

Por ahora solo ha pedido instalar node js y el vscode que ya teníamos.

Veremos según el profe distintos frameworks de front end.

Estructura Web

El DOM va a ser el archivo donde se estructura una pagina web.

- Comentario: JavaScript ya contiene todo lo que puede hacer JQuery, ya no debería usarse pero aun así se usa a veces.
- La estructura de una página web se divide en:
 1. Cabecera: HEAD. lo que no se ve, metadatos de la web.
 2. Cuerpo. BODY. lo que sí se ve, con etiquetas rollo imagen, text, etc.

JQuery

Funciona de la siguiente manera. Defino una etiqueta, una caja de html, y la cojo usando el `$()`. Si la etiqueta es img con piquitos por ejemplo pues sería `$(img)`.

Gestión de eventos, se hae un poco como flask. Hay eventos como click y otros métodos a los botones y ya le llamo de alguna forma a la función para gestionar lo que hace el botón.

- Llamadas ajax: son funciones javascript asíncronas. Cuando hay que meter opciones algo más especiales de que se ejecute algo en algún momento. Añade funcionalidades para comunicarse con el BackEnd.
- Angular en realidad lo que hace es una llamada ajax gigantesca. Se carga toda la página web, con sus distintas páginas html, en una sola y a partir de ahí cada página la carga sola sin necesidad de rellamar al backend.

Desde la consola del navegador podemos buscar clases. por ejemplo `$('body')` nos devuelve todo el body o cualquier otra cosa que tenga etiqueta.

Desde la página flukeout hay juegos para aprender css. Está muy bien para coger sintaxis.

si ponemos en google jquery cdn viene la linea que hay que poner en el HEAD. Se puede poner en el body antes de donde o usemos, pero así es como se recomienda.

A cada input type le puedo poner un NAME para identificar la etiqueta que es.

para escribir código javascript dentro de nuestra página, usamos la etiqueta script. Haré la comprobación cuando pinche en el botón.

Angular.

jerarquía de archivos:

- src/environments. Sirve para cambiar directivas del entorno de producción al entorno de desarrollo. Aquí por ejemplo no e usan acceso a base de datos, eso sería de back-end y no lo hace angular. Se puede hacer acceder a info del dispositivo pero no a una BD normal.

- package.json sirve para configuración de dependencias y otras configuraciones.
- main.ts es el que va a arrancar nuestra aplicación angular, en este caso con AppModule. no se toca casi nunca.
- el index.html. este es importante, dentro del body tiene la etiqueta app-root. es el componente principal. la idea es la misma que con JQuery, cambiar nuestro código html a pelo. cambiamos el comportamiento de app-root y esto se ve en el index.
- el appmodule.ts lo que tiene es la declaración del modulo. se declaran los componentes que utilizamos en los import. el providers para los servicios, y el bootstrap que es el componente de arranque, el AppComponent.
- Si vamos a AppComponent.ts tiene dentro del @Component el selector, una hoja de estilo, un template html y un titulo.

Superconsejito del día: nunca crear un componente a mano, sobretodo si se esta empezando. crearlo desde el cli de angular. *ng generate component < nombre >*

Vamos a crear un componente nuevo, con menu de navegación, otro que nos muestre la lista de empleados, y otro con un footer.

Para hacerlo, tras crear los componentes, hay siempre un html y un ts. en el ts cambiamos el modelo de la clase y en html su presentación. luego e el appcomponent.html metemos los selectors que haya que meter y palante.