



Curso de Front-End

Clase 03



Agenda de la clase



Agenda

- Repaso.
- Continuación con selectores CSS:
 - Agrupación.
 - Anidación.
 - Jerarquía.
- Otra forma de usar CSS: desde un archivo externo.
- CSS Box Model.
- Ejercicios.



Repaso

Tips generales

¡Practiquen mucho!



¿Cómo se aprende a tocar la guitarra?

Recuerden que no mandamos deberes, exámenes, pruebas, entregas, etc.

Pero eso no quiere decir que no practiquen por su cuenta. Lo ideal es conseguirse un **proyecto propio** (un sitio web para ustedes o para alguien más). Es inevitable notar la diferencia entre los que practican y los que no.



Tips generales

Por las dudas lo repetimos...

¡Practiquen mucho!



Sugerimos que practiquen en sus casas aprox. **6 horas** semanales.



Sintaxis de CSS

CSS consiste de una serie de reglas de estilo. Cada regla está conformada por un **selector** y declaraciones de pares “**propiedad-valor**” (o clave-valor).

```
selector {  
    property: value;  
}
```

Sintaxis:

El selector encierra las propiedades entre llaves { y }.

Los pares de propiedad-valor se separan por : y terminan con un ;.

```
body {  
    color: blue;  
    background-color: grey;  
}
```

En este ejemplo el selector es el tag `body` y se le asignaron 2 propiedades CSS.

La primera establece el color del texto en el `body`.

La segunda establece el color de fondo del `body`.



CSS – Varios valores en la misma propiedad

A veces, se le pueden asignar varios valores a la misma propiedad, separados por un espacio. Por ejemplo, esto sucede con la propiedad `font` (ver [docs](#)).

```
body {  
  color: #4286f4; /* Azul */  
  background-color: grey;  
  font: italic 1.4rem sans-serif;  
}
```

Son equivalentes.

```
body {  
  color: #4286f4; /* Azul */  
  background-color: grey;  
  font-style: italic;  
  font-size: 1.4rem;  
  font-family: sans-serif;  
}
```




CSS dentro de HTML

Se vieron **dos formas** de *embeber* CSS dentro de HTML.

```
<head>
  <style>
    p {
      color: #70AA31;
    }
  </style>
</head>
```

```
<body>

  <p style="color:#70AA31;">
    Texto del párrafo.
  </p>

</body>
```

¿Cuál de estas dos formas es preferible? ¿Hay algún caso en que sea necesario usar alguna de ellas?



Tipos de selectores CSS

```
body {  
  font-size: 1.4rem;  
}  
#titulo-principal {  
  font-family: Arial, sans-serif;  
}  
.importante {  
  color: #E52604; /* Rojo */  
}  
h2 {  
  font-family: Arial;  
}  
a:hover {  
  color: #33AA88;  
  text-decoration: underline;  
}
```

CSS permite aplicar estilos a:

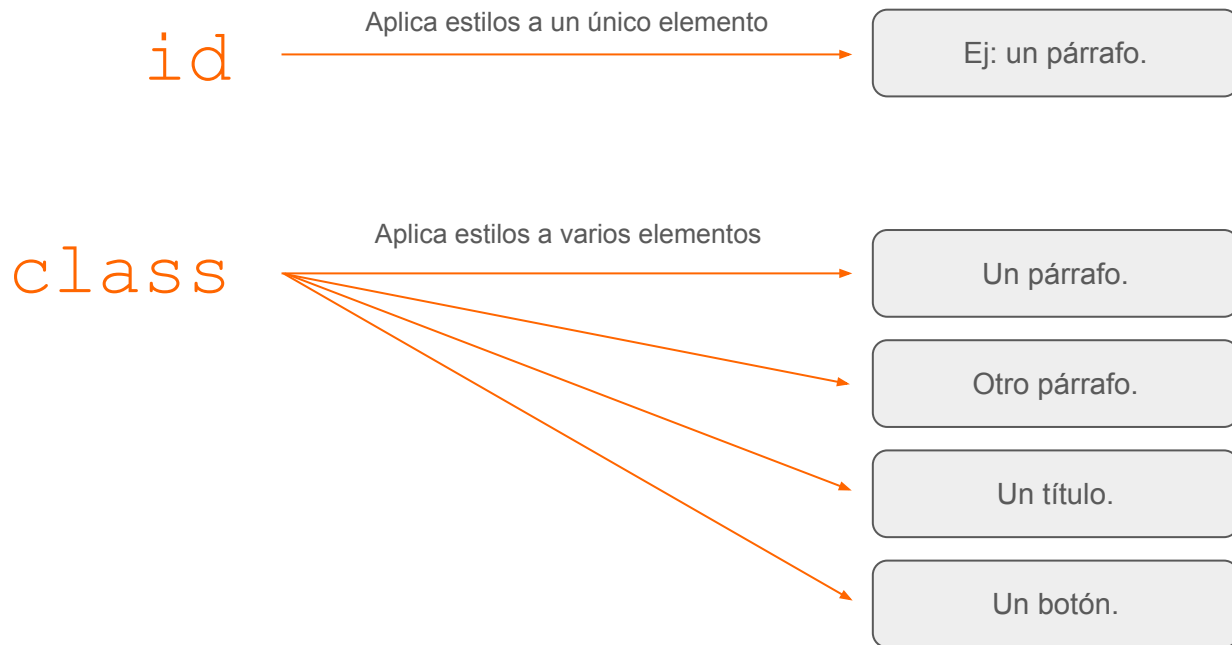
- Todos los elementos de cierto tipo.
- Un elemento que tenga cierto `id`.
- Todos elementos que tengan cierto `class`.

También se puede aplicar estilos a un elemento según su estado. Esto se hace usando ***pseudo classes***.

Nota: existen otras formas de especificar estilos, pero estas son las más importantes.

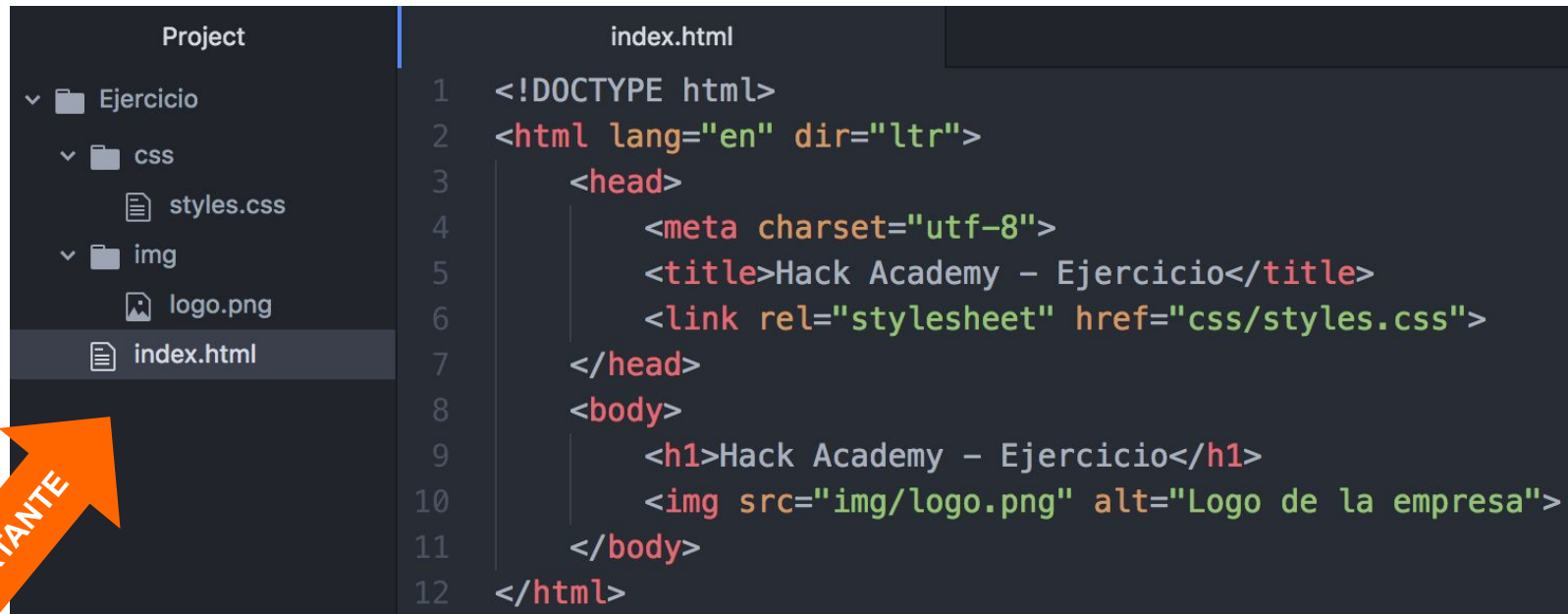


Tipos de selectores: `id` vs `class`





Estructura básica de un proyecto



IMPORTANTE

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.



CSS – Selectores anidados

Selectores anidados



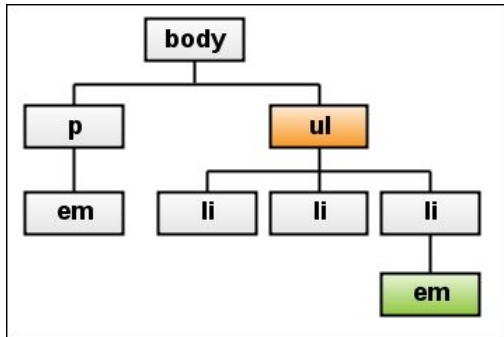
Es posible anidar selectores de esta manera:

```
ul em {  
    color: blue;  
}
```

Notar el espacio “ ” entre ambos selectores.

Esto selecciona todos los elementos `em` que sean descendientes de un `ul`.

```
<body>  
    <ul>  
        <li><em>Lorem</em> ipsum</li>  
    </ul>  
</body>
```





CSS – Agrupación de selectores



Agrupación de selectores

```
p, #seccion-principal, .descripcion {  
    font-size: 1.2rem;  
}
```

Notar la *coma* entre los selectores.

Equivalente a:

```
p {  
    font-size: 1.2rem;  
}  
#seccion-principal {  
    font-size: 1.2rem;  
}  
.descripcion {  
    font-size: 1.2rem;  
}
```




Jerarquía de selectores

(continuación de clase de 02)



Jerarquía de selectores CSS (continuación de clase de 02)

Reglas:

- Los `id` son más específicos que los `class` y
- `class` es más específico que `element`.
- Cuanto más “largo” el selector, más específico es.
- Si dos reglas son igual de específicas, la última gana.

```
.introduccion h2 {  
    color: blue;  
}  
h2 {  
    color: red;  
}
```

Gana la **primera regla** porque es la más específica.



Otra forma de usar CSS:
desde un archivo externo.

CSS en un archivo externo (1/2)



index.html

```
<h2>
  Hack Academy
</h2>
```

```
<link rel="stylesheet"
      href="styles.css">
```

styles.css

```
h2 {
  font-size: 3rem;
}
```





CSS en un archivo externo (2/2)

En el archivo HTML se coloca el siguiente código para enlazar a un archivo CSS externo:

```
<head>
  ...
  <link rel="stylesheet" href="css/styles.css">
  ...
</head>
```

En el archivo `styles.css` (dentro de la carpeta `css`) se coloca el código CSS (tal como lo hemos visto hasta ahora):

```
h2 {
  font-family: Arial;
  font-size: 3rem;
}
```



CSS Box Model

CSS Box Model

Un h1, un p, un div, un span, etc., todos son tratados como *boxes*.



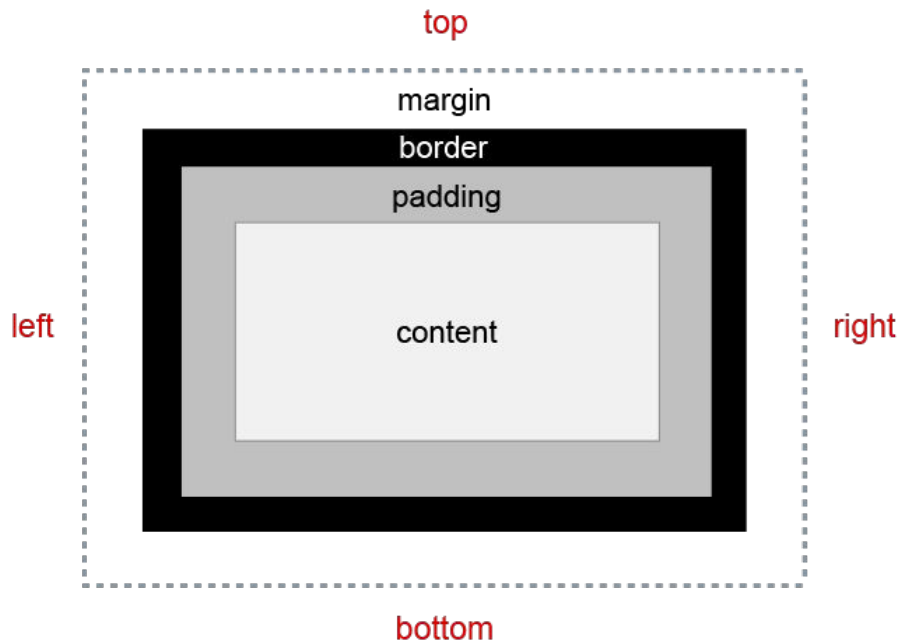
Todos los elementos HTML son tratados como *boxes* (cajas) por el navegador. Pueden ser `inline` o `block` boxes.

Un box consiste en:

- Content.
- Padding.
- Border.
- Margin.

Usar Chrome *Developer Tools* para inspeccionar *boxes*.

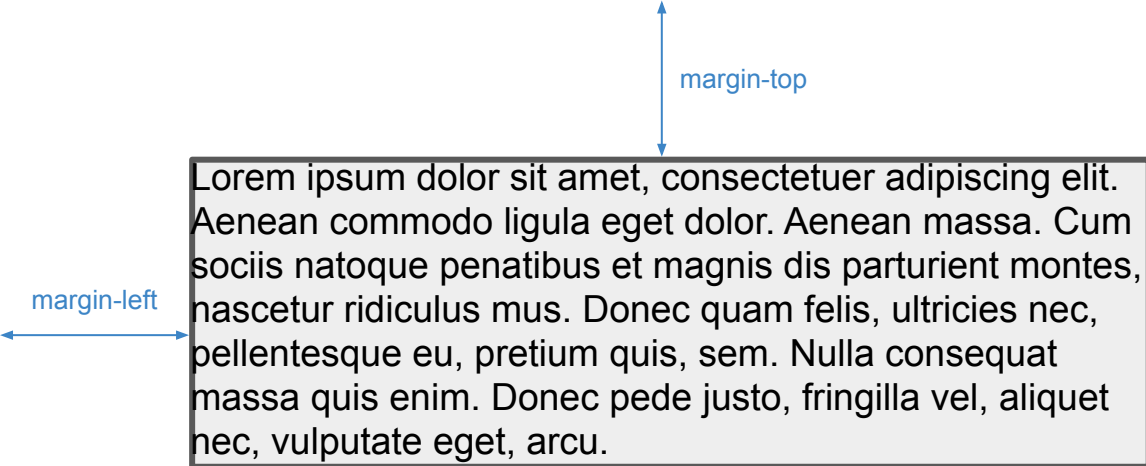
👉 Ver [video explicativo](#).





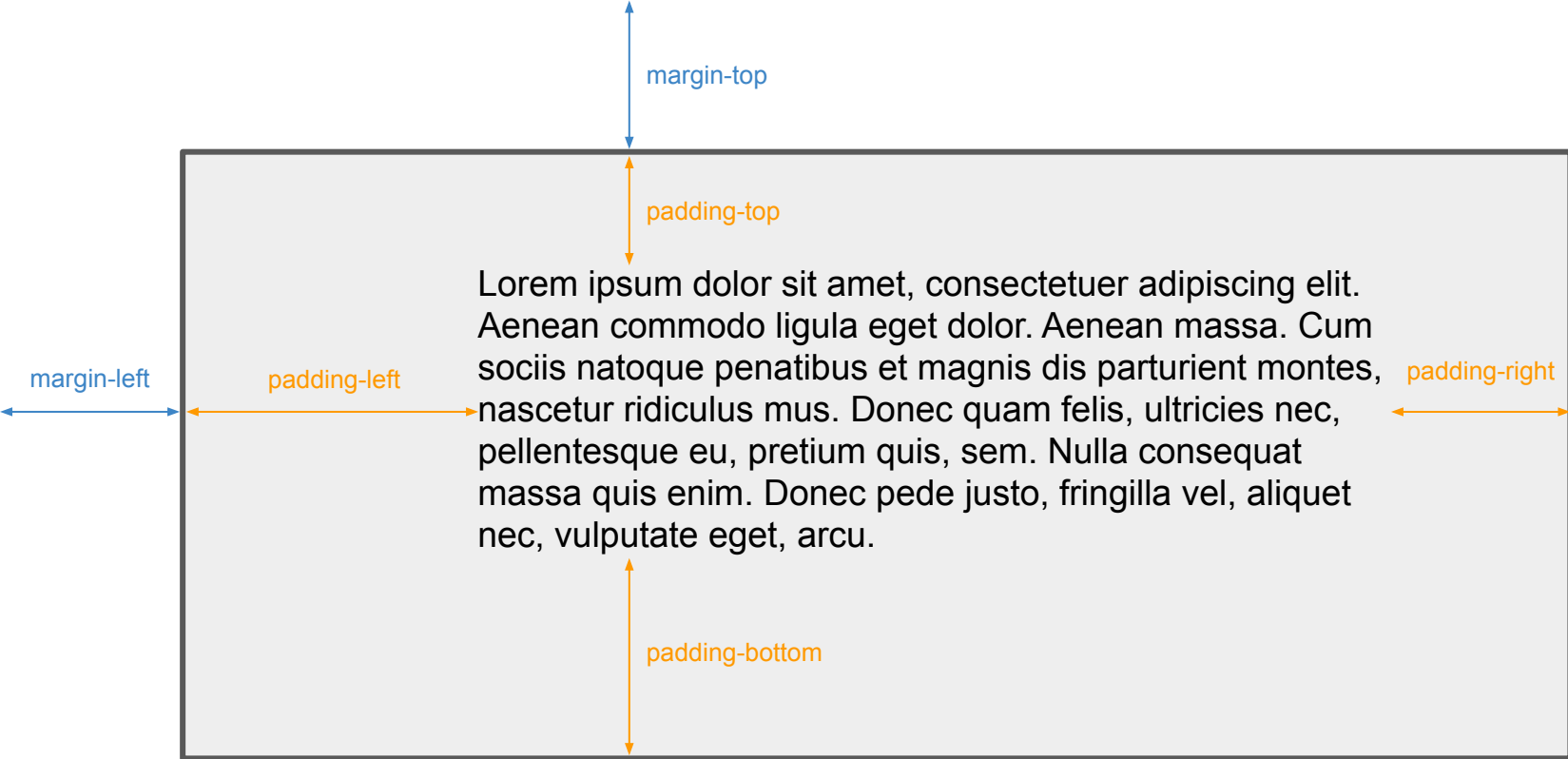
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

Una caja sin margin ni padding.



Se agregó `margin`.

Nota: si llegan a tener algún problema con los márgenes de dos elementos adyacentes, es decir, sobre márgenes que “chocan” entre sí, leer sobre [Collapsing Margins](#).



Se agregó padding.

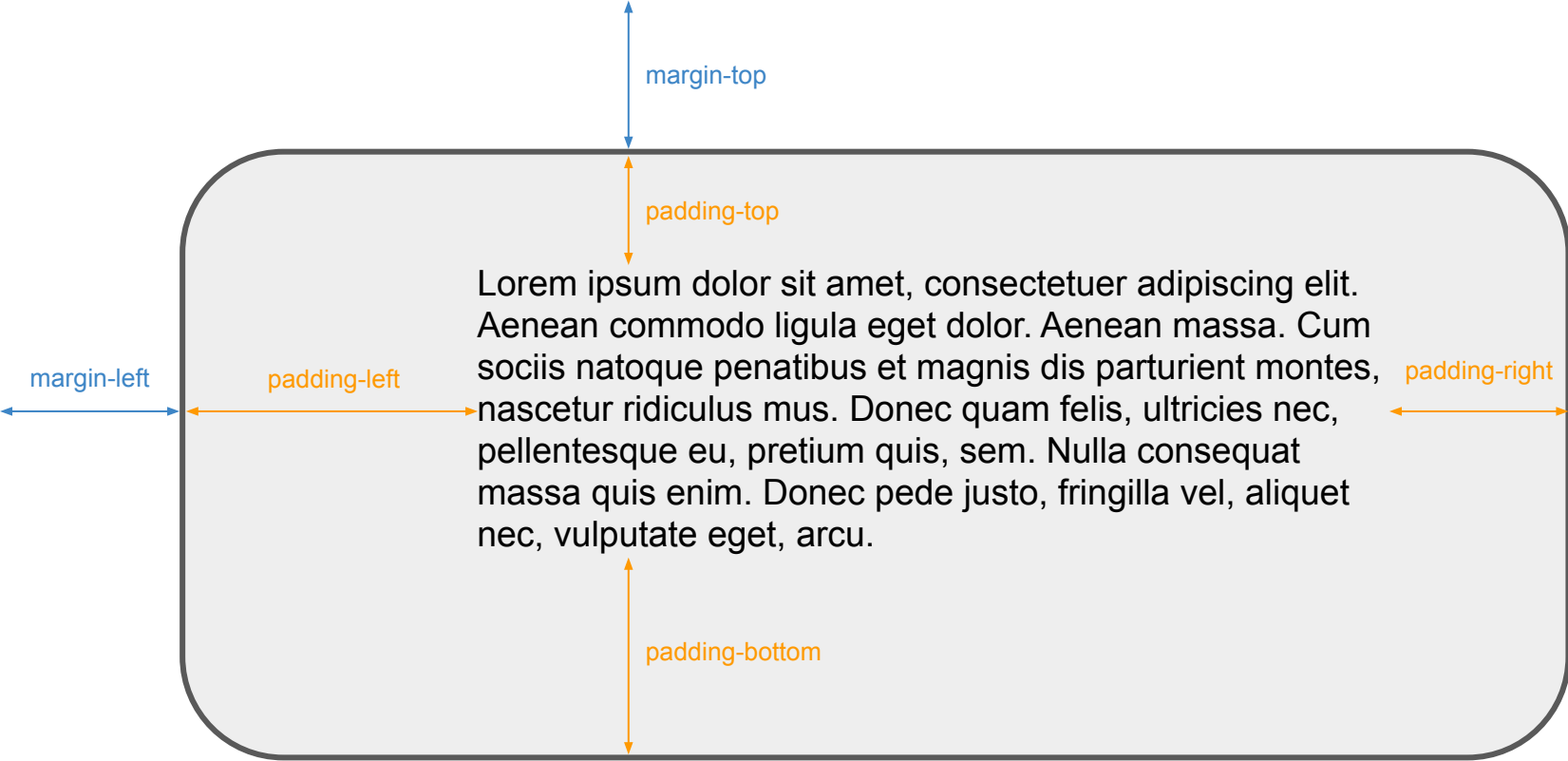


CSS Box Model – Border

```
#bloque1 {  
    border: 1px solid red;  
}  
  
#bloque2 {  
    border: 5px dashed #CCC;  
}  
  
#bloque3 {  
    border-top: 12px dotted #23c129;  
}  
  
#bloque4 {  
    -webkit-border-radius: 5px; /* Para compatibilidad con Chrome's viejos */  
    -moz-border-radius: 5px; /* Para compatibilidad con Firefox's viejos */  
    border-radius: 5px; /* Funciona en browsers modernos */  
}
```



PRUÉBENLO



Se agregó `border-radius`.



CSS Box Model – Margin

```
#bloque1 {  
    margin: 20px; /* 20px de margen hacia todos lados. */  
}  
  
#bloque2 {  
    margin: 5px 2px 1px 5px;  
    /* Márgenes aplicados en sentido horario (top, right, bottom, left). */  
}  
  
#bloque3 {  
    margin-top: 12px; /* Margen específico para el lado superior. */  
}  
  
#bloque4 {  
    margin: auto; /* Margen ocupará lo máximo posible, pero sólo hacia los costados. */  
}
```



CSS Box Model – Padding

```
#bloque1 {  
    padding: 20px; /* 20px de padding hacia todos lados. */  
}  
  
#bloque2 {  
    padding: 5px 2px 1px 5px;  
    /* Paddings aplicados en sentido horario (top, right, bottom, left). */  
}  
  
#bloque3 {  
    padding-top: 12px; /* Padding específico para el lado superior. */  
}  
  
#bloque4 {  
    padding: auto; /* No funciona. */  
}
```



Links útiles



Links útiles

Aconsejamos mucho que puedan tomarse un tiempo para leer sobre estos temas:

- [id vs. class.](#)
- [Box Sizing.](#) 🖱️ Particularmente la opción: `box-sizing: border-box;`
- [Collapsing Margins.](#)



Ejercicios

Nota sobre los ejercicios:

El curso contiene **ejercicios de más** (muchos de los cuales están marcados con la etiqueta “Extra”).

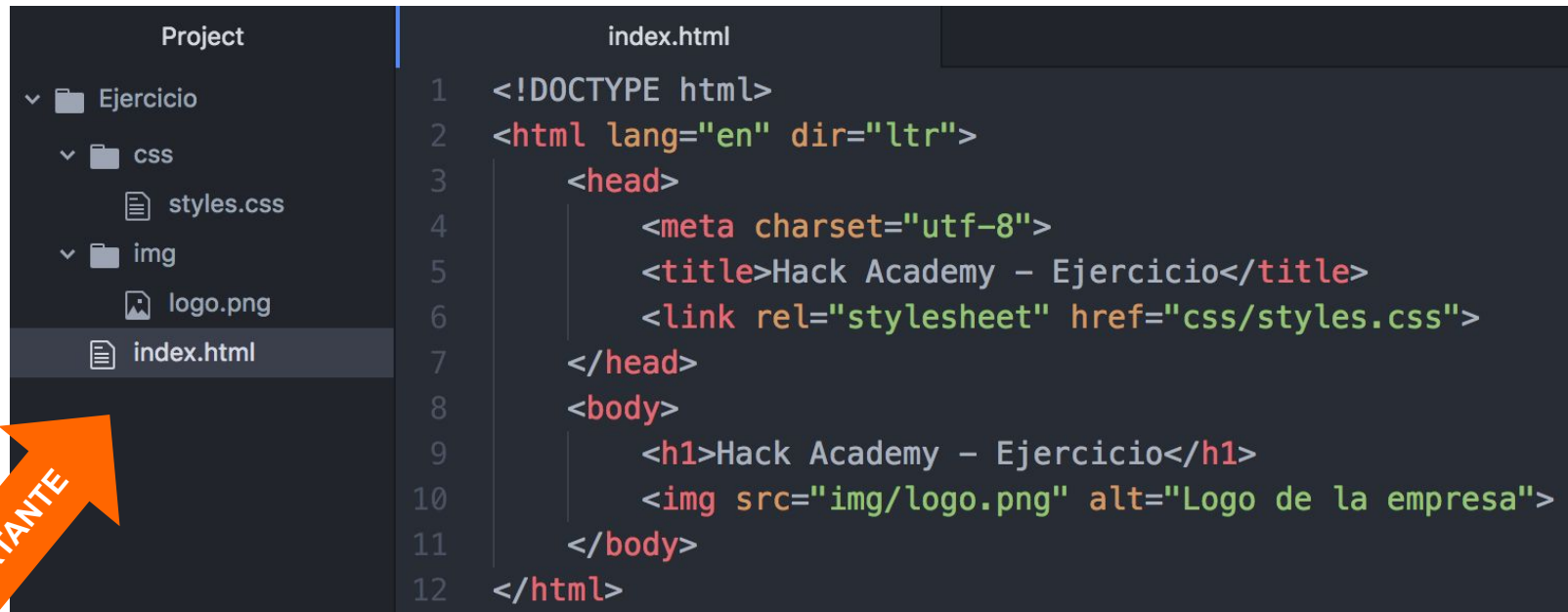
No se preocupen si no logran terminar todos los ejercicios de una clase; en general el tiempo no alcanzará.

Suele haber alumnos que avanzan más rápido y otros más lento, es totalmente normal, sobre todo al principio. La idea es que hayan **ejercicios para todos**.





Estructura básica de un proyecto



IMPORTANTE

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**.

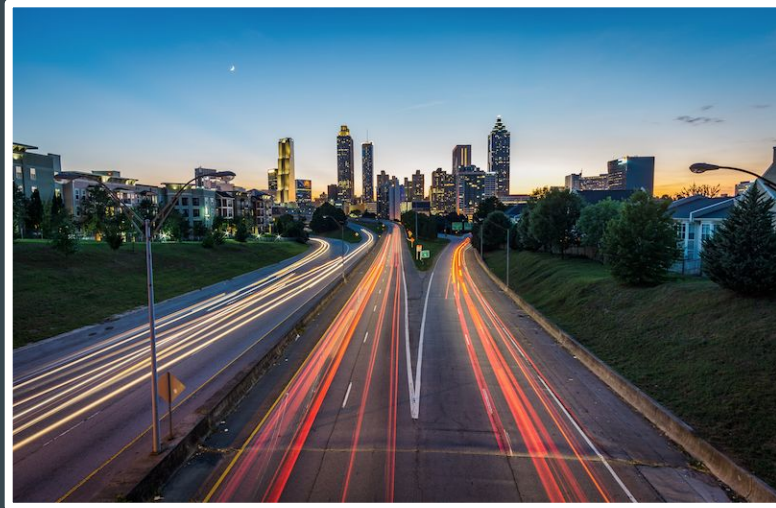
Para abrir un proyecto en VSC, ir al menú **File > Open Folder** en Windows o **File > Open** en Mac.



Ejercicio 1

1. Crear una carpeta en el Escritorio (o donde prefieran) con el nombre `Clase03_Ejercicio_1`.
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. Insertar en la página una imagen tal como muestra el diagrama a continuación.
 - a. La imagen debe estar centrada horizontalmente en la página.
 - b. La imagen debe estar separada 50px del borde superior de la página.
 - c. La imagen debe tener un borde de color blanco, sólido y de 5px de ancho.
 - d. El ancho de la imagen no debe superar los 600px.

Ejercicio 1 (cont)



Jackson Street Bridge
Atlanta, United States

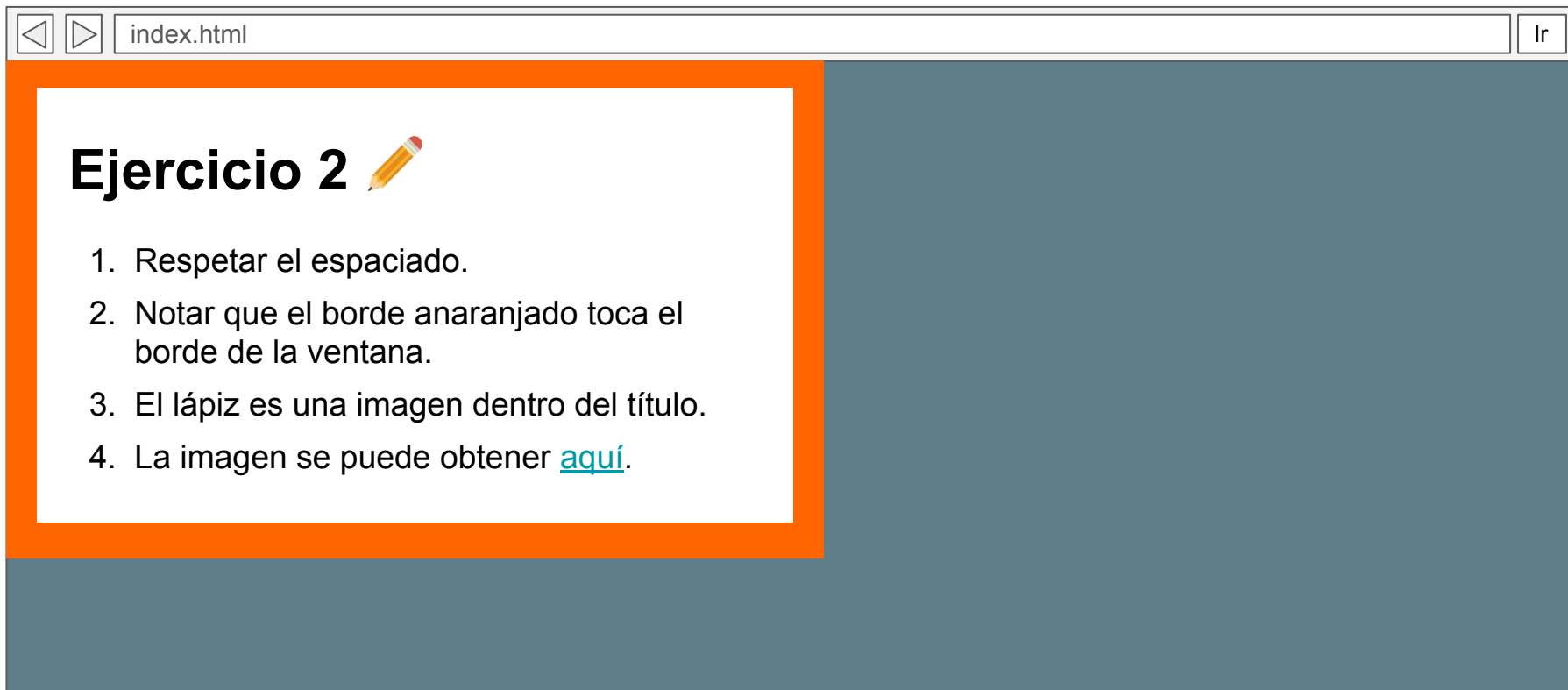


Ejercicio 2

1. Crear una carpeta en el Escritorio (o donde prefieran) con el nombre `Clase03_Ejercicio_2`.
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. Crear el HTML y CSS para lograr un resultado similar al siguiente diagrama:

Ejercicio 2 (cont)

Sugerencia: Empezar por lo más importante, lo estructural. Ej: dejar la imagen del lápiz para el final.





Ejercicio Extra

Estos ejercicios no se suelen resolver durante la clase sino que están planteados para que los alumnos puedan hacerlos para fuera de hora.

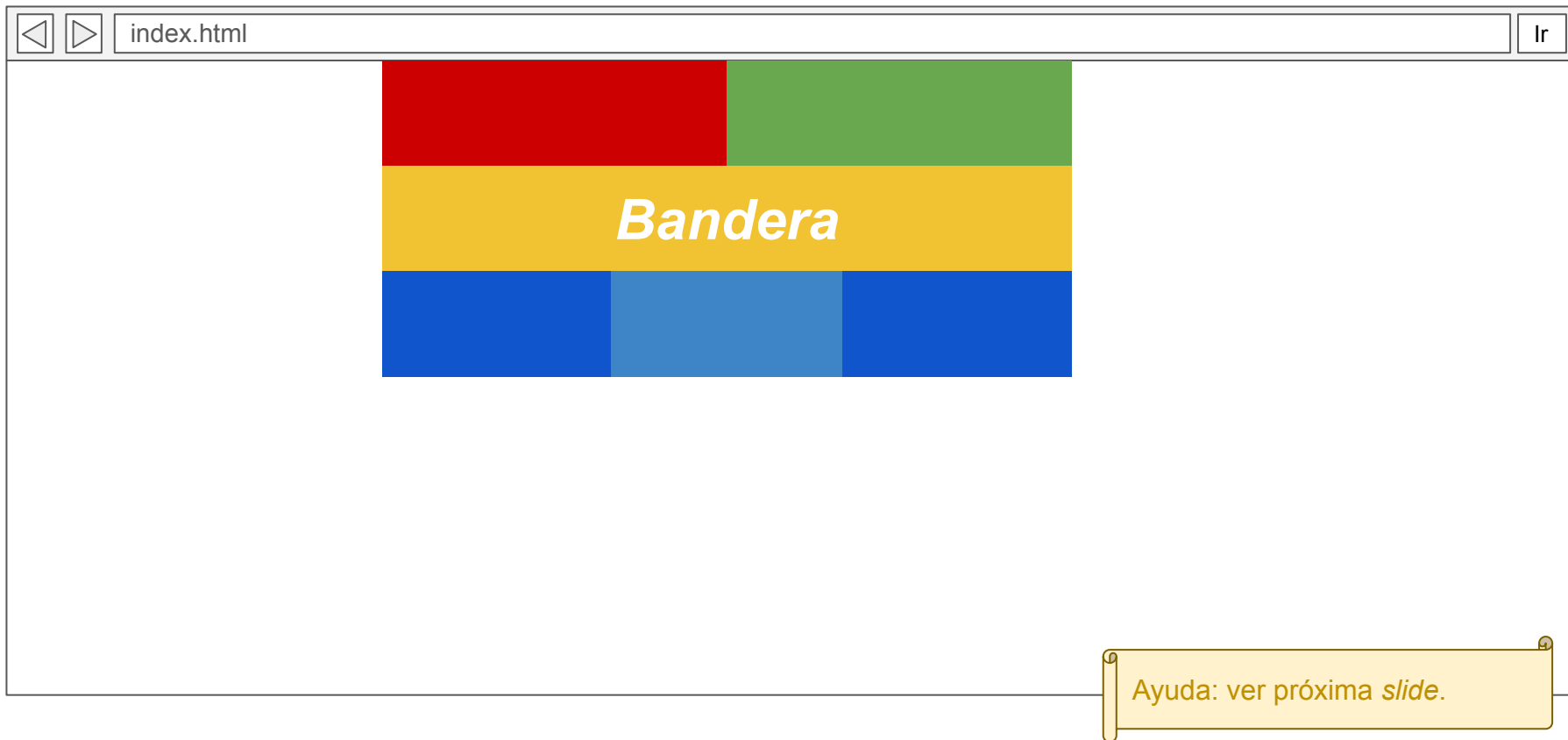


Ejercicio 3

1. Crear una carpeta en el Escritorio (o donde prefieran) con el nombre `Clase03_Ejercicio_3`.
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. Crear el HTML y CSS para lograr un resultado similar al siguiente diagrama:

Ejercicio 3

EXTRA



Ejercicio 3 – Ayuda

```
<div id="flag">  
  <div id="stripe-top">  
    <div id="stripe-top-left">  
    </div>  
  </div>  
  
  <div id="stripe-middle">  
    Bandera  
  </div>  
  
  <div id="stripe-bottom">  
    <div id="stripe-bottom-middle">  
    </div>  
  </div>  
</div><!-- /#flag -->
```

