



Curso de Front-End

Clase 06



Agenda de la clase



Agenda

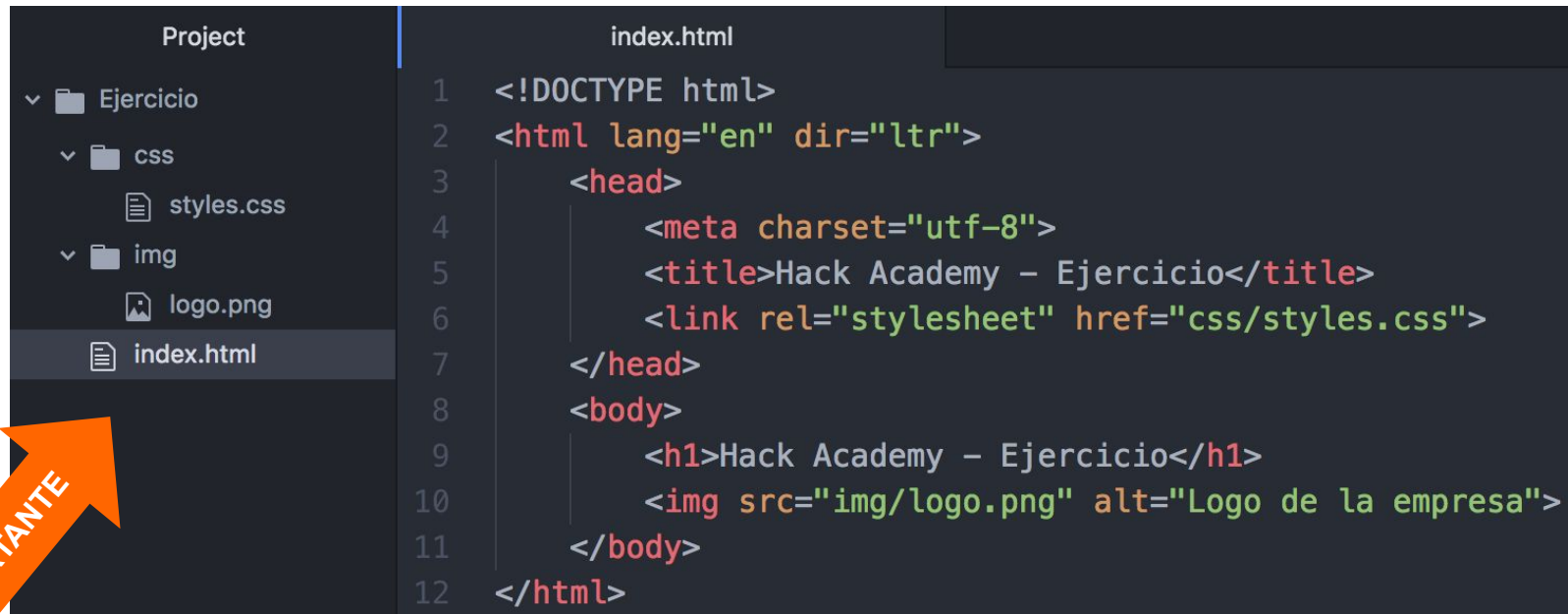
- Repaso.
- ¿Cómo funciona Internet? (en 3 slides).
- Formularios en HTML.
- Ejercicios.



Repaso



Estructura básica de un proyecto



En general, todos los proyectos (ejercicios) con los que trabajaremos tendrán esta estructura básica. Los nombres de los archivos y carpetas son arbitrarios, pero es una **convención** llamarlos de esta forma y en **minúscula**. Para abrir un proyecto en VSC, ir al menú **File > Open Folder** en Windows o **File > Open** en Mac.



Frameworks CSS – Bootstrap

Existen varios frameworks de CSS.

El más popular y el que vamos a usar en el curso es **Bootstrap**:

- Creado por empleados de Twitter en **2011**.
- Es *open-source*.
- Es un framework de HTML, CSS y JavaScript.
- Sólo para la parte de CSS contiene aprox. **11.000** líneas de código!
- Documentación: <http://getbootstrap.com>.



Bootstrap – Resumen de Ventajas y Desventajas

Ventajas	Desventajas
<ul style="list-style-type: none">● Grid System.● CSS reset.● Responsive design.● Facilidad de uso.● Mobile first.● Comunidad de usuarios.● Documentación.● Estilos y componentes.	<ul style="list-style-type: none">● Puede ser muy “pesado” para un proyecto chico.● Sitios web creados usando Bootstrap pueden quedar demasiado parecidos entre sí y por lo tanto “no destacarse”.



Bootstrap – Instalación (usando CSS remoto)

```
<head>

  <meta charset="utf-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  ...

  <link

    href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"

    rel="stylesheet"

    integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6D0itfPri4tjfhXaWutUpFmBp4vmVor"

    crossorigin="anonymous"

  />

  ...

</head>
```

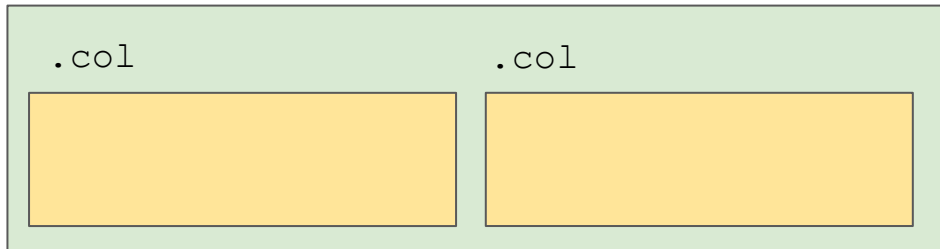



Bootstrap – Grid System (1/3)

El [Grid System de Bootstrap](#) es un sistema de filas y columnas que permite armar el layout de nuestra página de una **forma muy sencilla**.

Las filas y columnas no son más que `divs` con ciertos estilos CSS.

`.row`



Importante: Las clases `.row` y `.col` las provee Bootstrap. No las tenemos que crear nosotros.

```
<div class="row">
  <div class="col">
    ...
  </div>
  <div class="col">
    ...
  </div>
</div>
```

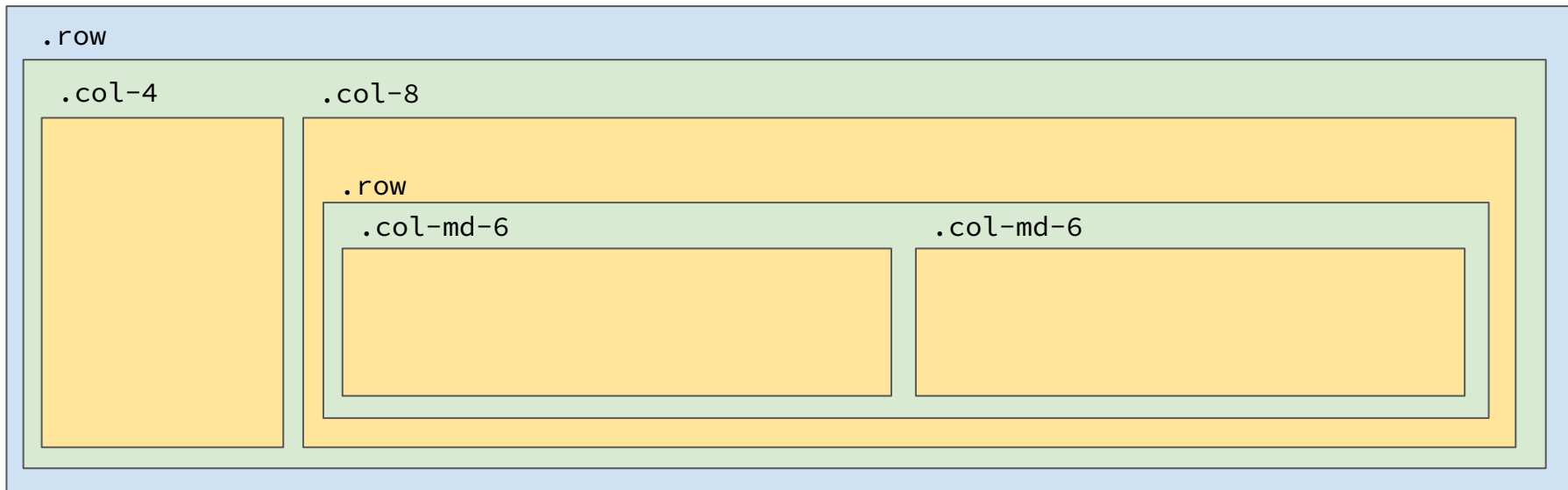
Notar que para lograr este layout no fue necesario escribir ni una sola línea de CSS.



Bootstrap – Grid System (2/3)

Las filas pueden colocarse *casi* en cualquier lugar, incluso dentro de otras columnas. Esto se conoce como anidación o [nesting](#). Además, si se especifica el ancho de las columnas, la suma de los anchos no debe superar 12.

`.container`





Bootstrap – Grid System (3/3) – Reglas

```
<div class="row">
  <div class="col">
    ...
  </div>
  <div class="col">
    ...
  </div>
</div>
```

- Toda columna debe ir inmediatamente adentro de una fila. Adentro de una fila, sólo se pueden colocar columnas.
- **Nuestro contenido** se agrega **dentro de las columnas** (jamás dentro de las filas).
- Dentro de las columnas incluso se puede agregar otros `<div>`.

IMPORTANTE



*“Cuando consulten una documentación,
verifiquen qué versión están consultando”*

Documentación de Bootstrap v5: <https://getbootstrap.com/docs>.



¿Cómo funciona Internet?

(en 3 slides)

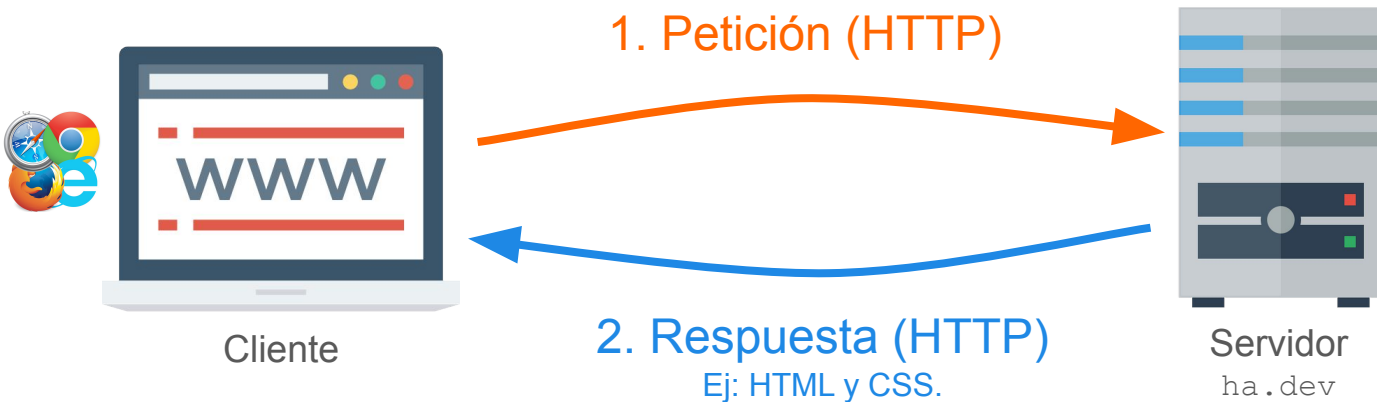
Arq. Cliente-Servidor (1/2)



Cuando un usuario escribe una URL en su navegador y presiona `Enter`, lo que hace es realizarle una **petición** (HTTP) a un servidor web.

Básicamente el usuario le dice al servidor: *"Dame lo que haya en esta URL: `https://ha.dev`".*

Arq. Cliente-Servidor (2/2)



Al recibir la petición, una aplicación en el servidor la analiza, la procesa y responde al cliente con una **respuesta** que suele ser en formato HTML (pero puede ser CSS, JavaScript, JSON, una imagen o texto plano). Los posibles formatos se pueden consultar [aquí](#).

El servidor jamás retorna código PHP, Java, Ruby, Python, C# u otro código de Back-End. Recordar que el navegador sólo "entiende" cosas como HTML, CSS, JavaScript e imágenes.



Clientes, Navegadores y Servidores

Los **clientes** son desktops, móviles, tablets, etc., que acceden a Internet a través de navegadores.

Los **navegadores** son programas que despliegan contenido (principalmente HTML) de una forma amigable para el usuario.

Existe una gran variedad de navegadores y cada uno tiene sus particularidades.

Dos navegadores diferentes pueden mostrar la misma página web de manera distinta.

Los **servidores** son computadoras que contienen la información (archivos HTML, archivos CSS, imágenes, etc) de un sitio web.

Suelen ser equipos poderosos que están prendidos y conectados a Internet las 24 horas.



Formularios en HTML



¿Te gustaría estudiar en Hack Academy?

Completa los siguientes datos para ponernos en contacto.

Nombre

Ej: María

Apellido

Ej: Pérez

E-mail

Ej: maria.perez@gmail.com

País

Uruguay (+598)



Teléfono Móvil (Sin incluir código de país)

Ej: 99 111-222

Curso que te interesa

Seleccionar...



Al completar este formulario aceptas recibir novedades sobre nuestros cursos.

Enviar →



*“Los **formularios** son de los componentes más **importantes** de un sitio web”*

A través de un formulario, se captan nuevos usuarios, se captan interesados en nuestros productos (*leads*), se generan ventas (formularios de *checkout*), etc. Por eso, es sumamente importante dedicar el tiempo suficiente para **optimizarlos**, es decir, hay trabajar para que su **conversion rate** sea el más alto posible.



Formularios HTML

Un formulario está determinado por el elemento `<form>`.

```
<form>
```

```
    . . .
```

```
    . . .
```

```
    . . .
```

```
</form>
```

Documentación: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms>



Formularios HTML – input

Para agregar un **campo de texto** se utiliza el elemento `<input>`.

```
<form>
  ...
  <input type="text">
  ...
</form>
```

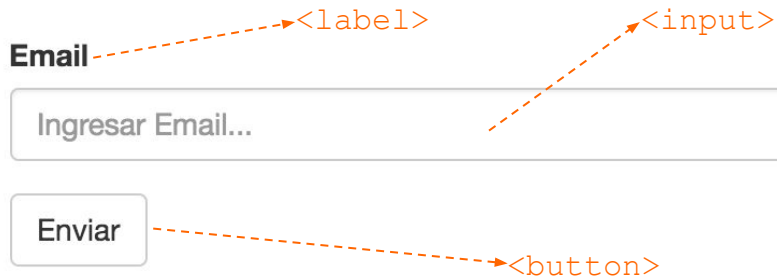


Formularios HTML – label, input & button

En general, todos los campos van acompañados de una **etiqueta** `<label>`.

Además, todo formulario suele tener al menos un **botón** `<button>`.

```
<form>
  <label for="email">Email:</label>
  <input id="email" type="text" name="email" placeholder="Ingresar Email...">
  <button type="submit" name="button">Enviar</button>
</form>
```





Formularios HTML – input

12:07 50%

ha.dev/aplicar

HACK ACADEMY

País

Uruguay (+598)

Teléfono Móvil (Sin incluir código de país)

99 111-222

Curso que te interesa

Seleccionar...

Al completar este formulario aceptas recibir novedades sobre nuestros cursos.

Enviar

Los elementos `<input>` tienen un atributo `type` el cual permite definir el tipo del campo de texto. Ejemplos:


Tipo	Para
text	Textos cualesquiera.
number	Números.
email	Correos electrónicos.
date	Fechas.
tel	Números de teléfono.
radio	Campos de tipo "radio" (sólo se puede elegir una opción).
checkbox	Para campos de selección múltiple.



Formularios HTML – select (combo-box)

```
<label for="pais">País</label>
<select id="pais" name="pais">
  <option value="AR">Argentina</option>
  <option value="CO">Colombia</option>
  <option value="UY">Uruguay</option>
</select>
```

País



Formularios HTML – Radio buttons (input)

```
<fieldset class="mb-3">
  <legend>¿Sabes HTML?</legend>
  <div class="form-check">
    <input id="sabe-html-si" class="form-check-input" type="radio" name="sabeHtml" value="S">
    <label for="sabe-html-si" class="form-check-label">Sí</label>
  </div>
  <div class="form-check">
    <input id="sabe-html-no" class="form-check-input" type="radio" name="sabeHtml" value="N">
    <label for="sabe-html-no" class="form-check-label">No</label>
  </div>
</fieldset>
```

¿Sabes HTML?

☒ Sí

☐ No



Formularios HTML – textarea

```
<label for="comentario">Déjanos un comentario</label>  
<textarea id="comentario" name="comentario" rows="3"></textarea>
```

Déjanos un comentario



Formularios HTML – Envío de Email

Para poder enviar los datos de un formulario HTML a una dirección de email (sin tener conocimientos de Back-End), pueden usar un servicio gratuito llamado **Formspree** (<https://formspree.io/>). 🖐 Requiere que se hagan una cuenta previamente.

Importante: nunca envíen datos confidenciales de esta forma.

```
<form method="POST" action="https://formspree.io/m321g2f">  
  
  . . .  
  
  . . .  
  
</form>
```

🖐 El número `m321g2f` es un ejemplo de un identificador generado por Formspree. Cada formulario debe tener su propio identificador.



Formularios y Bootstrap

Formularios y Bootstrap



Correo electrónico

Contraseña

Documentación: [aquí](#).

Notar que no fue necesario escribir ni una sola línea de CSS.

```
<form>

  <div class="mb-3">

    <label for="email">Correo electrónico</label>

    <input type="email" class="form-control" id="email">

  </div>

  <div class="mb-3">

    <label for="password">Contraseña</label>

    <input type="password" class="form-control" id="password">

  </div>

  <button type="submit" class="btn btn-success">Iniciar sesión</button>

</form>
```

Las clases `.mb-3` y `.form-control` las provee Bootstrap. No las tenemos que crear nosotros.

Ejercicio

⚠ Este es un ejercicio extra. Por lo tanto, en general, no se pretende que el mismo se resuelva durante el dictado de la clase. Esto puede depender de la “velocidad” de cada grupo.


🤔 Recordar que el curso tiene varios de estos ejercicios que sirven para que los alumnos sigan practicando por su cuenta.

👉 En caso de no haber terminado el **ejercicio 3 de la clase anterior**, es preferible que lo terminen antes de comenzar un ejercicio nuevo.

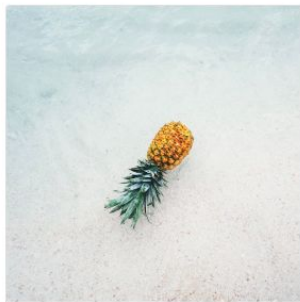


Ejercicio 1

1. Crear una carpeta en el Escritorio (o donde prefieran) con el nombre `Clase06_Ejercicio1`.
2. Abrir dicha carpeta en **Visual Studio Code**.
Esto se puede hacer yendo al menú: `File > Open Folder` en Windows o `File > Open` en Mac.
3. Desde VSC, crear un archivo llamado `index.html` dentro de la carpeta.
4. Desde VSC, crear una carpeta `css` y dentro de la misma el archivo `styles.css`.
5. Instalar Bootstrap usando el “Método 2” visto anteriormente.
6. Crear el HTML y CSS para lograr un resultado similar al siguiente diagrama.

Buscar vuelos 

 Buscar



Playa del Carmen

Lorem ipsum dolor sit amet consectetur adipiscing elit. Quae, totam. Nobis, tempora. Corrupti, hic labore!

 Comprar



San Andrés

Lorem ipsum dolor sit amet consectetur adipiscing elit. Quae, totam. Nobis, tempora. Corrupti, hic labore!

 Comprar



Búzios

Lorem ipsum dolor sit amet consectetur adipiscing elit. Quae, totam. Nobis, tempora. Corrupti, hic labore!

 Comprar

Buscar vuelos ✈

Ingresar una ciudad...

Q Buscar



Playa del Carmen

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae, totam. Nobis, tempora. Corrupti, hic labore!

🛒 Comprar



San Andrés

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae, totam. Nobis, tempora. Corrupti, hic labore!

🛒 Comprar



Búzios

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae, totam. Nobis, tempora. Corrupti, hic labore!

🛒 Comprar



Ejercicio 1 (cont) – Comentarios

- La idea es aprovechar [Bootstrap](#) al máximo pero también van a tener que escribir sus propios estilos CSS.
- Van a tener que **investigar mucho la documentación** de Bootstrap. Tienen que descubrir qué componentes les pueden ser útiles.
- La imágenes se compartirán en un archivo ZIP por Slack/Teams.
- El color amarillo de los botones debe ser #FFFF00 y #FFD600 al hacer hover.
- Los íconos (avión, lupa y carrito de compras) se deben generar usando [Google Material Fonts](#) ó [Font Awesome](#).