

Epic 1: Creación de la primera web

= web CampSite Academy =

Ahora que estamos un poquito más familiarizados con html es hora de empezar el proyecto. Recuerda crear una carpeta nueva con el nombre del proyecto, luego ábrela en Visual Studio Code.

Sprint 1

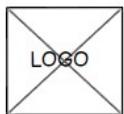
Crea home.html

Una vez la carpeta del proyecto esté abierta en VSC, crea un nuevo archivo llamado home.html. Empieza por el "esqueleto" del un html i empieza a trabajar en el tag <body>.

TIP: VSC viene con un emmet pre instalado, así que conoce algunos atajos dependiendo de la extensión del archivo. Si el archivo es .html y escribes html:5 y luego la tecla tab creará la estructura automáticamente.

En caso que lo hayas olvidado, para ver tu código en tiempo real guarda tu trabajo y arranca **Live Server**.

Antes de pensar sobre el estilo y la disposición de los elementos, debes añadir todo el contenido de la página. No te preocupes por cómo se ve, simplemente intenta organizar los tags debidamente. Ahora mismo debería verse tal que así:



Campsite Academy

Próximas fechas

26

Lun

Sept

Curso Full-Time

Online

Full-Stack MERN

[Inscribeme](#)

26

Lun

Sept

Curso Full-Time

Online

Data Science

[Inscribeme](#)

16

Lun

Ene

Curso Full-Time

Online

Full-Stack MERN

[Inscribeme](#)

Puedes descargar las imágenes [aquí](#):

Debe verse más o menos como el sketch anterior, simplemente unos cuantos tags en el archivo html. No le des importancia a que el texto de la barra de navegación no lleve a ningún lado, ya lo convertirás en links cuando crees más páginas. Lo mismo para los botones, ahora no sirven.

Como puedes ver, hay muchos elementos distintos: hipervínculos, imágenes, títulos, párrafos... intenta utilizar el tag correcto en cada ocasión. Puedes echarle un vistazo a esta [Cheat Sheet](#) si necesitas ayuda.

[Aquí](#) están los sketches de las otras páginas, que tendrás que hacer a continuación.

Crear conocenos.html y contacto.html

Siguiendo los pasos de la tarea anterior, crea dos archivos más: conocenos.html y contacto.html.

Ya tienes los sketches para estas dos páginas nuevas, con sus correspondientes imágenes

Añade los links

Ahora que tenemos la estructura y contenido, es hora de vincular todas tus páginas entre si. Para ello, utiliza el tag `<a>`. Hazlo dentro de la navbar `<nav>` , así cuando hagas click a home irá a `home.html`, y lo mismo para las demás. Esto hará que tu web sea un poco más dinámica.

Valida tu código

Existen validadores online donde puedes ver si hay algún error o advertencia en tu código, tales como [el HTML validator](#). Puedes copiar y pegar tu código para ver si está escrito como debería. Revisa todos los errores y advertencias que te salgan.

A partir de ahora, crea el hábito de validar tu código regularmente, ya que te evitará dolores de cabeza resolver pequeños problemas en vez de intentar resolverlos todos a la vez al final.

Añadir estilos y mejoras visuales

Hasta ahora hemos añadido el contenido. Pero como colocamos todo en su sitio? Pues bien, es hora de aprender **CSS**.

Puede que hayas visto diferentes manera de dar estilo a una página web, desde atributos de estilo dentro de los tags hasta el tag `<style>` dentro del `<head>` donde puedes escribir css. Desde ahora vamos a utilizar lo que es más común y preferible, el **archivo .css externo**.

Añade estilo a tu proyecto actual

Dentro de la carpeta del proyecto, crea una carpeta que se llame styles y dentro el archivo main.css (o style.css). Una vez lo tengas, conéctalo a tus html y sigue estos wireframes para la organización de los elementos.

Desde [aquí](#) te puedes descargar las imágenes con la disposición de los elementos.

Si tienes la misma estructura en todos los html con las mismas clases, te será mucho más fácil tener un estilo homogéneo entre todas las páginas.

Antes de saltar a la siguiente tarea, valida tu trabajo con el [CSS validator](#) y comenta tu trabajo con el mentor.

Sprint 2: Crea tu propio estilo

Ahora ya hemos visto la base tanto de html como de css. Has podido aplicar el estilo de un wireframe a la perfección, que es lo que a menudo se te va a pedir si en el proyecto hay un diseñador gráfico. Aquí no tenemos, así que es una gran oportunidad para ver tu gran talento artístico!

Story 1: Un nuevo archivo

No debes borrar todo el css que ya has hecho. Crea otro archivo css y no olvides hacer el link en cada uno de los archivos html.

Para esta tarea, y a partir de ahora, puedes hacer cambios estructurales en el html para que se ajuste lo máximo posible a tu diseño. El contenido debe permanecer igual.

Una vez hayas acabado, contrástalo con tu mentor para que te de inputs sobre tu diseño. Tened en cuenta que no vamos a juzgar el diseño en sí, no somos diseñadores. Aunque la totalidad de la página web debe homogénea y no romper con los estándares.

Sprint 3

Tablas

Vamos a continuar con más tags importantes de html. Cada día nos encontramos con una u otra estructuración de información en forma de tabla, ya sea en un calendario, clasificaciones en una liga... Así pues, veamos cómo crear una con el tag <table>.

Primero de todo, puedes hacer tus pruebas en tu archivo de test para no dañar accidentalmente el proyecto.

A grandes rasgos, las tablas se componen de **filas** <tr> y de columnas (o mejor dicho, de **celdas** <td> dentro de estas filas). Puedes probar con lo siguiente:

```
<table>
  <tr>
    <td>Lluís</td>
    <td>Barcelona</td>
    <td>ES</td>
  </tr>
  <tr>
    <td>Alberto</td>
    <td>Madrid</td>
    <td>ES</td>
  </tr>
  <tr>
    <td>Fernando</td>
    <td>Buenos Aires</td>
    <td>AR</td>
  </tr>
</table>
```

Acabas de crear tu primera tabla simple! Como verás, es una tabla con tres filas y tres celdas dentro de cada una de ellas.

Pero eso no es todo, podemos organizar las tablas con **encabezados** `<thead>` y **contenido** `<tbody>`. Una peculiaridad es que dentro del `<thead>`, las celdas pasan a ser **celdas de encabezado** `<th>`.

También existe el **pie de tabla** `<tfoot>` por si la última fila fuera diferente (estructuralmente hablando) a las demás, como por ejemplo la fila Total en una factura.

Si modificamos la tabla anterior:

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Ciudad</th>
      <th>País</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>Lluís</td>
      <td>Barcelona</td>
      <td>ES</td>
    </tr>
    <tr>
      <td>Alberto</td>
      <td>Madrid</td>
      <td>ES</td>
    </tr>
    <tr>
      <td>Fernando</td>
      <td>Buenos Aires</td>
      <td>AR</td>
    </tr>
  </tbody>
</table>
```

Hasta ahora hemos visto:

- <table> --> table
- <thead> --> table head
- <tbody> --> table body
- <tfoot> --> table foot
- <tr> --> table row
- <th> --> table heading
- <td> --> table data

Nuevo html, información de los cursos

Vamos a crear un nuevo archivo html llamado **info.html** donde, como el nombre indica, vamos a añadir algo de información sobre los cursos con tablas y listas (las veremos próximamente).

Y qué mejor información de los cursos que la de los alumnos! **Cread una tabla con tantas filas como alumnos, y en las columnas acordad entre vosotros qué poner;** por ejemplo el nombre, apellidos, ciudad, hobby...

Añadir estilos a la tabla

Las tablas son el elemento perfecto para ver un tipo de pseudoclases en CSS. Usadlas junto a los padding, border, text-align, font-size... para crear una tabla bien estilizada. Este [link](#) habla de otras propiedades de CSS para las tablas. Recuerda ir revisando el progreso con el mentor.

Listas

Las listas, como su propio nombre indica, sirven para listar un conjunto de elementos o texto. Pueden ir des de listas pequeñas como la lista de la compra, hasta listados con mucho texto como puede ser un contrato o la política de privacidad de una página web.

Hay tres tipos de listas, las **ordenadas**, **desordenadas**, y las listas de **descripción**. Todas tienen un esqueleto común: un tag padre, seguido de sus hijos.

En el caso de las listas desordenadas las abrimos con `` (unordered list), seguidas de cada uno de sus puntos `` (list item).

Lo mismo aplica para las listas ordenadas, la única diferencia es el tag padre.

En este caso empezamos la lista con `` (ordered list).

Las listas de descripción son un tanto diferentes. Las abrimos con el tag `<dl>` (description list). Consiste de un par de tags hijos, `<dt>` (define el término/nombre) y el `<dd>` (define dicho término/nombre).

Un ejemplo de listas sería:

```
<p>TO DO:</p>
<ol>
    <li>Levantarse de la cama</li>
    <li>Desayunar</li>
    <li>Ir a trabajar</li>
</ol>

<p>Ingredientes para el desayuno:</p>
<ul>
    <li>Café</li>
    <li>Leche</li>
    <li>Tostada</li>
</ul>

<p>Descripción de los ingredientes del desayuno</p>
<dl>
    <dt>Café</dt>
    <dd>Bebida caliente de color oscuro</dd>
    <dt>Leche</dt>
    <dd>Bebida fría de color blanco</dd>
    <dt>Tostada</dt>
    <dd>Pan pasado por la tostadora</dd>
</dl>
```

Copia y pega este código para ver las diferencias entre las listas.

Es posible cambiar el estilo de la listas, lo podéis ver con la propiedad list-style de CSS, que es una combinación de list-style-image, list-style-position y list-style-type.

Story 5: Amplia la información

Para seguir añadiendo contenido en html de información, utilizaremos las listas para mostrar todo lo que habéis aprendido hasta el momento, tanto de HTML como CSS. Podéis juntarlo en una lista o separarlo en dos, como quede mejor con vuestro diseño. Eso sí, intentad usar listas dentro de listas. Un ejemplo sería si queremos mostrar esto:

- HTML
 - tags
 - <p>: párrafo
 - : imagen
 - ...
 - atributos
 - class
 - ...

Nota: Siempre que queramos una lista dentro de otra, debe formar parte de un .

Ejemplo:

```
<ul>
    <li> item en la lista exterior </li>
    <li> Inicio lista interior
        <ol>
            <li> item en la lista interior </li>
        </ol>
    </li>
</ul>
```

Intenta también crear una **lista de descripción** con el horario detallado del día.

Aquí termina la página de información. Ahora que tienes todos los elementos, dale el estilo correspondiente y que vaya acorde con las otras páginas. No olvides ir revisando con el mentor.

Formulario de inscripción

Formulario de inscripción

Para finalizar con este proyecto, vamos a crear un formulario de inscripción a los cursos. Será estático, ya que en este proyecto no tocamos nada de JavaScript.

Siempre podéis volver en unas semanas/meses y mejorar el proyecto, ya sea organizando el Html, simplificando el Css o añadiendo JavaScript para dar dinamismo a la página y mejorar la funcionalidad.

Para que sea un formulario bien completo, vamos a dividirlo en tres partes: Información personal, información sobre el curso escogido y comentarios extra. Te recomiendo que, para una mejor distribución y organización de las partes, las separes en secciones diferentes (`<section>` o `<div>`).

En la primera parte de **información personal** veremos todo tipo de inputs. `text` para el nombre, apellidos, población; `number` para la edad y el código postal; `tel` para el teléfono; `email` para el correo electrónico, `radio` para el género... Añade todos los que creas necesarios.

En la parte de **información sobre el curso** serán inputs de selección. radio para el tipo de curso (Full-time o Part-time), select para las diferentes ciudades (pon tantas como te apetezca), date para la fecha de comienzo, checkbox para poder seleccionar tecnologías y/o herramientas previamente vistas (html, css, flex-box, Git, javaScript...). Igual que en la anterior, puedes añadir otros inputs que creas importantes.

Y por último, la parte de comentarios será un textarea. Añade un botón de Submit / Enviar, aunque no tenga funcionalidad.

TIPS:

- Recuerda poner el atributo required a los inputs que son obligatorios.
- Es un gran ejercicio para practicar flex-box. Dibuja la distribución en un papel y reproducelo en código. También es buen momento para utilizar la child-property: flex-grow. Intenta combinar filas de uno, dos y hasta tres inputs dependiendo del tamaño de estos. Ej.: los inputs de Nombre y Apellidos en la misma fila. Población, Provincia y CP en otra.
- Si utilizas <fieldset> para agrupar inputs, ten en cuenta que no acepta flex-box.
- No olvides conectar el tag <label> con su input correspondiente.

Para terminar, conecta los botones "inscribirme" del primer html para que te manden a ésta página.

Verifica el código, revísalo con el mentor y aprovecha este tiempo para corregir y mejorar todo el proyecto.