Javascript

BASIC FOUDATIONS-

So, there’s like two different value, it’s either object or primitive, primitive data types are number, string, Boolean, undefine, symbol, null, big int.

In javascipt we don’t need to insert the type of the data that we input, javascript has a built-in operation.

**2nd try:**

* Use ‘use strict’; at the first sentence in your code to prevent bug and lessen future problems
* Think if you should use let or const at the first place.
* Incrementing or decrementing had its own space code.
* Using addition in coercion will just add the number like the string, using minus while doing the string will minus it, while comparison syntax works
* First if, second if else, third else. It’s necessary
* Use === for greater result
* && and, || or
* Using switch statement, just add another case for different statement and don’t forget to add break and default for else statement.
* Using ternary operation we need a variable first then the statement and ? then proceed to else statement using : indicator.
* Calling a function is different from logging a console
* After making a variable inside a function don’t forget to return it.
* Using push in looping arrays is very important in the end of execution
* Looping backward don’t use hard code, we use length instead of literal 4. And you can loop inside a loop.
* While loop is just basically for loop the only difference is the variable is outside of a loop. And don’t forget to add the incrementation in the end of the loop.
* On if statement you actually don’t need to add bracket on execution
* Ctrl + d will detect the same word and copy it.
* If you want to solve something, first you have to understand the question first then make the task mini, and divide them individually.
* Use MDN documentation, stackoverflow and google
* Refactoring can make your code cleaner. And give your best avoiding repetitive code.
* When you only have one like if statement or for loop, you don’t actually need to add bracket on it.
* When you making a single line statement just use ternary operator and arrow function therefore you can save multiple line of code.
* Go in diagrams.net to design a flow chart
* getElementById is faster than querySelector
* EL means dom elements
* *Just skip the element to get the value on the array*
* Selecting the element needs a dot but not in editing the classList.
* Always make a variable for having an event so it’ll be much easier to solve things.
* Init – initialization (reusable code)
* You can access the variable inside the function using var.
* Hoisting will not work on function expression and arrow function.
* Var is the only one who create properties on the window object.
* In strict mode it will show you the undefined on using this on normal function.
* Arrow function do not get their own this keyword.
* Holy sheet var is dangerous xD.
* Using Destructuring can save your mind, when targeting the complicated value inside the object or on array.
* Spread operator a bit similar to destructing, the different is spread operator get all values from the array and doesn’t make any new variable
* Iterables are strings, sets, maps, array but NOT objects
* Passing a function or building a new array, the spread operator can only do.
* In set there’s no index so targeting by using index will not work, but also good to use arrays because you can convert it and arrays are applicable to anything.
* Working with strings, string is also a 0 based index, we used slice to start from start or from end.
* Convert the string first so things can be easily solved.
* Javascript does not have passing by reference, only passing by value
* There’s no really low-level function, it’s just depended on how they’ve been used.
* Call back function allowed us to make an obstruction,
* Arrow function don’t need parenthesis on parameter, no curly brackets and no return syntax.
* This keyword will work depend on how the function have been called.
* Whenever you are returning another function make sure you use bind so everything will work just fine.
* It’s always good to pass the function to variable and not put into global.
* Using function as a returned method is much modern than looping.
* Return keyword will be helpful to trying solve undefined.
* Don’t overuse Chaining Methods in making a real-world huge array.
* You can use underscore for undefined or unnecessary parameter of a callback function.
* It’ll be impossible for us to make a calculation number in a javascript using a number.
* Call a function that’s outside of a function, so it will make a good performance.