

HTML5

- A partir de HTML5 las etiquetas <div> se reemplazaron por 5 etiquetas: <header>, <nav>, <section>, <article>, <footer>, <div> para una mejor indexación.

Etiquetas para formularios y menu desplegables

Label for - inputs

```
<section>
  <form method="post" action="enviar.php">
    <label for="nombre">nombre:</label>
    <input type="text" name="nombre" placeholder="Ingresa tu nombre" required>

    <label for="correo">correo:</label>
    <input type="email" name="correo" placeholder=" Ingresa tu mail" required>

    <label for="telefono">telefono:</label>
    <input type="tel" name="telefono" placeholder="Ingresa tu telefono">

    <select name="provincias" id="prov" required>
      <option value="CABA">Seleccion</option>
      <option value="CABA">CABA</option>
      <option value="Buenos Aires">BS AS</option>
      <option value="Tucuman"> TUCUMAN </option>
    </select>
  </section>
```

- El atributo “value” es el dato que yo recibo, pero el nombre que ve el usuario es el que está dentro de los tags <option>NOMBRE</option>

Git y Github

Abrir el command prompt o bajar el git CMD. Básicamente con Github crea un repositorio remoto y en nuestra pc se crea un repositorio local, que se subira a github mediante git. La carpeta local que genera git se encuentra oculta dentro de nuestro proyecto.

Para subir a GITHUB por primera vez un proyecto hay que seguir los siguientes pasos:

```
Command prompt: cd + ruta
git init (unica vez)
git add .
git commit -m "first commit"
git remote add origin https://github.com/fer123/appmobilequba (unica vez)
git push -u origin master
```

“**git init**” se hace una sola vez para crear un archivo git dentro de nuestro proyecto. En otras palabras, con **git init** se crea una carpeta oculta dentro de nuestro repositorio local para ir guardando la información de nuestro proyecto. Con “**git add .**” agregamos lo que vamos subiendo. Con “**git status**” vemos el estado de nuestro repositorio local, por ejemplo si hay cosas para hacer commit. Luego generamos un “**commit**”, que es como un punto de registro o checkpoint de nuestro proyecto. Luego vinculamos mediante “**git remote add origin http..**” nuestro archivo oculto local de github con el repositorio remoto. Este proceso es por unica vez, una vez que vinculamos nuestra carpeta local con la carpeta remota (mediante git “**remote add origin http..**”) ya queda vinculada para la proxima vez que retomemos el proyecto. Por último, para subir a internet: “**git push -u origin master**”.

Para retomar un proyecto existente hay que seguir los siguientes pasos:

```
Command prompt: cd + ruta
Git add .
git commit -m "first commit"
git push -u origin master
```

Para cambiar el origen de un repositorio remoto

```
git remote set-url origin https://github.com/cminnelli/Workshop3TP-FINAL.git
```

Para ver en donde el html del repositorio remoto

```
Git remote -v
```

Otros comandos de git

```
Git pull
```

(Cuando trabajamos con co-working y estamos subiendo cosas entre colaboradores).

Git hub pages

<http://albertoromeu.com/como-publicar-tu-aplicacion-web-en-github-con-gh-pages/>

Las dos instrucciones que necesitamos para crear la rama gh-pages son:

```
git checkout -b gh-pages
```

```
git push -u origin gh-pages
```

Al crear la rama gh-pages podemos acceder a lo que publica Github a través de esta URL,

<http://ejemplo.github.io/wiw/>

Para cambiar de rama utilizamos siempre el comando

```
git checkout
```

Si normalmente desarrollamos en la rama master y queremos que los cambios que hemos hecho se publiquen en la web que genera Github con gh-pages, debemos seguir el siguiente workflow:

- Trabajamos con normalidad en la rama master y en algún momento decidimos subir cambios a Github:

```
git push origin master
```

A continuación, cambiamos a la rama gh-pages:

```
git checkout gh-pages
```

Mezclamos los cambios de la rama master sobre la rama gh-pages que obviamente estará desactualizada:

```
git merge master
```

Subimos los cambios a la rama gh-pages en Github

```
git push origin gh-pages
```

Lo que acabamos de hacer es subir los cambios de la rama de desarrollo a la aplicación web en Github. A continuación, si queremos seguir desarrollando, volvemos a la rama master:

```
git checkout master
```

Y este workflow lo podemos repetir hasta el infinito.

¿Cómo despublicar tu aplicación web de Github?

Si en algún momento, queremos despublicar la aplicación web en Github simplemente debemos borrar la rama gh-pages:

```
git push origin --delete gh-pages
```

Por cierto, que también se puede utilizar gh-pages como hosting de una web y utilizar un nombre de dominio.

CSS3

SELECTORES

.cuadro (Class)
#cuadro(ID)

Dos etiquetas con un mismo formato:

.cuadro1, .cuadro2 {}
.cuadro1 ul, .cuadro2.ul{}

Aplicar a una clase dentro de otra clase:

.clase1 (espacio) .clase2 {}

Aplicar a los que las etiquetas con más de una clase:

.redondel.rojo {}
(Aplica a todas las clases redondeles rojos è en HTML Class="redondel rojo").

Fontsize

Body font size: 10 px

P{
Font-size: 1.5 em (1.5 por la fuente base que es 10px =15px).

Tipografía, si tiene más de una palabra va entre "" => "times new roman".

Font-family: "times new roman", arial; (si no la encuentra).

Nuevas fuentes:google fonts=> quick use, copiamos el link en el head y nos tira el font-family a insertar en el css

La clase se le puede poner a un párrafo tanto como a un div u otra etiqueta pero si yo armo el código: section.seccion1{ } => (aplica a las section que tenga la clase seccion1)
Section .seccion1{ } (aplica todas las clases DENTRO de un seccion)

Background-image

Si estoy en la carpeta CSS3 y necesito salir para ingresar a otra carpeta, comenzamos la ruta con “..” (dos puntos)

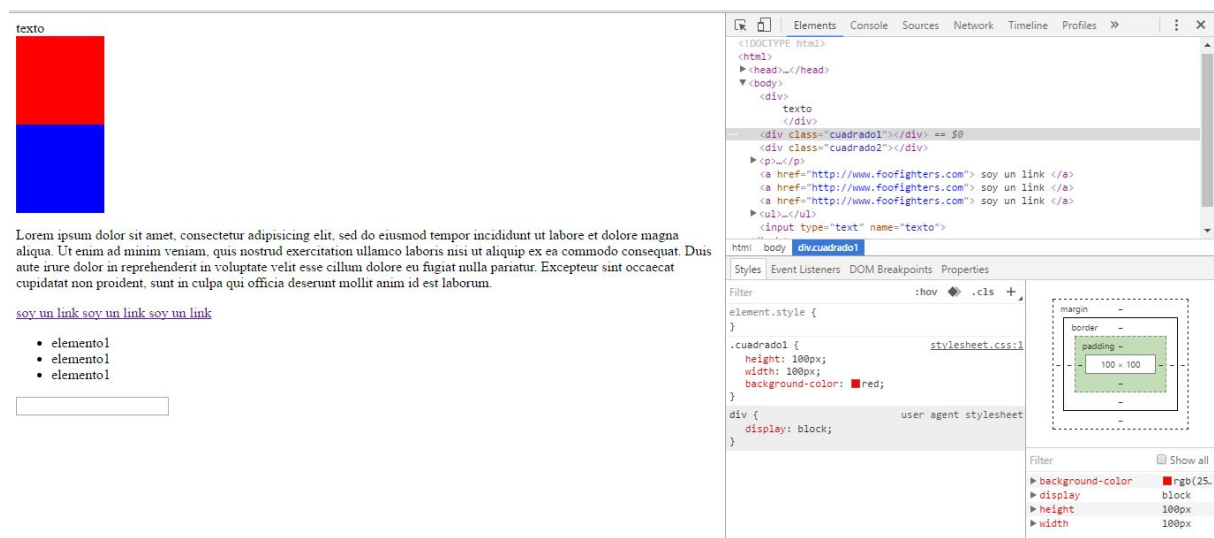
En vez de :

```
.seccion1{
    background-image:
url("http://2.bp.blogspot.com/_57ZnOGH3fTE/TVMZ_u_pNxl/AAAAAAAAA14/uLx2UImMnY
Q/s1600/www.fondosyfonditos.com.ar-HD-y-HQ-the-best-23423-32423-32+%252814%2529
.jpg");
background-repeat: no-repeat;
background-size: cover;
background-position: -50px 50px;
```

```
.seccion{
Background: url... no repeat; (hacerlo todo en una línea)
```

INSPECTOR DE ELEMENTOS es una herramienta de google chrome para inspeccionar la pagina, ahi vamos a poder ver los errores mediante “console” y hacer modificaciones con visualización en vivo.

Elementos de bloque: ocupan el 100% del ancho de la pantalla, aun si su contenido no llega a llenar todo. Ejemplo: DIV, ARTICLE, SECTION
Elementos de línea: por ejemplo los links, los input.
(lo podemos ver en google chrome, clickeando el “inspection” en cada uno de los elementos).



The screenshot shows the Google Chrome developer tools interface. On the left, the 'Elements' panel displays the DOM tree with a selected element: `<div class="cuadrado1"></div>`. Below the DOM tree, the 'Styles' panel shows the default user agent styles for a `div` element, including `display: block;`. On the right, a visual representation of the element is shown as a green square with a dashed border and padding, indicating its dimensions (100 x 100). The main content area of the browser shows a red square above a blue square, with text and links below them.

Atajo

Div.cuadrado1 + TAB = <div class="cuadrado1"></div>

Padding “calles externas del elemento”, “suman al espacio”

Padding:25 px => 25 px a los 4 lados.

Padding: x x x x ; arranca para arriba como la aguja del reloj.

padding : x x,

Dado que los div son en bloque, para cambiar su configuración por defecto y transformarlos en un bloque en línea => display: inline-block;

Ejemplo telefe logo:

En vez de hacer una clase para cada círculo y color, hago una clase para los redondeles, y una clase para la parte del color. De esta forma nos evitamos de repetir la forma del redondel en todas las clases. Se hace de la siguiente forma:

```
<div class="redondel azul"></div>
<div class="redondel verde"></div>
<div class="redondel rojo"></div>
```

Posicionamiento

Posición relativo: toma en cuenta su “hermano” anterior y de acuerdo a el se va a mover en la pantalla.

Posicionamiento absoluto: toma como valor de posicionamiento absoluto la página, SALVO que su padre sea relativo.

Posicion fixed: se queda fijo en el browser.

Z-index: pasar a otra capa y no depende de ningún objeto.

Cuando defino el posicionamiento de un objeto puede utilizar la variable top, bottom, left o right para desplazar.

Overflow: Hidden: Se utiliza para borrar todo lo que exceda de tamaño del padre. Si por ejemplo el div padre tiene un height de 300px y el párrafo excede los 300 px, el mismo desaparecerá. Es decir que para utilizar esta variable previamente se deberá definir el height del contenedor (padre).

Overflow: Scroll: Es el mismo caso que el overflow hidden pero en vez de que ocultarse cuando el hijo excede al padre, aparece un scroll en el contenedor padre. Para un scroll en los eje x se utiliza overflow-x:scroll, y para los del eje y: overflow-y:scroll.

overflow:auto: si tenemos elementos que flotan dentro de un contenedor, es necesario que el padre DIRECTO tenga la propiedad overflow:auto./// para que cuando hacemos float left o right, el padre tome la altura del hijo.

TIPS

El display:inline-block tiene que aplicarse al LI y Y el LIST-STYLE(-IMAGE) se aplica al UL.

Margin 0 auto: Con esta propiedad se centra el contenedor, pero solo se aplica a los elementos en bloque. Ejemplo:

```
.contenedor{  
    max-width: 900px;  
    margin: 0 auto;  
}
```

A cada elemento (div) que le pongamos la clase contenedor, aplicará el mismo seteo y se centrará el mismo.

Si ponemos un contenedor con ancho fijo en px cuando achicamos el navegador, los elementos se van a quedar estáticos, ya que los elementos internos por mas que esten el porcentaje el contenedor principal va a ser fijo. Para que sea variable el contenedor tiene que estar en porcentaje.

CSS3

Propiedad transition: 1s => se pueda aplicar a cualquier objeto.

Cuando se la aplicó a un elemento, por ejemplo a una <a> el efecto se aplica tanto cuando nos posamos, tanto cuando salimos. En cambio cuando lo aplicamos al hover, la transición al salir no se aplica, por este motivo se suele aplicar la transición al evento sin el hover.

```
-----  
.productos li{  
    padding: 15px;  
    box-sizing: border-box;
```

/* box-sizing: => border-box; box sizing es para que el padding no sume, sino que se incluya al tamaño original*/

First child, last child: para ir puntualmente al número de elemento.

```
.productos ul li:first-child{  
padding-left: 0px;  
}
```

```
.productos ul li:last-child{  
padding-left: 0px;  
}
```

ICONOS

<http://ionicons.com/#cdn>

Linkeo la página y lo establezco como un class con el nombre.

El icono es como un "font" por este motivo para cambiar el color lo trataremos igual que a una letra: color, font-size

POSICIONAMIENTO

Se creo un div padre y un div hijo. Se definió el div hijo con posición absoluta y para que el mismo se mueva relativamente al padre, se definió el padre como posición relativa. Si se establece un top y left del 50%, encuadra el cuadrado al centro, pero desde el punto 0,0. De esta forma no estaría quedando centrado. Aplicando la propiedad transform: translate(-50%,-50%) corre el 50% de su tamaño para que quede centrado.

Ejemplos:

```
.padre{  
    position: relative;  
    background:red;  
    height: 400px;  
    width: 100%;  
    margin: 20px auto;  
}  
  
.hijo{  
    background:blue;  
    position: absolute;  
    width: 200px;  
    height: 200px;  
    top:50%;  
    left: 50%;  
    transform: translate(-50%,-50%);  
}
```

MEDIA RESPONSIVE CON MEDIA QUERY

ETIQUETA VIEWPORT: PARA RESPONSIVE

En el head se escribe el siguiente código:

```
<meta name="viewport" content="width=device-width">
```

```
@media screen and (max-width: 800px) and (min-width:600px){
```

```
    nav{
```

```
        background-color: green;
```

```
    }
```

```
}
```

ANIMACIONES

```
@keyframes hola{
```

```
  from{
```

```
    background:blue;
```

```
  }
```

```
  to{
```

```
    background:red;
```

```
  }
```

```
}
```

```
.anim1{
```

```
  width:100px;
```

```
  height:100px;
```

```
  animation: hola 4s infinite;
```

```
  position: relative;
```

```
}
```

SASS

Para instalar SASS se necesita en el caso de windows, instalar ruby. Este programa traduce SASS a lenguaje css para que el HTML lo pueda interpretar.

Información de instalación <http://sass-lang.com/install>

Sintaxis: <http://sass-lang.com/guide>

Se crea carpeta sass y una css, en la carpeta css automáticamente se va a guardar la transformación de sass a css plano.

En el command prompt with ruby: CD + la ruta del proyecto:

```
CD + C:\Users\Carlos\Documents\CoderHouse\Clase 5 SASS
```

Para verificar que este correcto tipear DIR.

Para vincular el proyecto

(watch CARPETA ORIGEN - CARPETA DESTINO)

```
sass --watch sass/style.scss:css/style.css
```

(SASS va a ir a el archivo de sass y va a guardar la transformación a css constantemente mientras estemos trabajando.

El archivo de scss va a estar con plain text. Hay que cambiarlo a css

La consulta de ruby detecta errores en el archivo.

El HTML hay que vincularlo con el archivo de CSS no con el archivo de SASS, ya que html no sabe leer sass.

```
/ creando variable
```

```
$color: #FF0000;
```

```
body {
```

```
  // aplicando el valor de $color
```

```
  background-color: $color;
```

```
}
```

Una variable se podrá definir fuera o dentro de algún selector. Si se define fuera, dicha variable será global y podrá utilizarse en cualquier bloque, pero si se define dentro de un selector, la variable será local y únicamente se podrá utilizar en el selector que la contiene y en sus selectores anidados.

Una buena práctica común consiste en definir todas las variables globales al principio del fichero, para que puedan localizarse rápidamente.

El sass tiene que seguir la estructura del HTML

Para hacer un hover con SASS, dentro del selector se agrega una & que sirve para llamar al selector + hover:

```
a{
    text-decoration: none;
    background:white;
    border-radius: 5px;
    padding: 10px 30px;
    margin-top: 20px;

    &:hover{
        background:yellow;
    }
}
```

& es para hacer referencia al selector

El media query se puede hacer dentro de un selector si es poco lo que tengo que modificar, si el sitio necesita bastantes cambios para el responsive me conviene escribir la sintaxis a lo último.

EJEMPLO USO DE VARIABLES

(ver funciones de SASS en la página)

```
$color-boton:#678906;
```

```
$color-boton-hover:darken($color-boton, 20%); (O LIGHTEN)
```

```
.test{
  a{
    color:white;
    text-decoration: none;
    background-color: $color-boton;
    padding: 10px 30px;
    transition: 1s;
    &:hover{
      background:$color-boton-hover;
    }
  }
}
```

IMPORTAR ARCHIVOS

```
@import "reset";
```

```
@import "var" (variables);
```

```
@import: URL FONTS
```

```
@import: URL ION ICONS;
```

Todos los archivos tienen que estar en la carpeta "SASS"

EXTEND

Sirve para reutilizar propiedades y agregar propiedades de una clase.

```
.botton{
  font-family: arial;
  text-transform: uppercase;
  font-size: 10px;
  font-weight: bold;
  border-radius: 10px;
  padding: 10px 20px;
}

.principal{
  @extend .botton;
  background: red;
}
```

MIXINS

DEFINO VARIABLES

```
$base:blue  
$medium:silver  
$premium
```

```
@mixin crear_cajita($color,$color-oscuro){
```

```
    PEGAR LA CONFIGURACION DEL BOTON  
}
```

```
.medium{  
  @include{  
}
```

BOOTSTRAP

Es un framework hecho por twitter hecho para que hacer sitios responsive sea más fácil y también tiene una librería de objetos para dar estilo al sitio. Tanto bootstrap como jquery se baja desde su página. Hay dos formas de vincularlo: CDN o descargando los archivos a las carpetas del proyecto. Se recomienda el CDN en caso de proyectos pequeños. El riesgo de vincular a CDN (es decir de un servidor externo) es que se depende de su conexión.

Dos formas de centralizar nuestra página web. Clase container para centralizar los contenidos o container-fluid para no centralizar. (<div class="container-fluid">)

GRID SYSTEM

Bootstrap consta de un sistema de 12 columnas. Creando un div ROW y dentro creando divs con clases como por ejemplo "col-lg-12, se estaría creando una fila con una columna. Si dentro del DIV ROW, agrego 3 divs "col-lg-4, estaría creando una fila con tres columnas.

Agregar un span a un input: buscar el icono en Glyphicon, darle la class "form-control-feedback y al div darle la clase "has-feedback". El span debe ir entre medio del label y el input.

Para agregar un sub recuadro dentro de un input, hay que darle al div class input group y dentro un div con la class input-group-addon : <div class="input-group-addon">\${</div>. Debajo de esa sintaxis se coloca el input con class form-control.

BOTONES

tipos

```
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```

Tamaño

Btn-lg
Btn-sm
Btn-xs

IMAGENES

```



```

Responsive

```

```

TEXTOS

```
<p class="text-muted">...</p>
<p class="text-primary">...</p>
<p class="text-success">...</p>
<p class="text-info">...</p>
<p class="text-warning">...</p>
<p class="text-danger">...</p>
```

ALINEAMIENTO TEXTOS

```
<p class="text-left">Left aligned text.</p>
<p class="text-center">Center aligned text.</p>
<p class="text-right">Right aligned text.</p>
<p class="text-justify">Justified text.</p>
<p class="text-nowrap">No wrap text.</p>
```

SPAN

```
<p class="caret">No wrap text.</p>
```

ADDRESS

```
<address>  
  <strong>Twitter, Inc.</strong><br>  
  1355 Market Street, Suite 900<br>  
  San Francisco, CA 94103<br>  
  <abbr title="Phone">P:</abbr> (123) 456-7890  
</address>
```

BOOTSTRAP

Bootstrap ya tiene su reset, no hace falta incluirlo
Preferentemente usar los archivos en vez del CDN.

Si queremos modificar alguna clase no lo hacemos desde el archivo de bootstrap, sino que tenemos que pisar con nuestro css la clase (la misma clase o otra).

Breakpoint de media query que usar Bootstrap:

1770 / 992-1169 / 768-991 / 768-0

GRILLAS

Tratar de mantener los h1, p y demas fuera del div del row

Grid options

See how aspects of the Bootstrap grid system work across multiple devices with a handy table.

	Extra small devicesPhones ($<768\text{px}$)	Small devices Tablets ($\geq 768\text{px}$)	Medium devicesDesktops ($\geq 992\text{px}$)	Large devices Desktops ($\geq 1200\text{px}$)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
# of columns	12			

Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

4

Form Group por cada input

El form group te crea un row automaticamente apesar de no haberla creado

Hidden-xs :clase para eliminar un elemento para xs

Col xs 12 : si digo col xs 12 todo lo demas para arriba (md/lg) lo va a tomar con una grilla de 12 columnas.

Accesibilidad

Un solo h1 por página

Si vamos a repetir headings, encapsulados en articles

Propiedad alt: describir la imagen para el posicionamiento.

SUBIR PÁGINAS

1)Dominio ...www.charlie.com

2)Servidor

Se relacionan mediante el DNS

Bajar Filezilla: Puerta hacia mi servidor

GIT HUB PAGES

Una vez que subo nuestro proyecto al repositorio:

Master es el proyecto principal

Branch es una rama de nuestro proyecto principal. Abrimos una rama paralela con los archivos del proyecto master y hacemos modificaciones en base a ese.

Git branch

Crear branch git checkout -b "gh-pages"

Para cambiar al branch es sin el -b

Cambiar a git hub pages: git branch

Git checkout master para volver a trabajar sobre el master
