. Sus valores posibles son on (por defecto) y off.

### **novalidate:**

los formularios en HTML5 presentan capacidad propia de validación. El atributo novalidate es un atributo booleano, cuando está presente indica que los datos enviados no deben ser validados

Ejemplo combinando todos los atributos:

```
<form action="php/enviar.php" method="get"  
enctype="multipart/form-data" autocomplete="off" novalidate>  
    ...CONTENIDO DEL FORMULARIO...  
</form>
```

# Campos de un formulario

Los formularios pueden contener los siguientes tipos de campos:

- **Input**

Campo de una sola línea, puede ser de tipo text, number, date, color, radio, submit, reset, password, hidden, checkbox, file, email

- **Textarea**

Campo multilínea.

- **Select**

Combo desplegable que puede contener varias opciones(options)

- **Button**

Botón de envío de formulario

- **Datalist**

Sirve para asociar a un campo de tipo input una lista predefinida de opciones. Es uno de los campos agregados en HTML5

# Campos de un formulario

The diagram illustrates a form with the following fields and their associated labels:

- Surname\***: Short Text
- Date of birth**: Date
- Mobile / Home Phone Number**: Phone Number
- Membership Number**: Number
- Participant type\***: Drop down
- Do you have a medical condition?\***: Radio
- Conditions**: Multiple Checkbox
- Please provide details**: Long Text
- Declaration\***: Checkbox
- Info**: Description text only

The form fields are as follows:

- Surname\***: A single-line text input field.
- Date of birth**: A date picker input field.
- Mobile / Home Phone Number**: A single-line text input field.
- Membership Number**: A single-line text input field with the instruction "Please complete if known" below it.
- Participant type\***: A dropdown menu with the placeholder text "Please select a Participant type".
- Do you have a medical condition?\***: A radio button group with "Yes" and "No" options.
- Conditions**: A section titled "Please tick any that are appropriate" with four checkboxes: "Asthma", "Diabetes", "Heart Condition", and "High Blood Pressure".
- Please provide details**: A large, empty text area for providing details.
- Declaration\***: A checkbox followed by the text "I have read and accepted the [booking conditions](#)".
- Info**: A text block stating "If you have any other questions, please call us on 01234 56789".

# Campos de un formulario

**Enter Contact Information**

placeholder attribute

required attribute

input type email

input type tel

input type url

input type date

input type number

pattern attribute

input type range

input type color

input type file

Search:

input type search

Name:

E-Mail:

Phone:

Website:

Birthdate:

Quantity Attending:

Pattern:

Range:

Hair color:

Select photo images:  Add files

Submit

# Campos de un formulario

Los campos de un formulario tienen varios atributos, dos de cuales son obligatorios:

- **Atributo id:** permite identificar unívocamente un campo de un formulario y sirve entre otras cosas para su validación
- **Atributo name:** atributo obligatorio para que el valor del campo se envíe al enviar un formulario. Los campos sin el atributo name no son enviados.

Adicionalmente a estos atributos, es conveniente asociar a cada campo a una etiqueta label. La etiqueta label debe tener un atributo for con igual valor que el id del campo para que ambos queden asociados:

```
<label for="nombre">Nombre: </label>  
<input type="text" name="nombre" id="nombre">
```

# Campos de un formulario

Etiquetas(Ejemplo)	Descripción
<code>&lt;textarea name="nombre" cols=30 rows=4&gt;&lt;/textarea&gt;</code>	Crea una caja de texto, las columnas son el ancho y las filas son la altura (cols y rows).
<code>&lt;select name="nombre"&gt; &lt;option value="valor1"&gt; Valor 1 &lt;/option&gt; &lt;/select&gt;</code>	Menú desplegable (combobox) con 1 ítem ("Valor 1")
<code>&lt;input type="text" name="nombre" size=30&gt;</code>	Crea una caja de texto para entrada de información (una única línea)
<code>&lt;input type="checkbox" name="trabaja"&gt;</code>	Crea un checkbox que se puede seleccionar según corresponda ( en este caso si trabaja o no)
<code>&lt;input type="radio" name="sexo" value="Femenino"&gt;</code>	Crea un botón de radio que se puede seleccionar . Se generan como grupos de botones excluyentes.
<code>&lt;input type="submit" value="nombre"&gt;</code>	Crea botón de envío de formulario (tipo submit)
<code>&lt;input type="reset"&gt;</code>	Crea botón de reset (limpieza). Vacía el formulario
<code>&lt;input type="button"&gt;</code>	Crea botón
<code>&lt;input type="hidden"&gt;</code>	Crea un campo de ingreso oculto
<code>&lt;input type="password"&gt;</code>	Crea una caja de texto enmascarada



# Campos de un formulario

## NUEVOS VALORES EN EL ATRIBUTO TYPE DE LA ETIQUETA INPUT (HTML5)

Etiquetas(Ejemplo)	Descripción
<b>Tipo email</b> <input type="email" name="miemail" id="miemail">	Permite ingresar un cuadro de texto para cargar un email que es validado(con HTML5)
<b>Tipo dirección url</b> <input type="url">	Permite ingresar direcciones WEB absolutas, retorna error si el formato no es válido.
<b>Tipo teléfono</b> <input type="tel">	Permite ingresar números de teléfono, no requiere sintaxis particular , es una indicación para el navegador de acuerdo al dispositivo en el cual la aplicación es ejecutada.
<b>Tipo número</b> <input type="number" min="0" max="10" step="5">	Permite ingresar entradas numéricas. En el ejemplo, si se declara un valor 5 para el atributo step en un campo que tiene valor mínimo de 0 y máximo de 10, el navegador no permitirá especificar valores entre 0 y 5 y entre 5 y 10.
<b>Tipo rango</b> <input type="range" min="0" max="10" step="5">	Es un nuevo control que permite al usuario elegir un valor a partir de un rango.

# Campos de un formulario

## NUEVOS VALORES EN EL ATRIBUTO TYPE DE LA ETIQUETA INPUT (HTML5)

Etiquetas(Ejemplo)	Descripción
<b>Tipo fecha</b> <code>&lt;input type="date" name="fecha"&gt;</code>	Es un nuevo control que muestra un calendario cuando el usuario hace clic sobre el campo. El valor generado y esperado tiene la sintaxis año-mes-día.
<b>Tipo semana</b> <code>&lt;input type="week" name="semana"&gt;</code>	Es un nuevo control que permite seleccionar una semana en el año. Su sintaxis es año-semana (2016-w50)
<b>Tipo mes</b> <code>&lt;input type="month" name="mes"&gt;</code>	Es un nuevo control que permite seleccionar un mes en el año. Su sintaxis es año-mes.
<b>Tipo hora</b> <code>&lt;input type="time" name="hora"&gt;</code>	Es un nuevo control que permite seleccionar una hora. Su sintaxis es hora-minutos-segundos.
<b>Tipo fecha/hora</b> <code>&lt;input type="datetime" name="fechahora"&gt;</code>	Sirve para ingresar la fecha y hora completa, incluyendo la franja horaria.

# Nuevos Atributos en HTML5

Etiquetas(Ejemplo)	Descripción
<b>Requerido</b> <input type="email" <b>required</b> >	Este atributo no permite que se envíe el formulario si el campo se encuentra vacío
<b>Multiple</b> <input type="email" <b>multiple</b> >	En campos de tipo email, o url permite varias entradas, las mismas deben ser separadas por comas y se validan una a una.
<b>Autofoco</b> <input type="datetime" name="fechahora" <b>autofocus</b> >	Anteriormente el foco se ubicaba en los campos con el método focus() de <i>JavaScript</i> para seleccionar un campo; pero no tenía en cuenta si el usuario seleccionaba otro. Actualmente autofocus toma el foco cuando el usuario se posiciona sobre el elemento.
<b>Patrón</b> <input <b>pattern</b> ="[0-9]{4}" name="codigopostal" title="Inserte los cuatro dígitos de su código postal">	Sirve para validar a través de expresiones regulares que permiten personalizar reglas de validación, como por ejemplo validar el código postal con 4 dígitos.

# Unidad 3: CSS Básico - Modelo de Caja

Sintaxis	2
Formas de vincular css con el html	3
Selectores	5
1. Selector Universal	5
2. Selector de etiqueta	5
3. Selector de Clase	6
4. ID	7
Propiedades CSS	7
Modelo de caja	10
Vínculos	16
<b>Etiquetas de Bloque y de línea</b>	<b>18</b>
<b>Propiedad Display</b>	<b>18</b>

# Sintaxis

El código css, tiene una sintaxis diferente al del HTML.

La sintaxis css, sigue el siguiente formato:

```
selector { propiedad: valor; }
```

El selector es aquello a lo que le queremos dar formato.

La propiedad es la característica visual que queremos modificar.

Por ejemplo:

```
h1 { color: blue; }
```



h1 sería el selector, la propiedad a la que le vamos a asignar un valor es el color

(representa el color de fuente), y el valor en este caso es blue.

Esta regla css, establece que todas las etiquetas h1 de nuestros archivos html, van a tener color azul, como color de fuente.

Cada propiedad que yo quiera modificar, representa una declaración.

Y todas las declaraciones en css, terminan con punto y coma.

Si quiero modificar varias propiedades para un mismo selector, separo cada declaración con un punto y coma.

En este ejemplo estoy estableciendo que todos los párrafos van a ser de color azul, y el tipo de fuente tahoma.

```
p {  
    color: blue;  
    font-family: tahoma;  
}
```

## Formas de vincular css con el html

Hay tres formas establecer un estilo CSS en un documento HTML:

1. En línea (Incrustado dentro de una etiqueta HTML)

```
<h2 style="color: blue;" > Subtítulo </h2>
```

Esta es la forma menos recomendada.

2. Dentro del mismo documento, dentro de las etiquetas `<style></style>`

Las etiquetas style, van dentro del head del documento.

```
<style>

    h1{

        color: blue;

    }

</style>
```

3. En un archivo .css externo. Esta es la forma más recomendada, ya que permite, poder vincular todos los archivos de un sitio a un mismo archivo css, y compartir estilos entre ellos, de forma más optimizada y mantenible.

```
<html>

    <head>

        <link rel="stylesheet" href="css/estilos.css">

    </head>

    <body>...
```

Lo primero que tengo que hacer es crear un archivo con la extensión .css

Y después dentro de la etiqueta head del html, lo vinculo a través de la etiqueta link.

La etiqueta link tiene dos atributos:

- rel  
establece que tipo de archivo voy a vincular. El valor para las hojas de estilos es stylesheet

quedando: `rel="stylesheet"`

- href  
en este atributo la ruta a la ubicación del archivo css  
`href="css/estilos.css"`

## Selectores

### 1. Selector Universal

Se utiliza para seleccionar todos los elementos de la página. El selector universal es el \*

```
* {  
  
    margin: 0;  
  
    padding: 0;  
  
}
```

El selector universal se indica mediante un asterisco (\*). A pesar de su sencillez, no se usa habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página. Sirve por ejemplo, para sacar el formato por defecto que tienen todas las etiquetas html.

### 2. Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML correspondiente a los elementos que se quieren seleccionar.

El siguiente ejemplo aplica diferentes estilos a los encabezados y a los párrafos de



una página HTML:

```
h2{  
  
  color: red;  
  
}  
  
p {  
  
  color: black;  
  
}
```

### 3. Selector de Clase

El selector de clase, sirve para aplicar estilos a ciertos elementos de una página web. Por ejemplo si quisiera aplicarle un estilo bold a algunos párrafos de mi sitio. En ese caso, la etiqueta p, no me serviría ya que se los aplicaría a todos.

Para poder utilizar el selector de clase, hay que agregar el atributo class de HTML sobre el elemento que quiero seleccionar.

```
<p class="destacado"> Párrafo destacado</p>  
  
<p>Párrafo común.</p>
```

Una vez que tengo el html, creo en el archivo CSS una nueva regla llamada .destacado con los estilos que quiero aplicar.

En el css, los selectores de clase se indican anteponiendo al nombre de la clase un punto (.)

```
.destacado { font-weight: bold; }
```

Aplica lo declarado a todos los elementos que posean la clase indicada. Puede repetirse muchas veces en una misma página.

## 4. ID

El atributo id, es otra de las maneras de poder aplicar estilos a un elemento específico de nuestra página web. A diferencia de las clases, que se pueden repetir varias veces en una misma página, el atributo id, es unívoco, es decir, solo un elemento en nuestro html puede tener un mismo valor de id.

En el html:

```
<div id="contenedor"> </div>
```

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo del numeral (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#contenedor{ color: red; }
```

### Propiedades CSS

Propiedad	Valores
<b>color</b> <i>Color del texto</i>	Valores posibles: 1. Nombre de color green   red   blue   orange .... <pre>p{   color: red; }</pre>

	<p>2. Código hexadecimal (# antes del código hexadecimal)</p> <pre>p{   color: #000099; }</pre>
	<p>3. Color RGBA (red, green, blue, alpha)</p> <p>Los valores, van separados por comas. La A, representa la transparencia, los valores posibles van del 0 al 1 en decimales separados por punto.</p> <pre>p{   color: rgba(255,0,25,1); }</pre>
<p><b>font-family</b></p> <p><i>tipo de fuente</i></p>	<p>Arial, san-serif;</p> <p>Times New Roman, serif;</p>
	<pre>p{   font-family: Tahoma, san-serif; }</pre>

<b>font-size</b>  <i>tamaño de letra</i>	Medidas posibles : <ul style="list-style-type: none"> <li>• px (píxeles)</li> <li>• em (medida relativa)</li> <li>• % (porcentajes)</li> </ul> Ejemplo: 12px, 1em, 80%
	<pre>h1{     font-size:12px; }</pre>
<b>font-style</b>  <i>estilo</i>	Valores Posibles: normal   italic   oblique
	<pre>p{     font-style: italic; }</pre>
<b>font-weight</b>  <i>grosor de letra</i>	Valores posibles: 100   200   400 600   normal   bold
	<pre>p{     font-weight: bold; }</pre>

<b>text-align</b>	Valores posibles: left   right   center justify
<i>alineación de texto</i>	<pre>p{     text-align:center; }</pre>

## Modelo de caja

El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.

Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde.

Bajo este modelo, toda etiqueta html, cuenta con:

Un ancho (propiedad width)

- Un Alto (height)
- Un borde ( que puede ser sólido, a rayas, con puntos, con algún grosor, con algún color o incluso sin borde, esta propiedad es border)



- Un margen interno (padding)
- Un margen externo (margin)
- Un fondo (que puede ser un color o una imagen)

Propiedad	Valores
<b>width</b> <i>ancho</i>	Medidas posibles : px , em, % (porcentajes)  <pre>div{ width: 300px; }</pre>
<b>height</b> <i>alto</i>	Medidas posibles : px , em, % (porcentajes)  <pre>div{height: 100px;}</pre>
<b>padding</b> <i>Margen interior o relleno</i>	Medidas posibles : px , em, % (porcentajes)  <pre>p{     padding: 5px; }</pre> Agrega un padding en los 4 lados de la caja.

	<p>padding-top: 5px; (agrega un padding superior)</p> <p>padding-left: 5px; (agrega un padding izquierdo)</p> <p>padding-bottom: 5px; (agrega un padding inferior)</p> <p>padding-right: 5px; (agrega un padding derecho)</p>
<p><b>margin</b></p> <p><i>margen externo</i></p>	<p>Medidas posibles : px , em, % (porcentajes)</p>
	<pre>div{     margin: 5px; }</pre> <p>Agrega un margen en los 4 lados de la caja.</p>



	<p><code>margin-top: 5px;</code> (agrega un margen superior)</p> <p><code>margin-left: 5px;</code> (agrega un margen izquierdo)</p> <p><code>margin-bottom: 5px;</code> (agrega un margen inferior)</p> <p><code>margin-right: 5px;</code> (agrega un margen derecho)</p>
<p><b>border-color</b> <i>color de borde</i></p>	<p>Valores posibles: Nombre de color, Código hexadecimal, RGBA</p> <pre>div{     border-color: #000000; }</pre>



<b>border-style</b> <i>estilo de borde</i>	Valores posibles: solid   dashed   dotted   none   double
	<pre>div{     border-style:solid; }</pre>
<b>border-size</b> <i>grosor de borde</i>	Medidas posibles : px , em, % (porcentajes)
	<pre>p{     border-size:2px; }</pre>
<b>border</b> <i>borde (propiedad genérica)</i>	<p>Esta propiedad permite combinar todas las propiedades que vimos vinculadas a bordes en una sola.</p> <p>Es un genérico que nos permite especificar en la misma propiedad, el grosor tipo y tamaño.</p>
	<pre>p{     border:2px solid #000000; }</pre>

	<p>Al igual que con el margin y el padding, se puede especificar algún lado específico de la caja:</p> <p>border-left: 2px;</p> <p>border-bottom: 2px;</p> <p>border-right: 2px;</p> <p>border-top: 2px;</p>
<p><b>background-color</b></p> <p><i>color de fondo</i></p>	<p>Valores posibles:</p> <ol style="list-style-type: none"> <li>1. Nombre de color</li> <li>2. Código hexadecimal</li> <li>3. RGBA (en caso de querer un fondo de color con transparencia)</li> </ol> <pre>div{     background-color: #cc4433; }</pre>

<b>background-image</b> <i>imagen de fondo</i>	<p>La ubicación de la imagen se especifica con url(ubicación). Por ejemplo:</p> <pre>body{     background-image: url(imagen.jpg); }</pre>
<b>background</b> <i>fondo (propiedad genérica)</i>	<p>Al igual que border, existe la propiedad genérica background</p> <pre>body{     background: #cc4433 url(imagen.jpg); }</pre>

## Vínculos

La etiqueta a, de html es la que nos permite generar vínculos.

Los vínculos (links), tienen diferentes estados. El estado más conocido es el estado hover, que es cuando pasamos el mouse por encima de un vínculo, hay otros estados, como el estado active, que es cuando hacemos clic sobre el vínculo, o el estado visited, que es cuando ya visitamos un link.

Los vínculos por defecto son de color azul y subrayados.

Con la propiedad color, podemos modificarle el color a cualquier vínculo.

Y si quisiéramos sacarle el subrayado, la propiedad que tendríamos que utilizar es text-decoration.

Propiedad	Valores
text-decoration	Valores Posibles: none underline overline  a{  text-decoration: none;  }

Si quisiéramos que algo se modificara en un vínculo al pasar el mouse, tendríamos que poder darle formato, a un estado específico del vínculo, que es el estado hover.

Para poder darle formato, tenemos que poder seleccionarlo. El selector de estados, tienen el siguiente formato, `nombreDeEtiqueta:estado`

Por ejemplo `a:hover {}`

```
a{ text-decoration:none;}
```

```
a:hover{  
    color:red;  
    text-decoration:underline;  
}
```

# Etiquetas de Bloque y de línea

Existen dos tipos de etiquetas: las etiquetas de bloque y las etiquetas de línea.

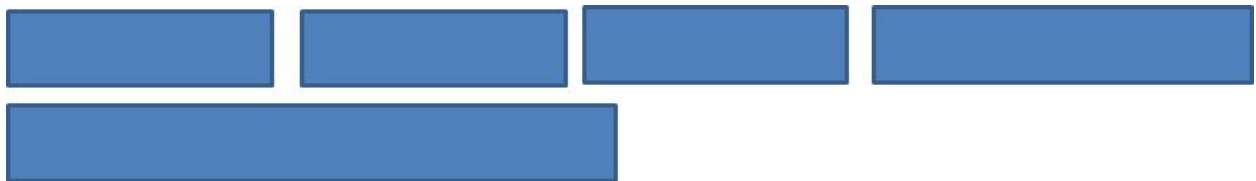
- Las etiquetas de bloque, ocupan el 100% del ancho, y el alto de su contenido.  
Ejemplos de etiqueta de bloque: div, p, h1-h6, article, section, main, nav, aside, header, footer, ul, li.



Al ocupar el 100% del ancho de la pantalla, no pueden ponerse una al lado de la otra, sino que van una debajo de la otra.

- Las etiquetas de línea, no tienen ancho por defecto, y su ancho es el ancho de su contenido. Por ende se ponen una al lado de la otra, si tuviera varias.  
A diferencia de las de línea, no pueden empujarse por los márgenes de arriba y abajo, ni tampoco centrarse en la pantalla, requieren para eso estar dentro de una etiqueta de bloque contenedora.

Ejemplos de etiqueta de línea: a, img, label, span.



## Propiedad Display

La propiedad display, me permite cambiar una etiqueta de bloque a línea, o de línea a bloque u ocultar un elemento. También sirve para habilitar dos propiedades que

vamos a estar viendo en las próximas clases que son flexbox y grid, que nos van a permitir maquetar a varias columnas.

Algunos de los valores posibles:

- **display: none;**  
Oculta el elemento
- **display: block;**  
Convierte un elemento en bloque
- **display: inline;**  
Convierte un elemento en línea
- **display: inline-block;**  
Convierte un elemento en línea bloque, esto en concreto es que el elemento, tiene solo el ancho de su contenido y se pone uno al lado del otro, como el comportamiento de las etiquetas de línea, pero toma el margin-top y margin-bottom, que no funcionan en etiquetas de línea
- **display: flex;**  
Habilita flexbox
- **display: grid;**  
Habilita grid

# Flexbox

## Flexbox

Flexbox es un módulo de css, para crear diseños flexibles.

Establece una serie de propiedades que nos permiten posicionar los elementos HTML, tanto horizontal como verticalmente con mayor facilidad que con los flotados.

Requiere que establezcamos un elemento contenedor y dentro de ese elemento contenedor, una serie de ítems que se adaptan, ajustando sus tamaños y disposición en función del espacio disponible de la caja contenedora.

Podemos incluso, desde el css, alterar el orden de los elementos y modificar la forma en que los elementos se ubican en pantalla (sin modificar el html).

## Definir flexbox:

Para poder utilizar flexbox, necesitamos crear un elemento contenedor (padre) y una serie de elementos hijos. Como el siguiente ejemplo:

```
<section class="contenedor">
  <article class="item1">1</article>
  <article class="item2">2</article>
  <article class="item3">3</article>
</section>
```

Al elemento contenedor le agregamos la propiedad display, con el valor flex, que establece que se trata de una caja flexible.

```
.contenedor{
```

```
display: flex;  
}
```

## Propiedades para contenedores:

### 1.- flex-direction

Esta propiedad establece de qué forma los elementos hijos del contenedor, se van a posicionar.

```
.contenedor{  
    flex-direction: row;  
}
```

Tiene 4 valores posibles:

#### a) flex-direction: row;

Ordena los elementos horizontalmente de izquierda a derecha en una fila.

#### b) flex-direction: row-reverse;

Ordena los elementos horizontalmente de derecha a izquierda en una fila e invierte el orden de los mismos.



c) **flex-direction:column;**

Ordena los elementos verticalmente de arriba abajo en una columna.

d) **flex-direction:column-reverse;**

Ordena los elementos verticalmente de abajo arriba en una columna.

## 2.- flex-wrap

Redistribuye los elementos para que entren una o más líneas.

a) **flex-wrap: no-wrap;**

Redistribuye los elementos para que entren en una sola línea, achicándolos si es necesario.

b) **flex-wrap: wrap;**

Redistribuye los elementos en varias líneas de izquierda a derecha y de arriba a abajo.

c) **flex-wrap: wrap-reverse;**

Redistribuye los elementos en varias líneas de derecha a izquierda y de abajo a arriba.

## 3.- justify-content

Distribuye la posición de los elementos y la disposición que tendrán en el eje **horizontal**.

a) **Justify-content: flex-start;**

Este es el valor por defecto, alinea los elementos a la izquierda.

b) **Justify-content: flex-end;**

Alinea los elementos a la derecha.

**c) Justify-content: center;**

Centra horizontalmente los elementos.

**d) Justify-content: space-around;**

Distribuye los elementos, dejando el mismo espaciado entre ellos.

**e) Justify-content: space-between;**

Distribuye los elementos ubicando el primero pegado a la izquierda del contenedor (sin espacio hacia la izquierda) y el último sin espacio a la derecha.

#### **4.- align-items**

Distribuye la posición de los elementos y la disposición que tendrán en el eje **vertical**.

**a) align-items: stretch;**

Los elementos hijos, ocupan todo el ancho (o alto) del elemento contenedor.

**b) align-items: flex-start;**

Los elementos se ubican desde el inicio del borde superior del contenedor.

**c) align-items: flex-end;**

Los elementos se ubican desde el inicio del borde inferior del contenedor.

**d) align-items: center;**

Los elementos se ubican centrados verticalmente.

**e) align-items: baseline;**

Alinea los elementos tomando como línea base la altura del texto.

## 5.- align-content

Permite distribuir verticalmente los elementos cuando hay varias filas.

# Propiedades para items:

## 1.-order

Con esta propiedad, podemos cambiar el orden de cualquier elemento hijo. Por ejemplo:

```
.item1{  
    order:3;  
}
```

## 2.-flex-grow

Es el crecimiento que va a tener un elemento en relación a los demás elementos hermanos. El valor por defecto es 1, si algún elemento tiene un valor de flex-grow mayor, el resto de los elementos, se redistribuyen en ancho.

Por ejemplo:

```
.item2{  
    flex-grow:2;  
}
```

## 3.-flex-shrink

Esto define la posibilidad de que un elemento flexible se contraiga si es necesario.

## 4.-flex-basis

Sirve para establecer un valor inicial de ancho o de alto de los elementos, previo a que ese ancho o alto sea modificado por las otras propiedades.

```
.item1{
```

```
flex-basis:30em;  
}
```

### 5.-align-self

Esta propiedad permite alinear un elemento en forma individual.

Los valores posibles son:

auto | flex-start | flex-end | center | baseline | stretch

## Compatibilidad con navegadores:

Como flexbox aún no es compatible con todos los navegadores, debemos agregar los prefijos -ms- , -webkit- a las propiedades de flexbox.

Podemos ver qué navegadores son compatibles en el sitio Can I Use:

<http://caniuse.com/#search=flexbox>

## Web para probar propiedades de flexbox:

<https://demos.scotch.io/visual-guide-to-css3-flexbox-flexbox-playground/demos/>

## Referencias:

- W3 Schools Css3 Flexbox

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

- A visual Guide to flexbox properties



Autor: Dimitar Stojanov

<https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties>

- Css-Tricks- A Complete Guide to Flexbox

Autor: [Chris Coyier](#)

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

- Understanding Flexbox: Everything you need to know

Autor: Ohans Emmanuel

<https://medium.freecodecamp.com/understanding-flexbox-everything-you-need-to-know-b4013d4dc9af#.mmfc1uaov>

# Unidad 3: Propiedades Avanzadas

Propiedades Avanzadas	1
@Font-face	1
Border-radius	2
Text-shadow	4
Box-shadow	4
Gradient	5
Opacity	5
Transform	6
Background	7
Background Repeat	7
Background Position	8
Los valores posibles para el eje horizontal(x) son:	8
Los valores posibles para el eje vertical(y) son:	8
Ejemplos de combinaciones:	8
Background Size	9
<b>Position</b>	<b>9</b>
Position Static	10
Position Relative	10
Position Absolute	11
Position Fixed	11
z-index	12

# Propiedades Avanzadas

## @Font-face

Para poder utilizar una fuente por fuera de las fuentes seguras (Arial, Times New Roman, Comic Sans, Tahoma, Verdana, etc), que son fuentes que siempre están instaladas en las computadoras de los usuarios, tenemos que tener nuestra fuente convertida a estos formatos:

- TTF (Chrome, Safari, Firefox, Opera)
- EOT (Explorer 6, 7 y 8)
- WOFF (Explorer 9, Firefox 3.6+, Chrome 5+)
- SVG (Opera, Chrome, iPhone iOS)

Una vez convertida, hay que crear un @font-face como primera instancia.

```
@font-face {  
  font-family: 'robotoregular';  
  src: url('Roboto.eot');  
  src: url('Roboto.svg');  
  src: url('Roboto.ttf');  
  src: url('Roboto.woff');  
}
```

Dentro de la función @font-face, hay que tener una propiedad font-family, con el nombre de la fuente entre comillas simples. Y un src, con la ubicación de cada archivo de la fuente en cada formato.

Una vez definida una fuente con @font-face, podemos usarla en cualquier etiqueta o clase, usando la propiedad font-family usando el nombre de la fuente establecido anteriormente entre comillas simples.

```
body{  
  font-family: 'robotoregular';  
}
```

Existen herramientas que permiten convertir las fuentes a estos formatos, como por ejemplo [Font-squirrel](#) o y también repositorios de fuentes @font-face, como el caso de google font, en donde directamente importamos la fuente y la referenciamos.

## Border-radius

Esta propiedad permite agregarle a cualquier caja o imagen un borde redondeado. Si tiene un solo valor afecta a los cuatro lados. Por ejemplo

```
article{  
  border-radius: 20px;  
}
```



Si tiene varios valores, cada lado, puede ser independiente en su redondeado e incluso no tener redondeado como en el siguiente ejemplo:

```
article{  
  border-radius: 150px 0 150px 0;  
}
```

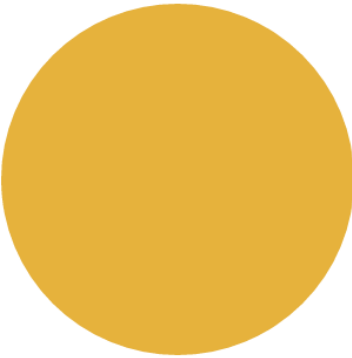


Si queremos hacer un círculo, podemos hacerlo con border-radius, con el valor 50%.

```
article{
```



```
border-radius: 50%;  
}
```



## Text-shadow

La propiedad text-shadow, permite aplicar un sombra a un texto. Posee 4 valores, que representan la distancia en x, en y, la nitidez de la sombra, y el color expresado en rgba (para darle opacidad).

## Sombra en Texto

```
h1{  
  text-shadow: 1px 1px 1px  
    rgba(0, 0, 0, 0.5);  
}
```

## Box-shadow

Esta propiedad permite darle sombra a las cajas o a imágenes.

Puede tomar cinco parámetros: el color, el desplazamiento horizontal, el desplazamiento vertical, el valor de difuminación, y la palabra clave inset para generar una sombra interna.

```
article{
```

```
box-shadow: 10px 10px 0px -2px rgba(0,0,0,0.2);  
}
```

## Gradient

Existen dos funciones para generar gradientes, linear-gradient y radial-gradient.

1. La función linear-gradient(posición inicio, color inicial, color final) puede ser aplicada a las propiedades background o background-image para generar un gradiente lineal.
  - Los atributos indican el punto inicial y los colores usados para crear el gradiente.
  - El primer valor puede ser especificado en pixeles, en porcentaje o usando las palabras clave top, bottom, left y right.

```
body{  
    background: linear-gradient  
    (to bottom, rgba(147,206,222,1) 0%,  
    rgba(117,189,209,1) 41%,      rgba(73,165,191,1) 100%);  
}
```

2. La función radial-gradient(posición inicio, forma, color inicial, color final) puede ser aplicada a las propiedades background o background-image para generar un gradiente radial.
  - La posición de inicio es el origen y puede ser declarado en pixeles, porcentaje o como una combinación de las palabras clave center, top, bottom, left y right.

Para ambos tipos de gradientes, existen generadores visuales como  
<https://www.cssmatic.com/es/gradient-generator>

## Opacity

La propiedad opacity, permite modificar la opacidad de una caja u elemento. Va en valores decimales entre 0 y 1, siendo 0 sin opacidad y 1 opacidad al 100%. El valor por defecto de opacity es 1.

```
article{
```

`opacity: 0.5; //` en este ejemplo, la opacidad está al 50%.  
`}`

### Probando Opacidad de un elemento

Todo el elemento tendrá menor opacidad

### Probando Opacidad de un elemento

Todo el elemento tendrá menor opacidad

## Transform

Esta propiedad modifica la forma de un elemento. Utiliza cuatro funciones básicas: scale (escalar), rotate (rotar), skew (inclinarse), y translate (trasladar o mover).

- La función scale recibe solo un parámetro. Un valor negativo invierte el elemento, valores entre 0 y 1 reducen el elemento y valores mayores que 1 expanden el elemento (por ejemplo, transform: scale(1.5);).
- La función rotate usa solo un parámetro expresado en grados para rotar el elemento (por ejemplo, transform: rotate(20deg);).
- La función skew recibe dos valores, también en grados, para la transformación horizontal y vertical (por ejemplo, transform: skew(20deg, 20deg);).
- La función translate mueve el objeto tantos píxeles como sean especificados por sus parámetros (por ejemplo, transform: translate(20px);).

## article{

`transform: scaleX(1);`  
`}`

**Scale** (en decimales expresados con valores numéricos separados por decimales expresados con .)

Escala de 100%, que es el tamaño normal de un objeto, es 1.

Podemos escalar, pasando dos valores, eso representan x e y, o especificar en qué eje queremos escalar.

- `transform:scale(x,y)`
- `transform:scale3d(x,y,z)`
- `transform:scaleX()`
- `transform:scaleY()`
- `transform:scaleZ()`

### **Translate (se expresa en px o ems)**

- `transform:translate(x,y)`
- `transform:translate3D(x,y,z)`
- `transform:translateX()`
- `transform: translateY()`
- `transform:translateZ()`

### **Skew (se expresa en deg)**

- `transform:skew(x,y)`
- `transform:skewX()`
- `transform: skewY()`

### **Rotate (se expresa en deg)**

- `transform:rotate(x,y)`
- `transform:rotateX()`
- `transform: rotateY()`

## **Background**

### **Background Repeat**

La propiedad `repeat`, sirve para controlar la repetición de un fondo. Por defecto tiene

el valor repeat, con lo cual si tenemos un fondo de un div, o de un body y ese fondo es de menor tamaño se repite como retícula.

Los otros valores que puede tomar la propiedad repeat son:

- **repeat-none**  
`background-repeat: repeat-none;`  
Esto hace que no se repita. Es decir se muestra una sola vez
- **repeat-x**  
`background-repeat: repeat-x;`  
Repite el fondo solo en el eje x (es decir horizontalmente)
- **repeat-y**  
`background-repeat: repeat-y;`  
Repite el fondo solo en el eje y (es decir verticalmente)

Más info: [http://www.w3schools.com/cssref/pr\\_background-repeat.asp](http://www.w3schools.com/cssref/pr_background-repeat.asp)

## Background Position

La propiedad background position, establece la posición inicial de una imagen de fondo

Tiene dos valores que representan el eje x y el eje y

Estos valores pueden estar expresados en píxeles, porcentajes o valores en palabras.

**Los valores posibles para el eje horizontal(x) son:**

- left
- center
- right

**Los valores posibles para el eje vertical(y) son:**

- top
- center
- bottom

**Ejemplos de combinaciones:**

- `background-position: center top;`
- `background-position: left center;`
- `background-position: 30% top;`
- `background-position: center -100px;`

Más info: [http://www.w3schools.com/cssref/pr\\_background-position.asp](http://www.w3schools.com/cssref/pr_background-position.asp)

## Background Size

La propiedad background-size, nos permite controlar el tamaño de un fondo. Por defecto está en auto, es decir que un fondo se muestra en el tamaño que tiene la imagen. Con la propiedad background-size, podemos hacer que un fondo se adapte al ancho, al alto o al ancho y alto (acá se deformaría) de un contenedor.

Los otros valores que puede tomar la propiedad background-size son:

- **cover**  
`background-size:cover;`  
El fondo se adapta al ancho del contenedor ocupando todo su ancho, el alto es proporcional.
- **contain**  
`background-size:contain;`  
El fondo se adapta al alto del contenedor ocupando todo su alto, el ancho es proporcional.
- **porcentaje**  
`background-size:100% 100%;`  
Se puede también establecer el tamaño en porcentajes o en píxeles, el primer valor representa el ancho y el segundo el alto. En este ejemplo, el valor representa que ocupa todo el ancho y todo el alto del contenedor, deformando el fondo (no mantiene la proporción de origen).

Más info: [http://www.w3schools.com/cssref/css3\\_pr\\_background-size.asp](http://www.w3schools.com/cssref/css3_pr_background-size.asp)

## Position

Permite ubicar un elemento en un lugar exacto de la pantalla.

Valores posibles:

- static
- relative
- absolute
- fixed

## Position Static

- Es el valor por defecto que tienen todos los elementos.
- Significa que ese elemento no está posicionado de ninguna manera, sino que está en su ubicación por defecto.

## Position Relative

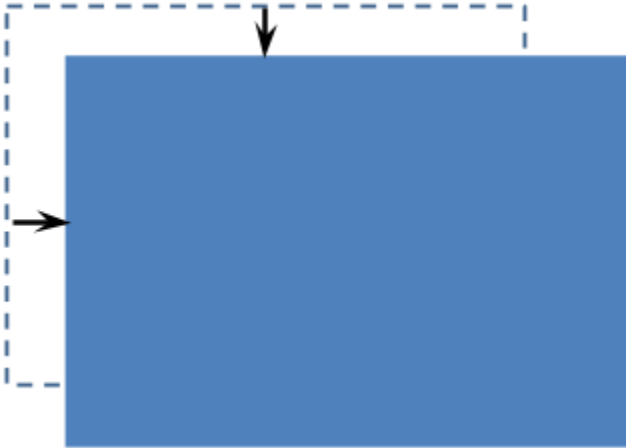
- Posiciona el elemento a partir de su posición original.
- Permite mover ese elemento desde su ubicación original, a la izquierda, derecha, arriba o abajo.

Para mover un elemento a la izquierda, arriba, abajo o derecha de su posición original, si ya tiene position relative, uso:

- top
- left
- right
- bottom

Ejemplo:

```
article{  
    position: relative;  
    top: 5px;  
    left: 5px;  
}
```



## Position Absolute

Un elemento posicionado absolutamente está posicionado relativamente a otro elemento, llamado bloque contenedor.

Por defecto es el body, pero puede estar dentro de otro elemento con position relative.

Los elementos con position absolute, salen del flujo del html y pierden su width de 100%, se comportan como si .

Para moverse en forma absoluta, también se utilizan las propiedades: top, left, right, bottom

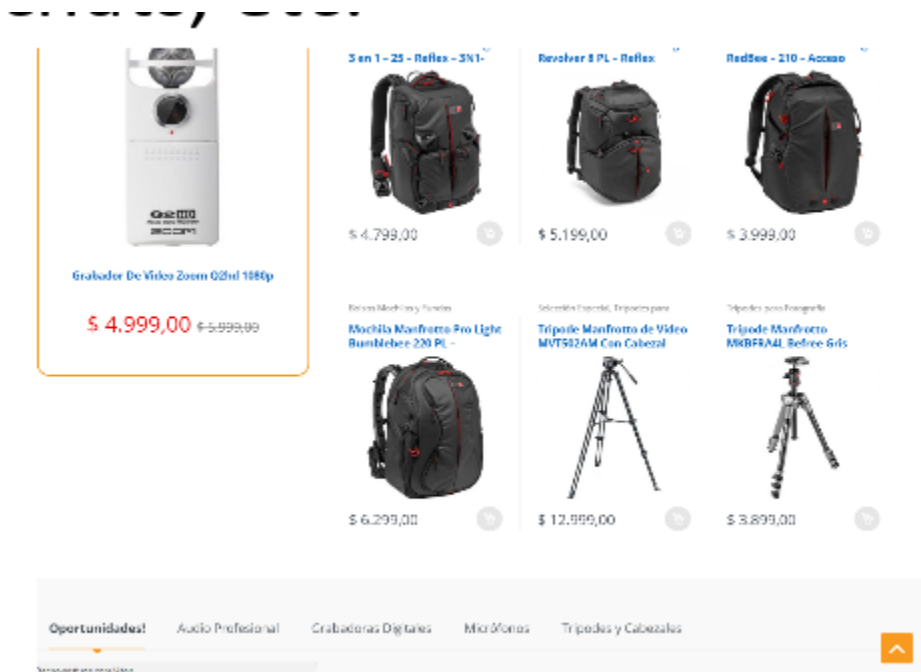
## Position Fixed

Propiedad que sirve para posicionar elementos en forma fija, relativa a la pantalla.

Si hago scroll, igual mantiene su posición.

Suele usarse para botón ir arriba, chats, etc.





## z-index

Propiedad que sirve el orden en la superposición de elementos

Los valores posibles son numéricos.

El elemento que tenga el valor más alto de z-index, se pondrá por arriba del resto.

Por Ejemplo: una elemento con z-index:2; se posicionará por arriba de un elemento con z-index:1;

Solo los elementos html, que tengan position relative, absolute, o fixed, tienen la posibilidad de usar esta propiedad.

# Responsive Web Design



Programación Web I / Móvil I

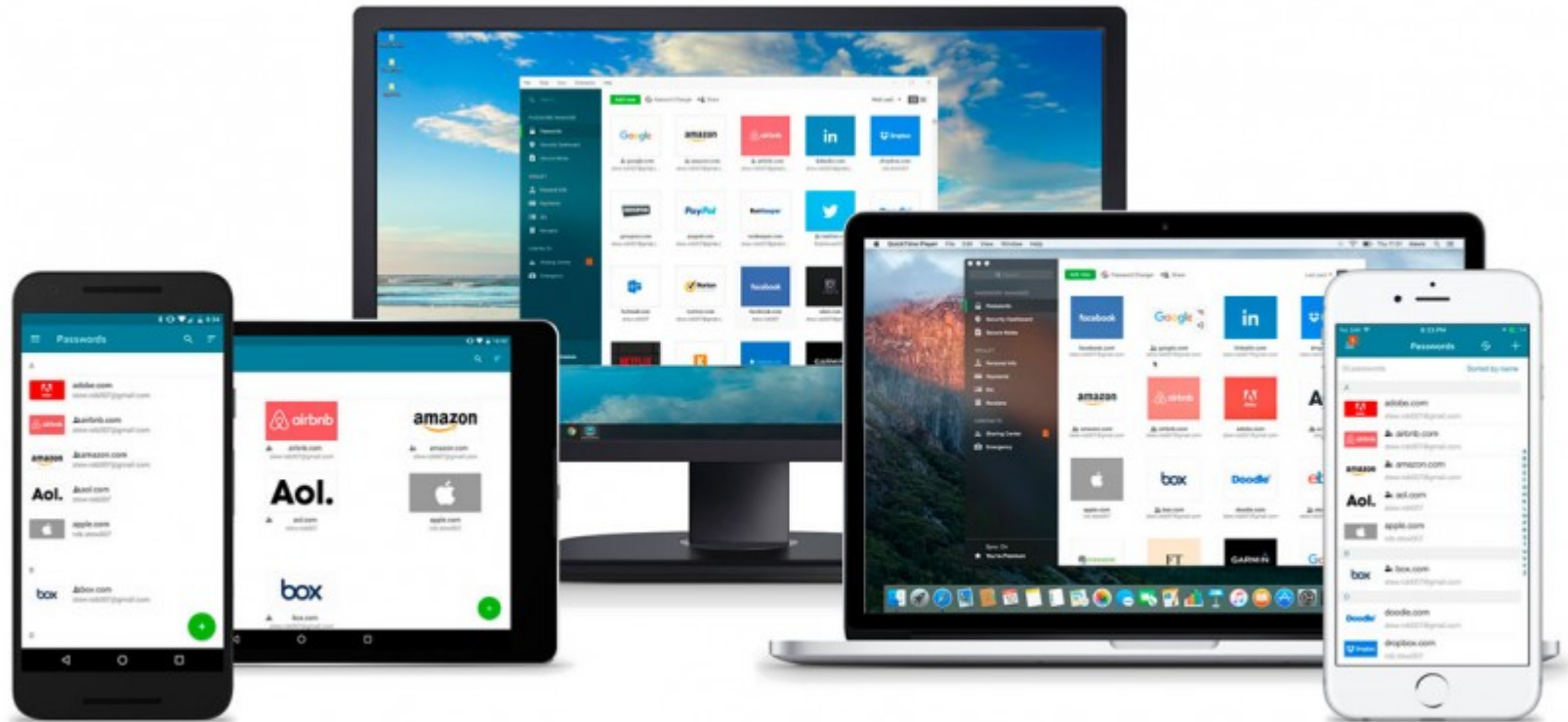
**Comisión Miércoles Noche**

DIS. Alicia Rosenthal

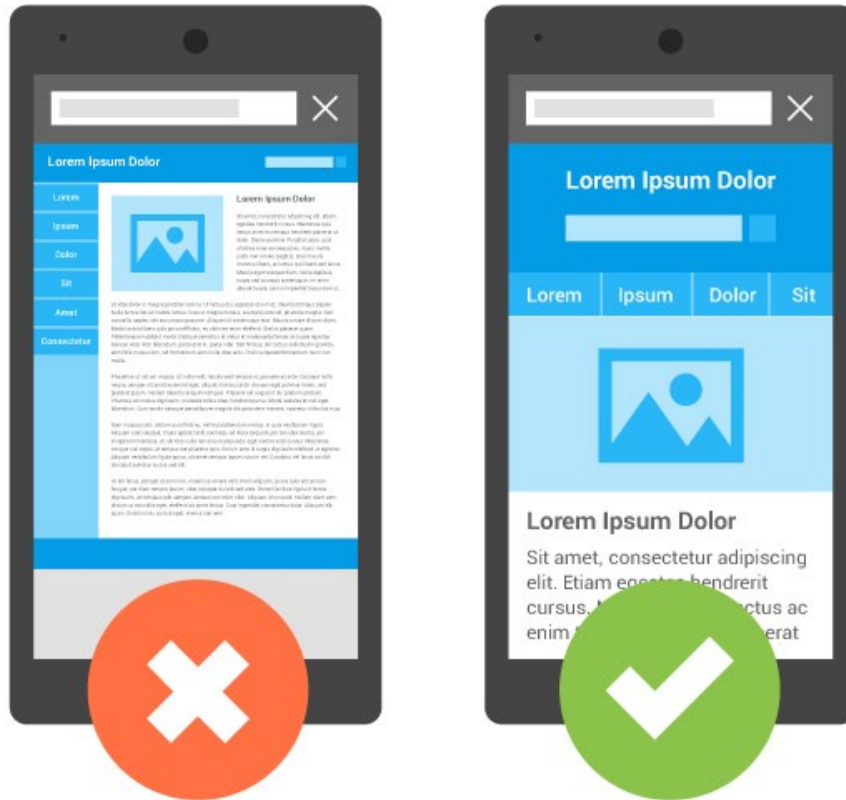
**Comisión Viernes Mañana**

Ing. Gabriel Panik

# Dispositivos



# MetaViewport



[https://www.w3schools.com/css/css\\_rwd\\_viewport.asp](https://www.w3schools.com/css/css_rwd_viewport.asp)

# MetaViewport

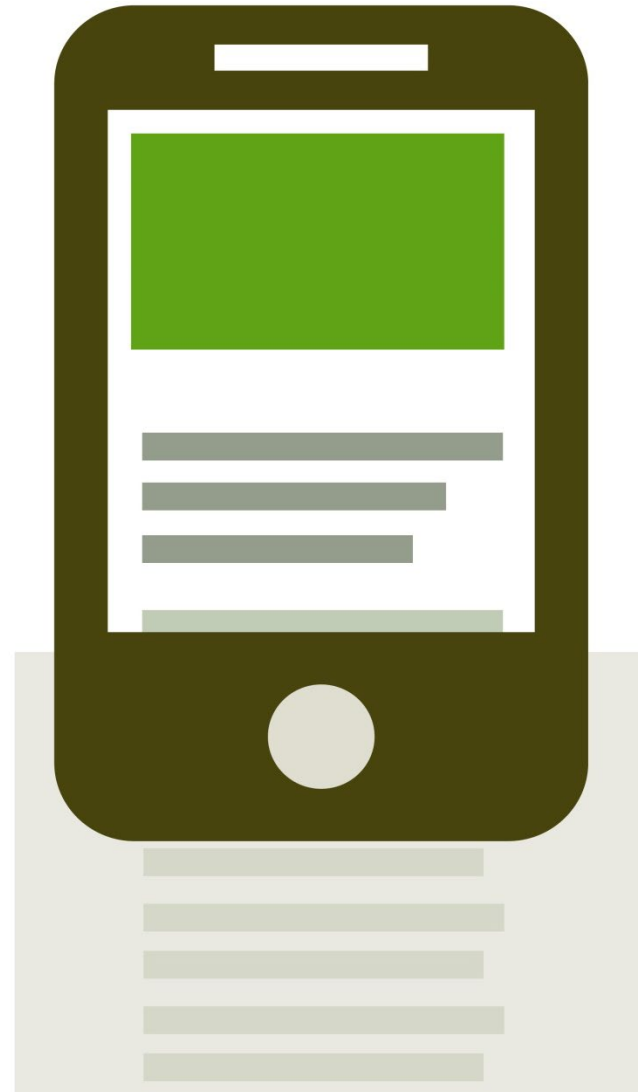
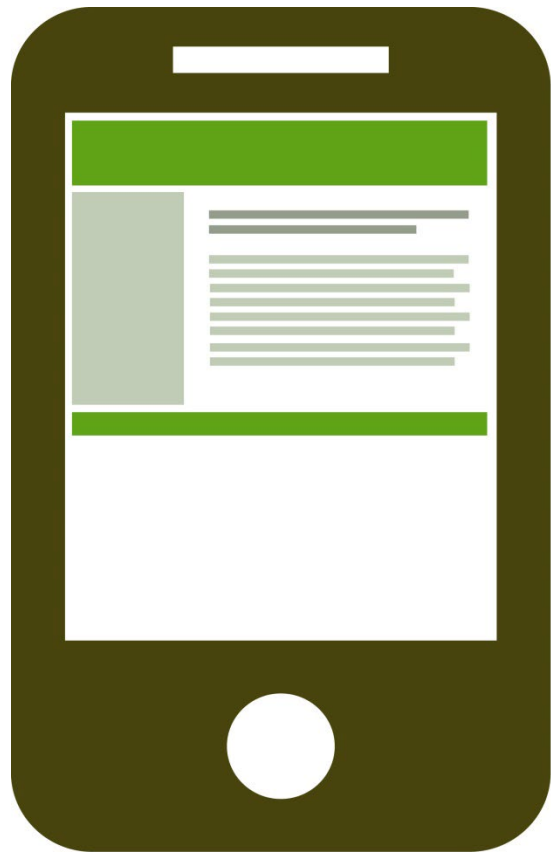
```
<head>
```

```
  <meta name="viewport"  
    content="width=device-width,  
    initial-scale=1, maximum-  
    scale=1">
```

```
</head>
```

Ejemplo: <http://www.edenor.com.ar/>

# MetaViewport



# Medias Queries

**Media Queries** es un módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo.

Se implementa mediante condiciones sobre el CSS que permiten decidir que porciones del mismo deben aplicarse dependiendo el contexto

# Medias Queries

## SINTAXIS:

```
@media type and (condicion) {  
}
```

## Ejemplos de aplicación:

```
@media screen and (min-width: 500px) {  
}
```

```
@media tv and (min-width: 700px) and  
(orientation: landscape) { ... }
```



# Medias Queries

## Dispositivos (media type)

screen

print

all

## Condiciones /expresiones

min-width: valor (expresado en px o en ems)

max-width: valor (expresado en px o en ems)

orientation: landscape | portait

# Medias Queries

**Ejemplo:** Modificar el color de fondo del body si estamos trabajando sobre pantallas con ancho mayor o igual a 768 pixeles:

```
@media screen and (min-width: 768px) {  
    body {  
        background-color: #ccc;  
    }  
}
```

# Medias Queries

**Ejemplo:** Modificar el color del texto de un párrafo a rojo en caso de que el dispositivo posea un ancho en píxeles entre 500 y 700:

```
@media screen (max-width: 700px) and  
(min-width: 500px) {  
  
  p {  
  
    color: red;  
  
  }  
  
}
```

# Mobile First



Responsive Web Design

---

Mobile First Web Design



# Medias Queries

```
body{  
  background: green;  
  font-size: 70%;  
}  
p{  
  font-size: 1em;  
}
```

Estilos Mobile

```
@media screen and (min-width:47em){
```

```
  body{  
    font-size: 90%;  
  }
```

Estilos Tablet

```
}
```

```
@media screen and (min-width:60em){
```

```
  body{  
    font-size: 100%;  
  }
```

Estilos Pcs de  
Escritorio

```
}
```

# Unidades de Medida

Para responsive, nos vamos a manejar con estas unidades de medida:

- **%** porcentaje. Para anchos de columnas, imágenes y video
- **em** para tamaños de fuentes, márgenes, paddings.

**1em = 16 píxeles aprox.**

# Tamaños de fuentes

Establecemos primero los tamaños de fuentes en em.

```
h1{  
    font-size: 1.6em;  
}  
h2{  
    font-size:1.5em;  
}  
p{  
    font-size: 1em;  
}
```

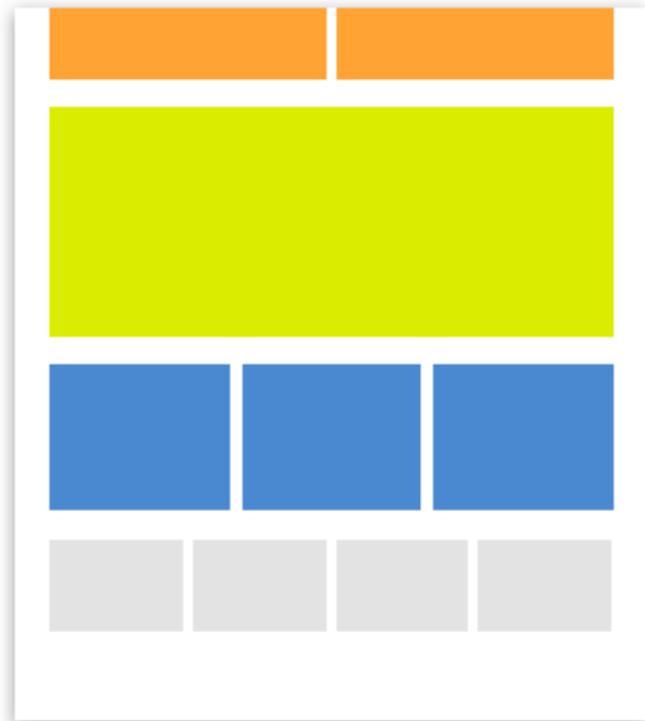
# Tamaños de fuentes

Trabajamos en el body con porcentajes para los distintos tamaños.

```
body{
    font-size:70%;
}
@media screen and (min-width:47em){
    body{
        font-size:80%;
    }
}
@media screen and (min-width:60em){
    body{
        font-size:100%;
    }
}
```



# Columnas



# Columnas

```
.col{
  background-color: #7b1873;
  margin-top:1em;
}
@media screen and (min-width:47em){
  .contenedor{
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
  }
  .col{
    width:calc(50% - 1em);
  }
}
@media screen and (min-width:60em){
  .col{
    width:calc(33% - 1em);
  }
}
```

# Imágenes responsivas

Establecemos la propiedad max-width, que hace que nuestra imagen sea adaptable al ancho disponible, hasta llegar a su ancho máximo que es el ancho que tiene el archivo.

El alto, lo ponemos en automático.

```
img{  
    max-width: 100%;  
    height: auto;  
}
```

# Videos Responsives

```
.contenedor {  
    position: relative;  
    padding-bottom: 56.25%;  
    height: 0;  
}
```

```
.contenedor iframe{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
}
```