

PROYECTO

Vamos a migrar nuestro proyecto junto con el started-kit de Adalab.

VAMOS A HACER NUESTRO PRIMER EJERCICIO:

Vamos a migrar nuestro proyecto a la carpeta versión-0. Recordemos de la clase de hoy:

- * Si parto de un repositorio cojo todo lo que tenia en la carpeta (archivos antiguos) y, en la misma carpeta, creo otra carpeta que se llame “version-0” y los meto todo ahí y copio la plantilla en la carpeta superior a “version-0”

Lo primero que hacemos es darle a:

```
npm install
```

Para que nos cree la carpeta node_modules en nuestro proyecto. Pero esto no se sube a github!! Lo tenemos que generar cada una en nuestro proyecto en local.

Obviamos vulnerabilidades. Porque me puedo usar mi proyecto sin problema.

Podemos quitar las vulnerabilidades copiando el código que nos dice y lo “actualizamos”.

¿Cómo copio lo que tenía en nuestro proyecto?

Ahora nos vamos al html que teníamos montado, lo abro y copio la estructura ¿Dónde? En la carpeta de src, en el index.html

Lo único que tocamos en un proyecto con adalab-web-started-kit es la carpeta src. ¡El resto no lo tocamos!!!!

Entonces, vamos al index.html en src y pegamos la página que teníamos.

Copiamos la carpeta de imágenes y la ponemos dentro de src (en este caso la teníamos borrada como plantilla y la copiamos entera pero si tuviéramos la carpeta sería sólo copiar las imágenes de una carpeta a otra)

Cuando acciono el `npm start`, éste copia los archivos de src y los pega en public, nosotros solo tocamos src y public se genera con el `npm start`. Las carpetas fuera de index.html de src las copia en la carpeta de assets, todas: css, images, ...

¿Qué pasa entonces con nuestras rutas?

Que tenemos que poner las rutas de public. Es decir, tenemos que imaginar que el html está en public. Por ejemplo: `./assets/images/camiseta.png`

Podemos reemplazar todas a la vez para que no haya que ir una a una. Ella lo hace con un comando para reemplazar: `CTRL + F` → ponemos `./images` y le damos a reemplazar por `./assets/images`

Mañana hablamos mejor del CSS, pero así a capón:

Copio todo lo que había en el `main.css` de la versión-0 y me voy a la carpeta de `scss` y lo pego en el archivo de `main.css`. ¿por qué no salen mis archivos en la página? Porque las rutas vuelven a estar mal: `./assets/css/main.css`

Cuando copiemos todo podemos borrar la versión-0.

¡Sólo vamos a tener un archivo CSS!!! Metemos el reset dentro del `main.css`. Mañana lo vemos mejor.

Mañana repartimos también el `css`.

PARTIALS

Dividimos el `index.html` en `partials`.

Esto lo hacemos después del migrar el proyecto y hemos añadido el `started-kit`.

¿Cómo hago lo de los `partials`?

- Cojo un trocito de mi `intex.html` (en `src`), por ejemplo el header, lo corto.
- Creo un fichero en `partials` que se llame `header.html`.
- Todavía no hace nada.
- Vuelvo a mi `index.html` y pongo:

```
<include src="./partials/header.html/><include>
```

(la etiqueta de cierre es necesaria porque es como está programado, aunque nunca vamos a poner nada dentro)

¿Que pasa con esto? pues que le digo que este trocito está en este archivo. Es decir, con `include` pone lo que contiene el fichero que tiene en la `src`.

Cuando nos vamos a `public` (en el resultado) vemos que lo que nos sale es lo que viene en el `index.html` de `src` y, en vez de `include`, vendría lo que tiene el archivo `header.html`.

Puedo hacer un `include` dentro de otro, pero como sería hermano del primero, no habría que poner lo de `partials` en la ruta, ya que ya está en esa carpeta: `./nombre_archivo.html`

Borramos `more-info.html` porque no lo vamos a utilizar.

PARÁMETROS (PARAMETRIZAR)

Hay veces que tenemos casi exactamente el mismo código repetido varias veces. Por ejemplo, para los artículos que tienen texto (cambia sólo el nombre), imagen (sólo cambia la ruta), ...

Hay unos `includes` que tienen el código común a todos pero se les pueden cambiar las palabritas, las partes de código, que cambian.

Copiamos el código del artículo en otro archivo en `partials`, por ejemplo `product.html`.

En las partes que quiero que cambien en el artículo les pongo por ejemplo: @@name, donde cambiaría (ya sea dentro como etiqueta h3, p, como en el href de un enlace, o en el src de una imagen o su alt). Les pongo dos @ y una indicación para saber el qué tengo que poner.

Y, cuando ponemos el include en el index.html lo llamamos donde el src, después de éste (del include, como si fuera una clase después del src) y ponemos: name="Camiseta" y en el index.html de public me crea el artículo con nombre "camiseta".