1. ¿Has utilizado algún framework para desarrollo de frontends (Vue, React, Angular, etc.)? En caso afirmativo ¿cuál/es y con cuál finalidad acádemica o profesional? Brindar una descripción macro del proyecto y responsabilidades.

En mi trabajo me están capacitando en Angular actualmente, para luego poder utilizarlo en proyectos. Por lo tanto, los únicos desarrollos que realice en este framework fueron prácticas correspondientes a la capacitación.

1. ¿Experiencia con Javascript ES6?: Arrows, let + const, map + set, Proxies, etc.

Tengo experiencia con Javascript ES6, realice un curso en Udemy y luego utilizamos Javascript para desarrollar un proyecto en la universidad.

1. ¿Experiencias con consumo de RestApis? En caso afirmativo, ¿utilizando cuáles librerías/lenguajes y con cuál finalidad académica o profesional? Brindar una descripción macro del proyecto y responsabilidades.
2. ¿Experiencia con distributed version control systems? ¿Con cuáles?

Si, tengo experiencia con Git, utilizando Github y Gitlab.

5 - Ejercicio: Dado las funciones JavaScript que retornan promises request1() y request2(). Para todos los casos tener en cuenta que ambas funciones pueden lanzar excepciones y estas deben ser tratadas.

1- Escribir una función callRequests1 que realicen lo siguiente: -

- Llamar a request1() y esperar a que termine su ejecución

- Llamar a request2() y esperar a que termine su ejecución

- Escribir por consola ambos resultados

async function callRequests1() {

try {

const result1 = await request1();

const result2 = await request2();

console.log(result1);

console.log(result2);

} catch (error) {

console.error(error);

}

}

2-Escribir una función callRequests2 que realicen lo siguiente:

- Llamar a request1()

- Llamar a request2()

- Escribir por consola ambos resultados ni bien se obtengan.

function callRequests2() {

Promise.all([request1(), request2()])

.then((results) => {

console.log(results[0]);

console.log(results[1]);

})

.catch((error) => {

console.error(error);

});

}

3-Escribir una función callRequests3 que realicen lo siguiente:

- Llamar a request1()

- Llamar a request2()

- Escribir por consola ambos resultados cuando ambos llamados concluyan.

function callRequests3() {

let result1, result2;

request1()

.then((res) => {

result1 = res;

if (result2) {

console.log(result1);

console.log(result2);

}

})

.catch((error) => {

console.error(error);

});

request2()

.then((res) => {

result2 = res;

if (result1) {

console.log(result1);

console.log(result2);

}

})

.catch((error) => {

console.error(error);

});

}

6-Ejercicio:

Se solicita escribir las class contenedor y interno de tal forma que los div internos se acomoden según el Breakpoint:

- En XS: debe mostrar una imagen por fila

- En SM o MD: 3 por fila

- LG o más: 4 por fila

Además:

- El contenedor debe expandirse horizontalmente al ancho total de la página

- Los internos deben tener un alto de 100px fijo

- Los div internos deben expandirse horizontalmente hasta ocupar todo el ancho del contenedor.