

Natalia García Quintanilla

Herramientas HTML y CSS

PEC1 - Diseño de una web

1. Entorno de trabajo.

Basaremos el desarrollo de esta web en el entorno de trabajo indicado en el Módulo 2 de la asignatura.

Una vez realizada la instalación de npm, seguí los pasos para incorporar Parcel a mi proyecto. Después, creé un ejemplo simple de index.html, ejecuté npm run build, y después npm run dev para comprobar que el servidor se había lanzado y funcionaba correctamente.

A continuación, instalé rimraf y añadí las líneas indicadas al archivo package.json para el parcel del index y la limpieza de caché.

En lo que respecta a la compatibilidad en distintos navegadores, añadí el archivo .browserslistrc a la raíz del proyecto. También incorporé el plugin Autoprefixer de PostCSS, así como el archivo .postcssrc. Creé mi propia hoja de estilo y ejecuté npm run build.

2. Selección de la plantilla

Como base para nuestro sitio web tomaremos una de las plantillas disponibles en <https://html5up.net/>. Para que el diseño no recaiga únicamente en algo realizado por terceros, distinguiremos a partir de ahora entre dos hojas de estilos:

- **style.css** (en la raíz del proyecto). Aquí será donde introduzca mi diseño personal, animaciones, colores, etc.
- **main.css** (en la carpeta `assets/css`). Donde se indicarán las clases previamente establecidas por la plantilla original, así como la adaptación responsive mediante media queries.

NOTA: para la modificación de las clases por defecto utilizaremos la herramienta inspeccionar elemento del navegador, en aras de localizar los estilos aplicados las distintas secciones.

3. Index (index.html)

Se trata de una página sencilla, con una barra de navegación desde la que se puede acceder tanto a las categorías como a las páginas de detalle (esta barra será común al resto de ventanas, con una `class="current"` que señalará el apartado actual).

En ella, quedan presentadas las tres categorías a tratar sobre el desarrollo front-end: Module Bundlers, Preprocesadores y Lenguajes procesados en cliente. La presentación incluye iconos svg de <https://fontawesome.com/>, importados mediante la etiqueta `<link>` en el head. Posteriormente se añadirán en `package.json` como dependencia adicional, ejecutando `npm install @fontawesome/fontawesome-free`

También se incluye una animación del texto. Tomé el código base de un CodePen más complejo y lo apliqué únicamente a esta sección creando el `id="homepage"` y aplicando esos estilos a este identificador.

4. Module Bundlers (bundlers.html)

La página presenta la siguiente estructura:

- Título
- Subtítulo
- Imágenes a modo de ejemplo (una visible)
- Botón para la ejecución del ejemplo
- Definición de los Module Bundlers
- Subtítulo
- Prestaciones de los Module Bundlers
- Barra lateral con enlaces a ejemplos (detalle) de Module Bundlers

Implementación del ejemplo:

Las imágenes (bundle1.png y bundle2.png en carpeta images) fueron obtenidas de la web:

<https://dev.to/tanhauhau/what-is-module-bundler-and-how-does-it-work-3gp2>

Después, se optimizaron para reducir su peso mediante una reducción de la anchura original en Photoshop (en dist se mantienen las dimensiones a pesar de haber instalado el plugin imagemin).

La idea es que bundle1 aparezca al cargar el documento, y sea sustituida por bundle2 tras hacer clic en el botón inferior, de forma que la visualización del proceso sea más dinámica y requiera de una acción por parte del usuario. Para implementar esta funcionalidad utilicé el archivo index.js, vinculado mediante la etiqueta <link> al resto de ventanas. Aquí jugué con la visibilidad de cada imagen, aplicando un display al id de cada una según el caso. Las funciones de index.js son llamadas en bundlers.html, por un lado en el onLoad del body; por otro, en el propio botón.

La documentación para esta página se obtuvo de la web indicada arriba.

5. Parcel (parcel.html)

Esta es la página detalle asociada a los Module Bundlers. Presenta la siguiente estructura:

- Título
- Imagen ilustrativa (optimizada)
- Definición de Parcel
- Subtítulo
- Automatización de las conversiones
- Subtítulo
- Optimización de código
- Botón de enlace a las instrucciones de instalación (https://parceljs.org/getting_started.html), con un atributo `target="_blank"` para no sacar al usuario de nuestra web.

La documentación para esta sección se tomó de la web:

<https://pablomagaz.com/blog/empaquetando-aplicaciones-con-parcel>

6. Preprocesadores (preprocesadores.html)

La categoría relativa a los Preprocesadores contiene:

- Título
- Subtítulo
- Imagen ilustrativa (optimizada)
- Definición de Preprocesador
- Directrices
- Subtítulo
- Imágenes del ejemplo práctico (una visible)
- Botón para la ejecución del ejemplo
- Barra lateral con enlaces a ejemplos (detalle) de Preprocesadores

De nuevo, nos valemos de funciones JavaScript en el archivo `index.js` para visualizar el proceso de transformación de lenguajes mediante imágenes dinámicas. Para el contenido, me he ayudado de las diapositivas publicadas en:

<https://slideshare.net/LearnByWatch/preprocessor>

7. PostCSS (postcss.html)

Esta es la página detalle asociada a los Preprocesadores. Presenta la siguiente estructura:

- Título
- Imagen ilustrativa (optimizada)
- Definición de PostCSS
- Subtítulo
- Características principales
- Botón de enlace al repositorio de GitHub de PostCSS

(<https://github.com/postcss/postcss>), con un atributo `target="_blank"` para no sacar al usuario de nuestra web.

La documentación para esta sección se tomó de la web:

<https://webdesign.tutsplus.com/tutorials/postcss-deep-dive-what-you-need-to-know--cms-24535>

8. Lenguajes front-end (lenguajes.html)

La categoría relativa a los Lenguajes front-end contiene:

- Título
- Definición de los 3 lenguajes procesados en cliente
- Imagen ilustrativa (optimizada)
- Información adicional sobre HTML, CSS y JavaScript
- Barra lateral con enlaces al detalle de estos lenguajes

La documentación para este apartado, así como para la ventana detalle de HTML, se tomó del libro: *El Gran Libro de HTML5, CSS3 y JavaScript*, de J.D. Gauchat. Editorial MARCOMBO, tercera edición, 2017. ISBN: 978-84-267-2463-2

9. HTML (html.html)

Esta es la página detalle asociada a los Lenguajes. Presenta la siguiente estructura:

- Título
- Imagen ilustrativa (optimizada)
- Definición de HTML
- Definición y características de los atributos
- Subtítulo
- Tabla etiquetas básicas
- Subtítulo
- Tabla atributos básicos

Para el contenido de las tablas hubo que utilizar las llamadas HTML entities en aras de distinguir los símbolos <> de las propias etiquetas.

Recurso: https://www.w3schools.com/html/html_entities.asp

10. Hoja de estilos (styles.css)

Como se dijo anteriormente, aquí se incluirán los estilos independientes de la plantilla, que yo misma he añadido para adaptar la web a lo que se nos pide.

Etiquetas:

- body: con altura de 100vh (responsive), un fondo gris por si se añadieran secciones adicionales y una alineación de texto justificada.
- table, th, td y tr: para implementar el diseño de las tablas. Me ayudé de las pautas de https://www.w3schools.com/css/css_table.asp. También modifiqué el color de la cabecera y añadí un borde para separar ambas columnas.

Clases:

- `boton` y `boton: hover`: para darle un margen a los botones con respecto al resto de los elementos, así como un cambio de color al pasar el ratón por encima de los mismos.
- `lista`: para definir el estilo de las enumeraciones.
- `centrar`: para centrar elementos.
- `pic`: para centrar las imágenes y reducir su anchura relativa sin perder calidad.
- `icon.major` e `icon.major: hover`: clase propia de la plantilla detectada mediante las herramientas del navegador. Con ella convertimos los iconos principales de la página de inicio en botones que cambian de color y el diseño del cursor al pasar sobre él, indicando que se trata de un enlace.
- `animar`: para la animación de las imágenes en las páginas de detalle. Incluye `@keyframes` para implementar el movimiento.

Identificadores:

- `homepage`: para aplicar la animación de CodePen en el texto de la página de inicio. Incluye `@keyframes` asociados al nombre de la animación (`float`) y variables de color.

10. Publicación en remoto

Una vez tenemos todos los archivos en local, creamos una cuenta y un repositorio en GitHub al que llamaremos Herramientas.

A continuación hemos de instalar Git Bash para ejecutar el código que vinculará nuestro proyecto con el repositorio creado y subirá los archivos.

En Git Bash, accedemos a la ubicación de nuestra raíz y ejecutamos:

1. `$ git init`
2. `$ git add .`
3. `$ git commit -m "First commit"`

4. \$ git remote add origin <https://github.com/ngarci20/Herramientas.git>
5. \$ git remote -v
6. \$ git push origin master

Enlace al repositorio: **<https://github.com/ngarci20/Herramientas>**

Una vez tenemos el repositorio, debemos vincular los archivos a un dominio. Para ello empleamos Netlify y seguimos los siguientes pasos:

1. Creamos una cuenta y un nuevo site basado en GitHub.
2. Seleccionamos nuestro repositorio de GitHub.
3. Branch to deploy: master
4. Build command: npm run build
5. Publish directory: dist

Por último, personalizamos el dominio en Domain Settings. El resultado final se encuentra publicado en:

<https://flamboyant-pasteur-56c2af.netlify.com/>

11. Problemas

Todo funciona correctamente en local. Sin embargo, tanto al ejecutar `npm run dev` como en el servidor remoto, dejan de funcionar las funciones de mi archivo `index.js`, así como las animaciones incluidas en `style.css` (el resto de estilos de este archivo sí se incorporan a la web). Tampoco funcionan los despleguables del menú de navegación.

No he podido detectar el fallo. Tanto la hoja de scripts como la hoja de estilos están vinculadas en el `index.html`, y al ejecutar `npm run dev`, se incorporan al archivo `package.json`. También puede verse el código de estas hojas accediendo al código fuente desde el navegador. Sin embargo, siguen sin implementarse.