Basic Problem Solving: Conditional Statements, Iterative Statements.

Relevel
by Unacademy

Educator Introduction (5 mins)

Please recap the following concepts taught in previous session

- Intro to Programming
- Intro to JS
- Why JS
- Intro to Mozilla Firefox multi-line console
- Hello World
- console.log()
- Variables
- Operators
- Coercion
- Datatypes

List of Concepts Involved (5 mins)

What are Conditional Statements

- if/else-if/else constructs
- switch case
- ternary operators
- Comparison between the different conditional constructs.

What are Loops/Iterative Statements

- for loops
- while loops
- do-while loops
- Comparison between the different iterative constructs.

Jump Statement - break and continue

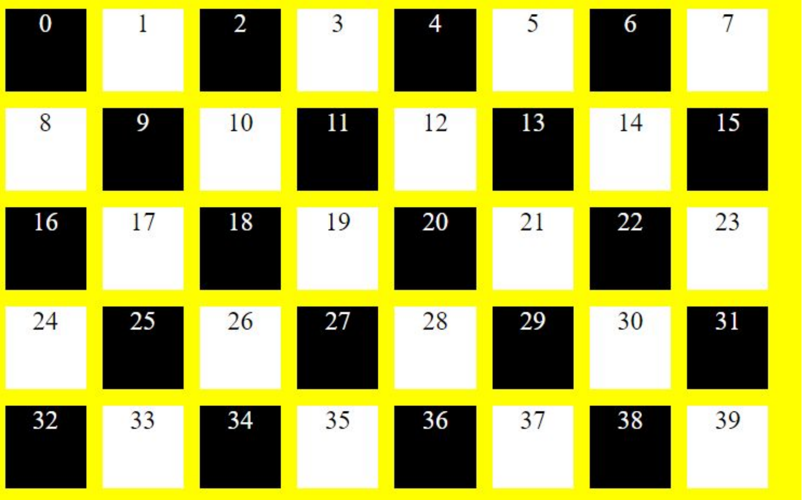
- Type Coercion

Application Introduction (5 mins)

We will be creating a chessboard

We will be solving a coding question having concepts of iterative,conditional statements.

Chessboard UI

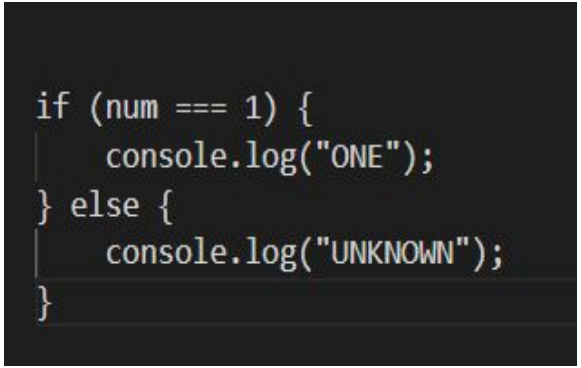


0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

Conditional Statements (45 mins)

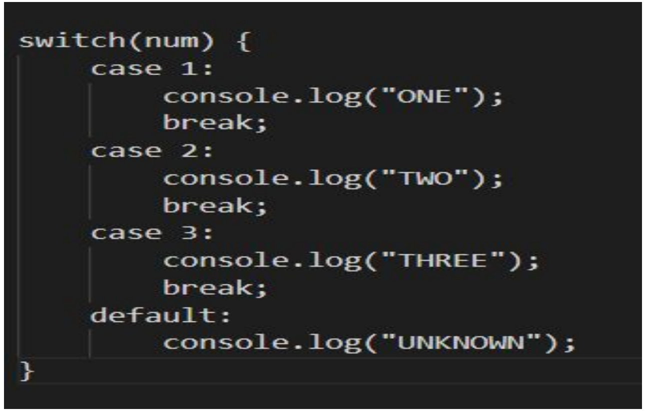
Conditional Statements helps in controlling behavior in javascript and determine which code to execute.Will go through different types of conditional statement .

Eg: If else statement



```
if (num === 1) {
    console.log("ONE");
} else {
    console.log("UNKNOWN");
}
```

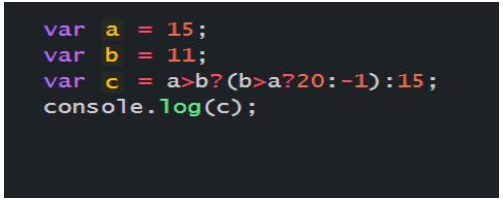
Eg: Switch Statement



```
switch(num) {
    case 1:
        console.log("ONE");
        break;
    case 2:
        console.log("TWO");
        break;
    case 3:
        console.log("THREE");
        break;
    default:
        console.log("UNKNOWN");
}
```

Try this question

What is the output of the following



```
var a = 15;
var b = 11;
var c = a>b?(b>a?20:-1):15;
console.log(c);
```

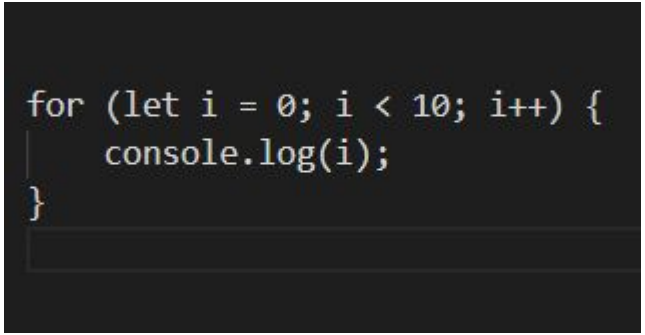
Answer: 1

Explanation: As a>b so it will enter the second ternary expression and the second condition turns out to be false so -1 will be assigned to c

Iterative Statements (45 mins)

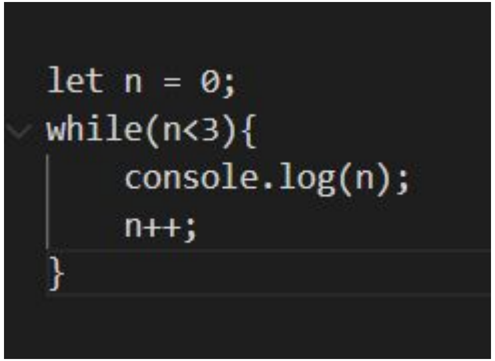
Iterative statements offer a quick and easy way to do something repeatedly.They come in handy when you want to run the same code over and over again each time with a different value.There are many different kind of iterative statements but they all do the same thing repeating an action some number of times.

Eg: For loop



```
for (let i = 0; i < 10; i++) {
    console.log(i);
}
```

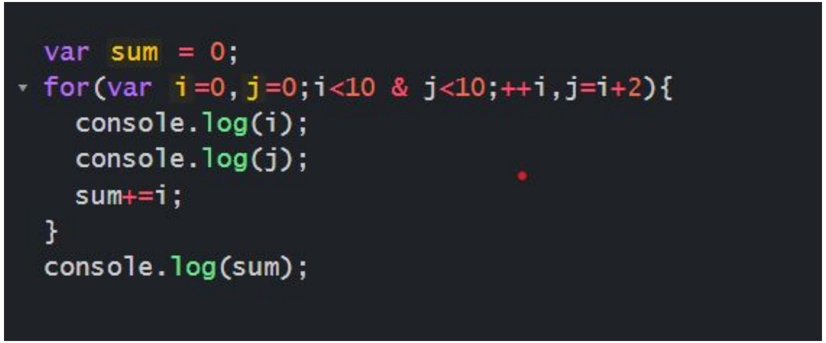
Eg: While loop



```
let n = 0;
while(n<3){
    console.log(n);
    n++;
}
```

Try this question:

What is the output of this question



```
var sum = 0;
for(var i=0,j=0;i<10 & j<10;++i,j=i+2){
   console.log(i);
   console.log(j);
   sum+=i;
}
console.log(sum);
```

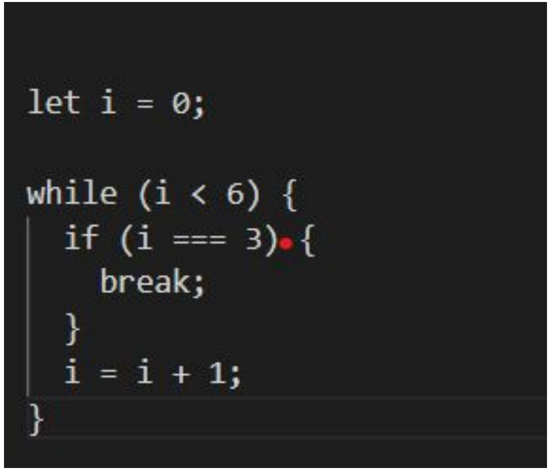
Answer: 28

Explanation: For loop will continue to increase i value from 0 to 8 and when the value of i becomes 8 then j will have 10 as its value which will make loop break,so variable sum will have the summation of i’s value from (0 to 7)

Jump Statements (20 mins)

Jump statements cause an unconditional jump to another statement in the code. They are used to break flow of switch statements and loops

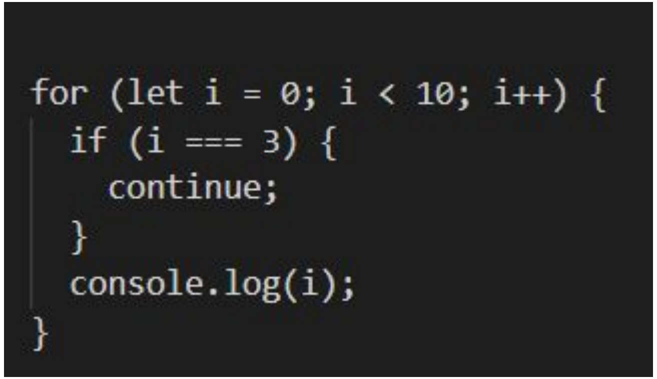
Eg: Break Statement



```
let i = 0;

while (i < 6) {
  if (i === 3) {
    break;
  }
  i = i + 1;
}
```

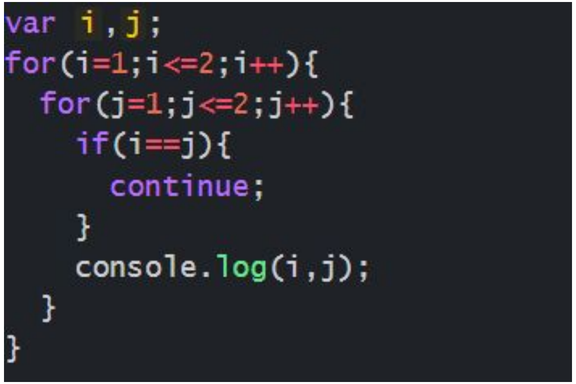
Eg: Continue Statement



```
for (let i = 0; i < 10; i++) {
  if (i === 3) {
    continue;
  }
  console.log(i);
}
```

Try this question

What is the output of this question?



```
var i,j;
for(i=1;i<=2;i++){
  for(j=1;j<=2;j++){
    if(i==j){
      continue;
    }
    console.log(i,j);
  }
}
```

Answer: 1,2;2,1

Explanation: When the value of i==j if condition becomes true which makes it to execute the continue statement taking the execution to the last line of second nested loop

Type Coercion

Type Coercion refers to automatic conversion of values from one data type to another data type. For **Example -**

- Number to String data type
- String to Number data type
- Boolean to Number data type and so on.

Type conversion takes place when we apply some operator on the operands or values.

Number to String Conversion

When we add any string or non-string value to a string, it will convert non-string value to string automatically.

Example -

var x = 1 + ‘2’ -> return 12

var y = ‘2’ + 1 -> return 21

var z = false + ‘Hello’ -> return falseHello

String to Number Conversion

When we perform any operation using operators, any non-number value will be automatically converted to number. Operators can be ‘-’, ‘*’, ‘/’, ‘%’

Example -

var x = 10 - ‘2’ -> return 8

var y = 10 * ‘2’ -> return 20

var z = 10 % ‘5’ -> return 0

Boolean to Number Conversion

When we add any boolean value to a number, the boolean value will be converted to number. false will be converted to 0 and true will be converted to 1.

Example -

var x = true + 20 -> return 21

var y = false + 20 -> return 20

Equality Operator

When we compare two values using the equality operator, it will perform type coercion first and then perform comparison. If we are comparing 1 number and 1 non-number then non-number will be converted to number first and then comparison will take place.

Example -

10 == ‘10’ -> return true

false == 0 -> return true (false converted to 0 here)

Practice Question

1)Program to check if a number is a perfect square

2) Program to check if a number is odd or even

Upcoming Class Teaser

- What are functions
- How to write a function
- Invoking a function
- Terminologies involved - parameter list, function body, function signature, return
- Uses/Advantages of using functions
- Hoisting of functions
- What is Scope
- Global Scope
- Function Scope
- Block Scope

Thank you