* Pregunta de == y === :

Ojo: La respuesta lo escribo desde la perspectiva de Javascript.

Cuando se realiza la comparación del operador == es cuando tienes tipos diferentes, el sistema lo convierte en comparables con tal que sea trascendible en dicha comparación. Un ejemplo sería:

5 == ‘5’, regresa un verdadero (o true) o 0 ==”” regresa un verdadero

En cambio, un === compara de distinta forma, pues es más rigurosa:

10===’10’ regresa un falso, pues serán los mismos valores, pero la diferencia radica que los tipos son distintos, pues uno es un número y el otro es un string.

* Diferencias entre let, var y const :

-Respecto a const viene a ser una constante, pues al igual que las matemáticas, al asignar como const, este no varía el valor en nuestro programa. Ejm:

const Y =4 , donde Y tendrá el valor de 4 y no va a cambiar.

-El let puede cambiar su valor, pero si se declara en una parte del programa, pues solo funcionará en dicha parte. Ejm en un for:

function f5(){for (let j=0; j< 30;j++){ } }

-El var puede cambiar su valor, pero es a nivel global y también en el local.

* Habilidades para ser un buen Front End

Para ser un buen front end se necesita tener conocimientos de HTML, CSS y Javascript. Pues los 2 primeros sirven para el maqueteo Web. Javascript es fundamental, pues para darle el funcionamiento se necesita saber eventos y también una habilidad para programar. Al tener una buena base para programar en Javascript, se recomienda saber React, pues este framework facilita el trabajo del funcionamiento de las páginas web.

* CSS flex box

Si lo utilicé, desde mi perspectiva es de fácil manejo para acomodar los objetos

* Diferencia entre: function Person(){}, var person = Person(), and var person = new Person()

- function Person(){}= Este es una función y no se ejecuta.

- var person = Person()= Este es una variable llamado person y se le asigna al valor de la función Person, pues este devuelve un valor.

- var person = new Person()= Se crea una instancia del objeto constructor de Person, esto es añadiendo la palabra new

* Diferencia entre: forEach() y map()

El método map devuelve elementos transformados y la cantidad de datos no varía, mientras que forEach es un método mutador.