**JS MILESTONE 4**

Statements

Conditional Statements:-

● The if Statement

● If Else

Looping Statements:-

● The do-while Statement

● The while Statement

● The for Statement

● The for-in Statement

Other Statements :-

● The break and continue Statements

● The with Statement

● The switch Statement

1. Conditional Statements:

* If 🡪 to specify a block of code to be executed, if a specified condition is true
* else 🡪 to specify a block of code to be executed, if the same condition is false
* else if 🡪 to specify a new condition to test, if the first condition is false
* SYNTAX: if(condition) {};
* If is case sensitive
* SYNTAX: if(condition){};

else if (condition){};

else{};

1. Looping statements:

* If you want to run same code over and over again each time with a different value loops are used.
* *for* - loops through a block of code a number of times
* *for/in* - loops through the properties of an object
* *for/of* - loops through the values of an iterable object
* *while* - loops through a block of code while a specified condition is true
* *do/while* - also loops through a block of code while a specified condition is true
* FOR 🡪 for (*expression 1*;*expression 2*;*expression 3*) {  
    // *code block to be executed*  
  }
* Expression 1 is executed (one time) before the execution of the code block.
* Expression 2 defines the condition for executing the code block.
* Expression 3 is executed (every time) after the code block has been executed.
* In JS, expression 1 is optional. We can initiate many values in expression 1 separated by comma.
* Expression 1 can be omitted when we initialise it at the starting of the loop.
* let i = 2;  
  let len = cars.length;  
  let text = "";  
  for (; i < len; i++) {  
    text += cars[i] + "<br>";  
  }
* Expression 2 is also optional. If exp 2 is true, then the loop will start over again. If it is false, the loop will end.
* If exp 2 is omitted then *break* must be given.
* Exp 3 can also be omitted. Often it increments the value of initial value
* It may increase/ decrease/ i+=15 and so on
* If it is omitted then values must be incremented inside the loop.
* **For in** loops through the properties of an object.
* EX: const person = {fname:"John", lname:"Doe", age:25};  
  let text = "";  
  for (let x in person) {  
    text += person[x];}
* Each iteration returns key. The key is used to access the value of the key.
* This can also be used in arrays.
* If index value of array is important, then instead of for-in use Array.forEach()
* EX: const numbers = [45, 4, 9, 16, 25];

let txt = "";

numbers.forEach(myFunction);

function myFunction(value, index, array) {

txt += "Index: " + index + ", Value: " + value + ", Array: " + array + "\n";

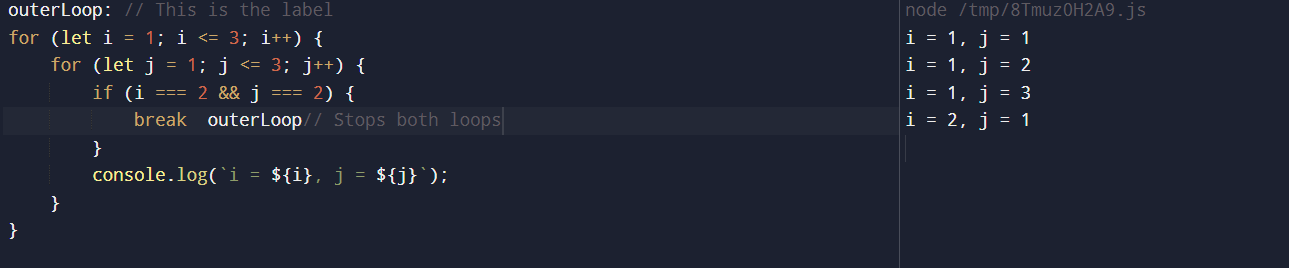
}

console.log(txt);

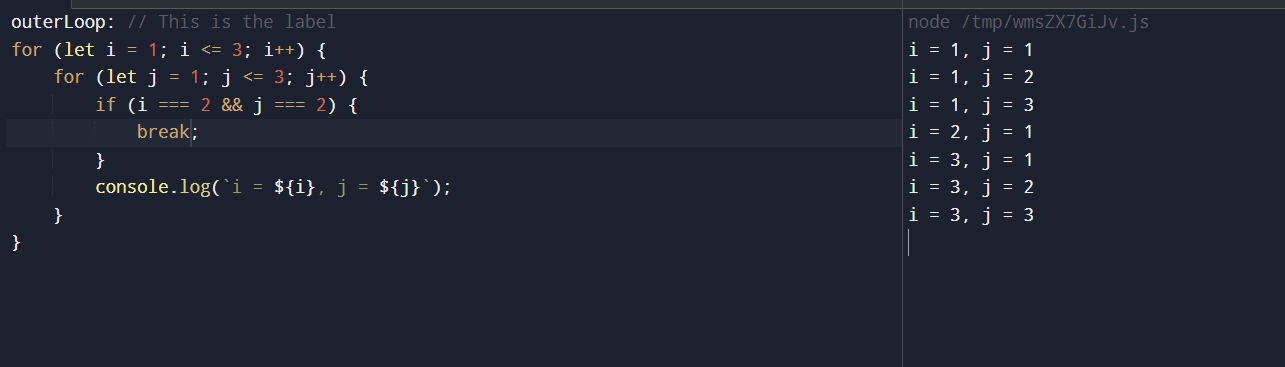
* Index returns index value, value returns value in the index, array return the whole array.
* **For-of** loops through the values of an iterable objects.
* For-of can be used over variables such as arrays, strings, maps, nodelists.,
* For-in is best suited for objects, for-of best suited for arrays, string, maps etc.,
* **While** loop – loops through a block of code as long as specified condition is true.
* while (condition) {  
  *// code block to be executed*}
* **Do while** – this loop will execute the code block once, before checking if the condition is true.
* do {*// code block to be executed*}  
  while (condition);
* variable must be increased in the loop for both while and do-while. Else the loop will never end.

1. Other Statements:

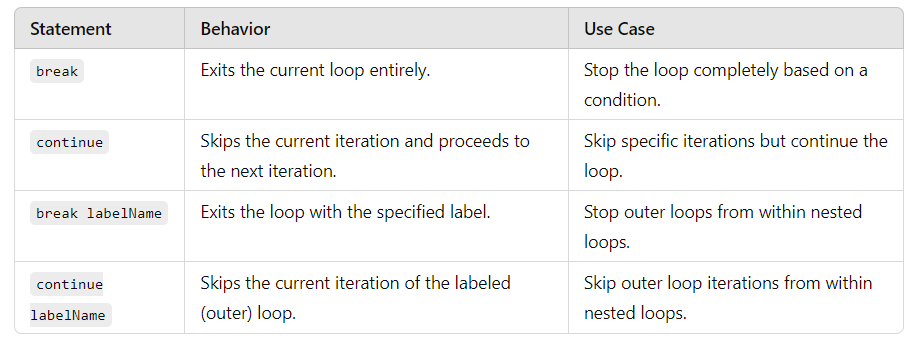
* Break – jumps out of a loop/ switch statement.
* Continue – jumps over, ie., breaks one iteration in the loop
* Label – it is name given to loops. So it is easy to control them easily.
  + Break with label – it stops specific loop that is named.



* As “break outerLoop” is given it ends the outerLoop even if the innerLoop is not completed.



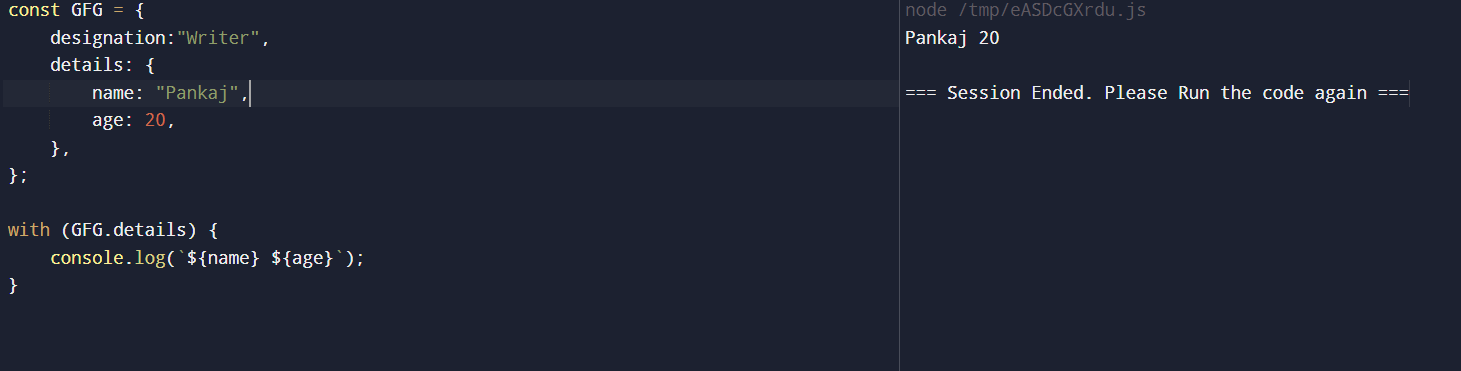
* As break is only given, the inner loop is only skipped.
* *break* only exits the inner loop and resumes the outer loop after that.
* *continue outerLoop* immediately skips the rest of the outer loop iteration and moves to the next iteration of the outer loop

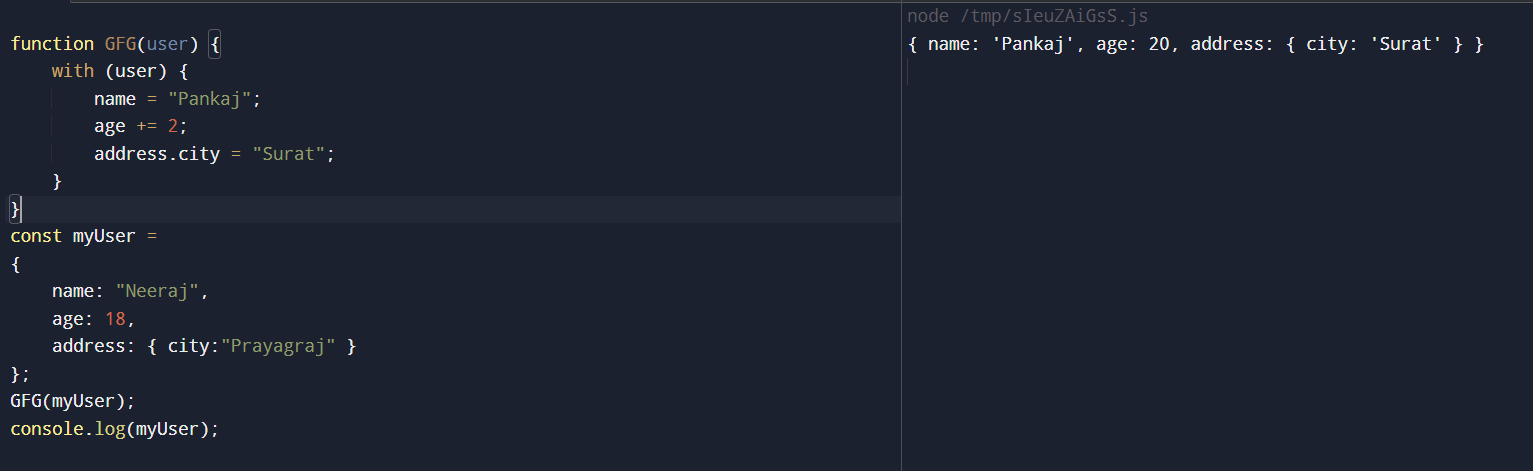


* With statement
* Syntax:
* with (expression)

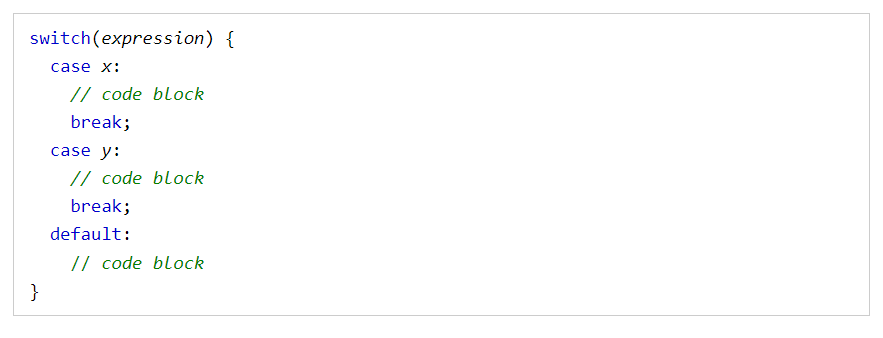
statement

* Simplified access to object properties.
* It is depreceated in most of the browsers due to problems like scope ambiguity, performance impact, security vulnerabilities.
* With can be used to access properties of object and modify properties of an object.

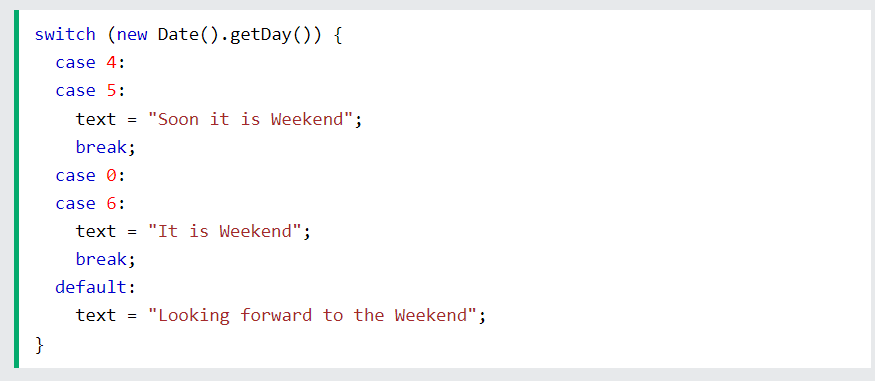




* Switch statement – used to perform different actions based on different conditions.
* Switch statement is used to select one of many code blocks t be executed.



* The default keyword specifies the code to run if there is no case match:
* Default keyword needs not to be mentioned at the end of the statement. It can also be at the top of the statement.
* Common code blocks can also be mentioned



* If multiple cases matches a case value, the first case is selected.
* If no matching cases are found, the program continues to the default label.
* If no default label is found, the program continues to the statement(s) after the switch.