

Switch-case Loops



IT TALENTS
Training Camp

Problems with if-s

- ▶ What if we need to compare if a variable holds one of 20 possible values ? 20 If-s ?
- ▶ What if we need to execute the same statement when the variable is equal to three different values among those 20 ? Same code executed on several lines ?

The 'switch - case' statement

- ▶ Selects for execution a statement from a list, depending on the value of the switch expression
- ▶ The expression is evaluated
- ▶ When one of the constants specified in a case label is equal to the expression, the statement that corresponds to that case is executed
- ▶ If no case is equal to the expression
 - If there is a default case, it is executed
 - Otherwise the control is transferred to the end of the switch statement

```
let day = 5;

switch(day){
  case 1 :
    console.log("Monday");
    break;
  case 2 :
    console.log("Tuesday");
    break;
  case 3 :
    console.log("Wednesday");
    break;
  case 4 :
    console.log("Thursday");
    break;
  case 5 :
    console.log("Friday");
    break;
  case 6 :
    console.log("Saturday");
    break;
  case 7 :
    console.log("Sunday");
    break;
  default:
    console.log("No such day");
    break;
}
```



‘switch - case’ good practices

- ▶ Recommended variable types are **String** and **Number**
- ▶ Only discrete values supported in cases comparisons
- ▶ **break** is always a good idea to be used
- ▶ **default** is always a good idea to be used
- ▶ Multiple cases can execute a single statement
- ▶ Always handle the most probable cases first



Problem - print all the numbers

- ▶ From 1 to 5
- ▶ From 1 to 1000
- ▶ From 1 to n
- ▶ From n to m



What is a loop?

- ▶ A loop is a structure that allows a sequence of statements to be executed more times in a row
- ▶ Loops have a boolean condition and a block of code for execution. While the condition is true, the block is being executed.
- ▶ A loop that never ends is called an infinite loop



Why we use loops ?

- ▶ With loops we can execute similar statements many times
- ▶ We gain benefits from the code reuse
- ▶ Our code becomes much, much simpler

While loop

- ▶ The while loop is the simplest type of loop in Javascript.
- ▶ However that is not to say that it's not powerful.
- ▶ The basic syntax of the while loop is

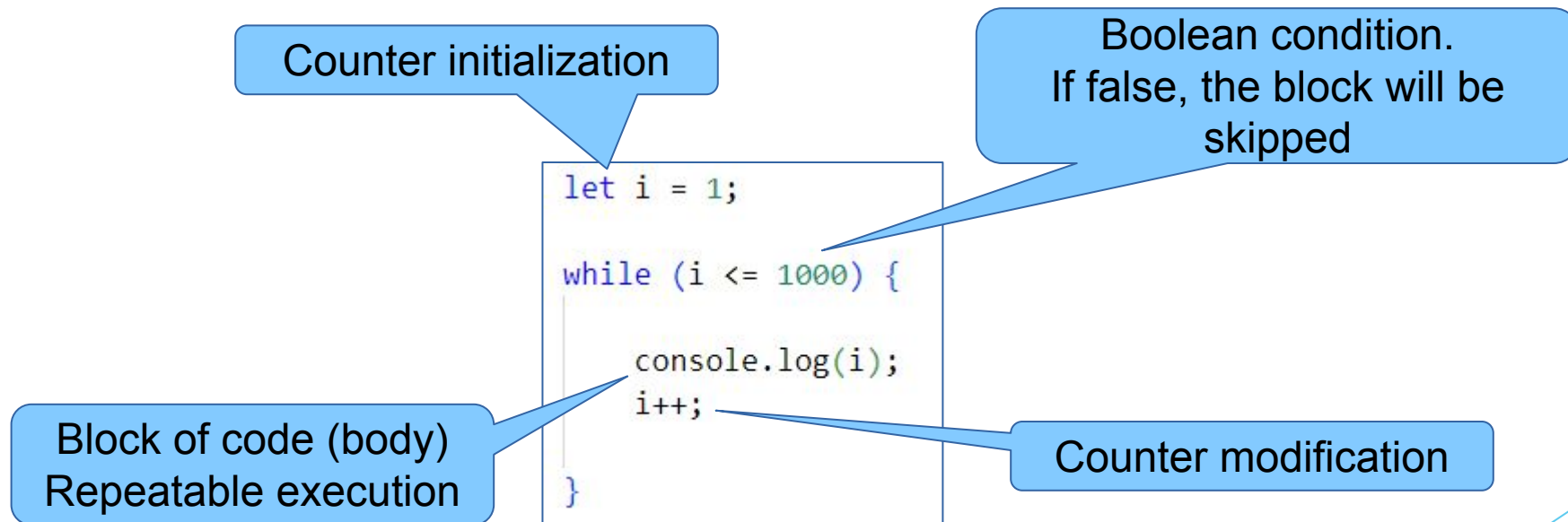
```
while( condition ){  
    statement;  
    statement;  
}
```

While loop executes
the block of code
while the condition is true

- ▶ One execution of the block of code of the loop is called an **iteration**

While loop

- ▶ While the condition is true, the block is being executed.



While loop simple tasks

- ▶ Print „Hello World“ in the console - 10 times
- ▶ Print the numbers from 1 to 100



Do-While loop

- ▶ The do-while loop is similar to the while loop
- ▶ With a do-while loop the condition is evaluated at the end of the iteration.
- ▶ The loop expressions will be executed at least once

```
do {  
    statement;  
    statement;  
}while( condition );
```



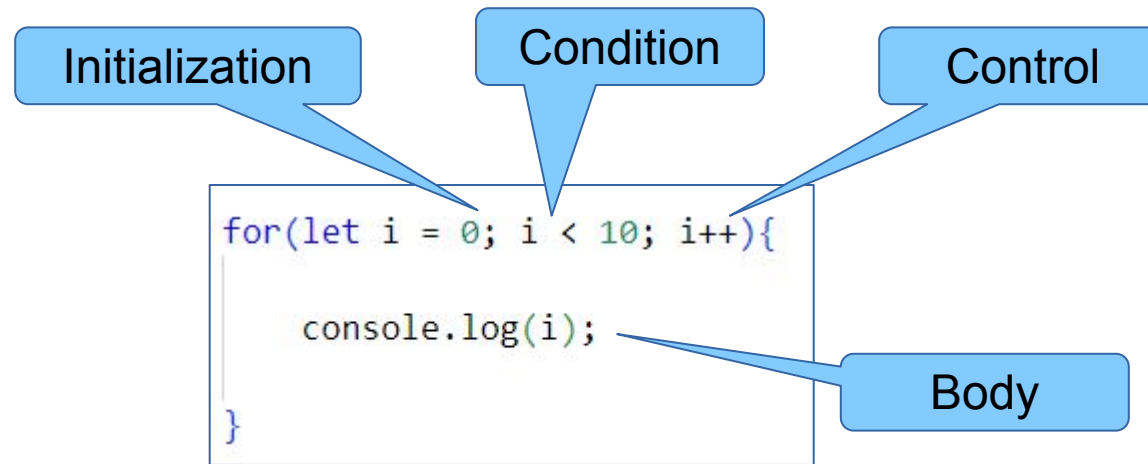
For loop

- ▶ The for loops are another commonly used loop
- ▶ There are three important expressions which make the magic

```
for( init_expr; condition_expr; control_expr ) {  
    statement;  
    statement;  
}
```

For loop

- Consists of
 - Initialization
 - Condition
 - Update statement
 - Body



Nested Loops

- ▶ Loops could be nested in each-other
- ▶ We can embed loops of different kind
- ▶ There is no limit how deep we can go nesting

```
for(let i = 0; i <= 4; i++){  
  for(let j = 0; j <= 3; j++){  
    console.log(i + " , " + j);  
  }  
}
```



```
0 , 0  
0 , 1  
0 , 2  
0 , 3  
1 , 0  
1 , 1  
1 , 2  
1 , 3  
2 , 0  
2 , 1  
2 , 2  
2 , 3  
3 , 0  
3 , 1  
3 , 2  
3 , 3  
4 , 0  
4 , 1  
4 , 2  
4 , 3
```



Break keyword

- ▶ Problem - try to quit a for-loop during the execution of the repeatable block
- ▶ **break** is a keyword
- ▶ A statement by itself
- ▶ It doesn't require anything else
- ▶ It stops the execution of the loop

```
for (let i = 0; i < 50; i++) {  
  if (i === 7) {  
    break;  
  }  
}
```

The loop will end when I is
7

Continue keyword

- ▶ Problem - try to omit specific block of code in the body - for example sum all numbers between 1 and 100 but omit all numbers between 51 and 74
- ▶ **continue** is a keyword
- ▶ A statement by itself
- ▶ It doesn't require anything else
- ▶ It stops the current iteration of the loop, but doesn't stop the loop

```
let sum = 0;
for (let i = 0; i < 100; i++) {
  if (i > 51 && i < 71) {
    continue;
  }
  sum = sum + i;
}
```

If i is between 51 and 71
The loop will skip the
statements

Practice tasks

- ▶ Print all numbers between 1 and 100
- ▶ Print all numbers between 1 and -100
- ▶ Print all even numbers between 10 and 50
- ▶ Print all numbers between 5 and 55 that are divisible by 3 and 4
- ▶ Print all numbers between n and m , where n and m are defined by the user. The numbers displayed must always be in ascending order