

PLA 2: HTML5

Módulo 1

Dedicación 4 días (20 horas)

Entrega 9 de mayo de 2024

Descripción y objetivos

En esta segunda PLA trabajaremos con el lenguaje de marcas HTML5, que es lo que nos permite crear la estructura y contenido de una página web. Veremos cómo crear cabeceras, listas, enlaces, tablas y formularios.

También veremos cómo añadir imágenes, audios, vídeos y otros recursos externos.

El objetivo de las PLA 2 y 3 es maquetar una aplicación web con HTML5 y CSS utilizando las etiquetas semánticas correctas y la distribución de componentes adecuada para su correcta visualización en pantallas de tamaño convencional y también en dispositivos móviles. En la PLA 2 ponemos el foco en HTML5, mientras que en la PLA 3 lo pondremos en CSS.

Justamente porque en esta PLA 2 trabajamos HTML5 y no tocamos todavía cuestiones de estilos (ya sabe que CSS nos permitirá configurar aspectos como las fuentes, los colores, los layouts, los tamaños, etc.), en esta PLA es necesario dar por buenos los estilos que su navegador le propone para cada tipo de etiqueta, sin entrar a modificarlos con CSS. En esta PLAN 2 no puede haber ni una sola línea de CSS.

Recuerde que, en paralelo a la resolución de la PLAN, conviene que comience a pensar en su proyecto final. En el repositorio [Public APIs](#) encontrará una gran cantidad de API públicas que puede utilizar. Céntrese en las API que tengan HTTPS y CORS informados con un Yes (pero si le llama mucho la atención alguna que no cumple este requerimiento, hablemos y estudiaremos si puede utilizarla o no). Buscando por su cuenta puede encontrar muchas otras API públicas y gratuitas, por ejemplo en [Free Public APIs for Developers](#). En cualquier caso, asegúrese de no introducir nunca su tarjeta de crédito para registrarse en ninguna API pública. Si le solicitan una tarjeta de crédito, busque otra API.

Tareas a resolver

En esta PLAN debe crear varios documentos HTML y deberá inventar toda una serie de contenidos. Pero no ponga textos sin ningún sentido ni interés. Aproveche la PLAN para crear una especie de apuntes sobre lo que va aprendiendo. Por ejemplo, en la página `texts.html` puede explicar las diferentes

etiquetas que sirven para escribir texto y mostrar algunos ejemplos. Seguramente se topará con el problema de cómo hacer que un documento HTML muestre las etiquetas (p.ej. `<p>` o ``) en lugar de interpretarlas. Intente encontrar la solución por su cuenta o pregunte en el foro y miraremos cuál alternativa podemos proponer.

Aunque el libro [HTML5 Notes for Professionals](#) le puede ayudar en algunos puntos y que en [W3Schools](#) puede encontrar contenidos que le ayuden a entender algunos conceptos en momentos puntuales, conviene que os acostumbre a trabajar tanto como sea posible con la MDL. En concreto comience con estos enlaces y, a partir de aquí ya través del buscador, vaya localizando todos los artículos que necesite:

- [Conceptos básicos de HTML](#)
- [Empezando con HTML](#)
- [Estructurar la web con HTML](#)
- [HTML: lenguaje de marcado de hipertexto](#)
- [Referencia de elementos HTML](#)

Siempre que se solicite una explicación breve debe ser de máximo 5 líneas y debe estar incluida en el contenido visible del archivo .html correspondiente abriéndolo con un navegador web. No se ha de esconder la explicación a los comentarios del código fuente utilizando blogs `<!-- -->`.

1. Estructura de un documento HTML

Empezaremos el estudio del lenguaje HTML analizando la estructura básica de un documento. Consulte y trabaje los apartados de la MDL que hablan sobre las etiquetas básicas que nos permiten crear un documento HTML.

¿Qué hay que entregar?

Un archivo `index.html` que haga de puerta de entrada para el resto de archivos que deberá crear en esta PLAN. Añada comentarios con la sintaxis `<!-- strawberry fields forever -->` por explicar qué hace cada sección de su documento. Utilice en este archivo tantas etiquetas diferentes como sea necesario de todas las que verá en los siguientes apartados de esta PLAN.

2. Textos

Como es evidente, una parte básica de cualquier documento HTML son los textos. Los podemos dividir en diferentes categorías: títulos (o cabeceras), subtítulos, párrafos, textos en cursiva o en negrita, etc.

Busque los apartados de la MDL donde hablan sobre las etiquetas que nos permiten trabajar con los diferentes tipos de texto:

- Encabezamientos `<h1>` a `<h6>` y párrafos.
- Etiquetas para mostrar tipos especiales de texto como ``, ``, `<yo>`, ``, `<marca>`, `<pequeño>`, ``, `<ins>`, `<sub>` i `<sup>`.
- Etiquetas con semántica especial como `<blockquote>`, `<q>`, `<abr>`, `<dirección>`, `<cita>` i `<bdo>`.

Lea también las secciones Usage Notas de los artículos [: The Strong Importance element](#) y [: The Emphasis element](#). Hablan sobre el uso de las etiquetas `` vs `` y también `` vs `<i>`. Si es necesario, busque otros artículos en la red que traten este tema en concreto.

¿Qué hay que entregar?

Un archivo `texts.html` con ejemplos de, al menos, las etiquetas `<h1>` a `<h3>`, `<p>`, ``, ``, `<blockquote>` y `<cite>`. ¡Pero puede ir más allá de estas! Recuerde que también puede utilizar textos en cualquiera de los otros archivos que forman esta PLAN y que conviene utilizar las etiquetas más adecuadas para cada caso.

Una breve explicación (recuerde: máximo 5 líneas e incluida en el contenido visible del archivo anterior cuando lo abrimos con un navegador web) sobre cuándo conviene más utilizar la etiqueta `` y cuándo tiene más sentido utilizar ``. Y una segunda explicación similar respecto a las etiquetas `` y `<i>`.

3. Imágenes

Aparte de los textos, las imágenes toman un papel primordial en casi cualquier sitio web, ya sea en formato grande (piense en las fotografías en páginas de noticias o en cualquier red social) o en forma de pequeños iconos.

Busque los apartados de la MDL donde hablan sobre la etiqueta que nos permite trabajar con imágenes.

¡Cuidado! Cuando trabajamos con imágenes locales debemos mirar muy bien cómo ponemos la ruta para que la imagen se muestre correctamente. Puede ocurrir que usted la vea bien en su máquina pero que cuando haga el deploy en un servidor la imagen no se muestre. Para evitarlo, utilice siempre rutas relativas y no rutas absolutas y abra su sitio web con la opción Go Live de la extensión Live Server. Esta precaución aplica igualmente cuando tenga que trabajar con otros archivos multimedia como audios o vídeos.

¿Qué hay que entregar?

Un archivo `images.html` con varios ejemplos de cómo mostrar imágenes en una página web. Debe incluir, como mínimo, dos tipos de imágenes: locales y remotas. Para las imágenes locales utilice cualquier imagen que tenga en su máquina (y adjunte al directorio correspondiente para que se muestre en la página web). Para las imágenes remotas, utilice una URL como <https://picsum.photos/500/300> o similar. Y recuerde que también puede utilizar imágenes en cualquiera de los otros archivos que forman esta PLAN.

Recuerde que en el PDF de la sesión síncrona del 2 de mayo encontrará, en la última página Demo – Herramientas

útiles, enlaces a varios sitios que le permiten descargar imágenes (tanto fotografías como iconos) de cualquier temática y en varios formatos.

4. Enlaces

HTML significa lenguaje de marcado de hipertexto. Y lo de hipertexto no significa otra cosa que... Hypertext es texto displayed en el ordenador de display or otras electrónicas de referencias (hyperlinks) a otro texto que el reader puede acceder inmediatamente. Es evidente, por tanto, que las páginas web no pueden concebirse sin enlaces.

Consulte y trabaje los apartados de la MDL donde hablan sobre la etiqueta que nos permite trabajar con enlaces.

¿Qué hay que entregar?

Un archivo `links.html` que explique, en todos los casos con una explicación en forma de texto acompañada de ejemplos suficientes, los siguientes conceptos: 1) el funcionamiento del atributo `target`, 2) la diferencia entre rutas absolutas y rutas relativas y 3) el funcionamiento de la opción para enlazar directamente con una dirección de correo.

5. Listas

Pese a no tener la misma importancia fundamental de los anteriores componentes estudiados, las listas suponen una forma muy cómoda de estructurar la información de una misma temática. Por ejemplo, quizás todas las etiquetas HTML que va aprendiendo podrían formar parte de una lista y todas las tareas pendientes de resolver podrían ser otra. Aparte, a menudo utilizamos las listas para crear los menús de nuestros sitios web.

Consulte y trabaje los apartados de la MDL donde hablan sobre las etiquetas que nos permiten trabajar con listas.

¿Qué hay que entregar?

Un archivo `lists.html` que muestre, con suficientes ejemplos variados, el funcionamiento de las etiquetas ``, `` y ``. Cree, entre otras, una lista con las diez etiquetas HTML que le parecen más importantes de entre todas las que ha trabajado en esta PLAN. Recuerde que también puede utilizar listas en cualquiera de los otros archivos que forman esta PLAN si esto le permite estructurar mejor la información.

6. Tablas

Si alguna vez ha trabajado con cualquier informe en formato tabular o ha utilizado Excel, sabrá que a menudo una tabla es la mejor manera de mostrar una información que sigue una misma estructura. En las páginas web también nos encontramos de forma habitual con la necesidad de mostrar información tabulada: los elementos de una cesta de la compra (artículo, número de unidades, precio, subtotal...), los valores de unas acciones (título, precio de apertura, precio de cierre, volumen...), etc.

Consulte y trabaje los apartados de la MDL donde hablan sobre las etiquetas que nos permiten trabajar con tablas, por ejemplo:

- [tablas HTML](#)
- [<table>: El elemento Tabla](#)

¿Qué hay que entregar?

Un archivo `tables.html` que incluya una tabla similar a la que puede encontrar cuando entra en el foro del aula (allí donde pone Debate, Iniciado por, Último mensaje...). Añada un mínimo de 5 filas con contenido inventado. Si lo desea, puede añadir imágenes (avatar, icono estrellito). Ya sabe que en cualquier navegador puede utilizar las opciones Inspect Elemento Show Page Source o similares para ver el código HTML de una página web. Así que este ejercicio puede resultarle bastante sencillo, pero intente entender exactamente qué está haciendo línea a línea en lugar de simplemente copiar un contenido desde la página web del campus.

Una breve explicación (máximo 5 líneas, contenido visible) sobre si es buena idea definir la estructura de una página web (menú superior, contenido principal, menú con opciones a derecha e izquierda, pie de página, etc.) aprovechando el estructura de dos dimensiones que nos permite crear una mesa.

Razone muy bien su respuesta, tanto si opta por el sí como si lo hace por el no.

7. Formularios

Cada vez que, como usuarios de una página web, debemos rellenar algún tipo de información (texto, marcar una opción entre varias opciones, seleccionar un número o una fecha, etc.) estamos utilizando un formulario. Cuando lo enviamos, toda la información que hemos agasajado llegará al servidor que previamente se haya definido. No piense en un formulario sólo en el sentido tradicional, formado por una variedad de campos a llenar. Todo lo contrario: si necesita simplemente que los usuarios de su página web le hagan saber qué edad tienen o cuál es su color preferido, esto también implica tener que crear un formulario.

La creación y el trabajo con formularios es una de las competencias más importantes del mismo módulo 1. Invierta todo el tiempo que convenga en esta tarea, porque le será muy útil en el futur.

Consulte y trabaje los apartados de la MDL donde hablan sobre las etiquetas que nos permiten trabajar con formularios:

- [<form>: El elemento Formulario](#)
- [<input>: El elemento de entrada \(entrada de formulario\)](#)

¿Qué hay que entregar?

Un archivo `forms.html` que incluya un formulario web que utilice, como mínimo, las siguientes etiquetas: `<form>` , `<input>` (de los tipos `texto` `number` , `radio` , `caja` , `entregar` , `color` , `fecha` , y varios más de los disponibles), `<label>` Investiga y juegue con los , `<seleccionar>` i `<área de texto>` .
diferentes atributos disponibles para cada tipo de elemento. Utilice los ejemplos necesarios para que su formulario tenga un aspecto visual agradable sin utilizar CSS en ningún momento.

8.Estructuración en secciones

Las aplicaciones web tienen distintas secciones. Puede haber una parte que incluya los enlaces a las distintas páginas del sitio web. Otra parte que muestre la cabecera de la página con una imagen gorda y un título impactante. Otra que, debajo de todo, muestre la información legal y los créditos de forma discreta y sin robar atención al contenido principal de la página... Todas estas partes forman la estructura del sitio web.

Lea las diferentes secciones de los capítulos 19, 20 y 21 del libro HTML5 `Notes for Professionals` . Después, haga un repaso a los apartados de la MDL donde hablan sobre las etiquetas que nos permiten trabajar con secciones.

¿Qué hay que entregar?

Un archivo `sections.html` donde se muestra, con explicación de texto y demostración práctica con ejemplos, el funcionamiento de las etiquetas `<nav>` , `<encabezado>` , `<pie de página>` i `<div>` .

Una breve explicación (máximo 5 líneas, contenido visible) sobre para qué sirve añadir `role="navigation"` en un elemento `<nav>` o un elemento `<div>` . Consulte la página [ARIA: navigation Role](#), pero explique el concepto con sus palabras.

9. Multimedia

Si ha llegado hasta aquí puede sentirse satisfechos con su trabajo y con todo lo que ha aprendido y practicado. Tiene ya las herramientas para crear páginas web con un gran número de elementos diferentes. Pero es evidente que en Internet de hoy en día los audios, los vídeos y otros elementos multimedia juegan un papel muy relevante. Si entra a ver cualquier vídeo de YouTube e inspecciona el código HTML de la página mientras tiene suficiente paciencia para ir entrando en etiquetas `<div>` cada vez más anidadas, al final llegará a una etiqueta `<video>` que es la que muestra el vídeo en sí mismo.

Lea las primeras secciones del capítulo 25 del libro [HTML5 Notes for Professionals](#) para tomar una idea básica de cómo funcionan las etiquetas `<audio>` y `<video>`. Después, consulte y trabaje los apartados de la MDL donde hablan sobre las etiquetas que nos permiten trabajar con recursos multimedia. Entre otros, [Multimedia and Embedding](#).

¿Qué hay que entregar?

Un archivo `media.html` donde incruste un archivo de audio y un archivo de vídeo locales. Añada también, con `<iframe>`, un vídeo de YouTube o un mapa de OpenStreetMap. Los archivos locales deben ser breves. Además, no olvide adjuntarlos al directorio correspondiente y tenga presente lo que se ha explicado anteriormente sobre `Go Live` y las aplicaciones que funcionan en nuestra máquina pero dejan de hacerlo cuando hacemos deploy en un servidor.

Entrega

Un archivo `c2409_PLA2_Apellido_Nombre.zip` con tantos archivos `.html` como sea necesario con las soluciones de las tareas propuestas. Obligatoriamente debe haber un archivo `index.html` que sea el punto de entrada en su página web. Y cualquier otro archivo debe estar enlazado con etiquetas `<a>` desde ese archivo `index.html` o desde cualquier otra página que, a su vez, esté enlazada desde la página `index.html`. Por tanto, desde la página inicial se debe poder acceder a todos los archivos `.html` que envíe.

notas finales

1. Será sensacional si utiliza un repositorio privado en GitHub con el nombre `cifo-frontend-2024` para ir subiendo el contenido de cada PLAN en directorios separados. Los archivos correspondientes a esta PLAN deberían ir al directorio `plan2`. Y será aún mejor si, además, el historial de commits muestra cómo ha ido solucionando esta PLAN poco a poco, tarea a tarea, apartado a apartado. Esto permite

ver cómo ha ido trabajando las tareas a desarrollar. En un proceso de selección laboral ésta es una de las cosas que más valoran los entrevistadores.

2. **Nunca copie / plagie ninguna** respuesta ni fragmento de código sin añadir las referencias correspondientes. En programación es habitual —y correcto— copiar pequeños bloques de código de distintos sitios. Pero si una parte relevante de lo que entrega no es de elaboración propia es necesario que lo mencione. No hacerlo supondrá una calificación de cero puntos.
3. Si necesita más información sobre alguno de los temas tratados en esta PLAN, el libro [HTML5 Notes for Professionals](#) incluye ejemplos de prácticamente todas las etiquetas HTML que debe utilizar para la resolución de todas las tareas. Y en la MDL ya sabéis que siempre está todo.
4. Opcionalmente, añada un último archivo `metas.html` donde explique la función de las etiquetas `<meta>` de uso más habitual. Puede explicar, por ejemplo, qué hacen los atributos `charset` o `content`.
5. Las guías de estilo son muy importantes y conviene que las tenga presentes desde el principio, para que conseguir escribir un código limpio, elegante y menos propenso a errores. Tiene muchas a su disposición... y también se puede crear una propia, aunque esto sólo se recomienda si ya tiene bastante experiencia escribiendo código HTML. Para empezar puede consultar, por ejemplo, la [Google HTML/CSS Style Guide](#).
6. Si bien herramientas como GitHub Copilot o ChatGPT son de innegable ayuda a la hora de programar, es muy importante que entienda perfectamente todo el código que entregue y que lo sepa explicar cuándo se lo pidan. En caso contrario le será difícil superar entrevistas de selección laboral para acceder a posiciones de programación.