



Web Engineering Front-end Pt. 3

7. JavaScript: Loops I

Presented by Krystal Educational Platform



Contents



1. while loops
2. do-while loops
3. break statements
4. continue statements

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute





7.1 while loops



What are loops?



In programming, we will often need to repeat similar actions. For example, if we wanted to output the numbers 1 to 10.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



What are loops?



```
console.log(1);
console.log(2);
console.log(3);
console.log(4);
console.log(5);
console.log(6);
console.log(7);
console.log(8);
console.log(9);
console.log(10);
```

What are loops?



In the previous example, the only thing that changes between each line of code is the number.

while loop



while loops allow us to repeat the same code as long as a condition returns True.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



while loop syntax



```
while (condition) {  
    // code to be run if condition is true  
}  
  
// program then returns to the beginning of the loop
```

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



while loop example

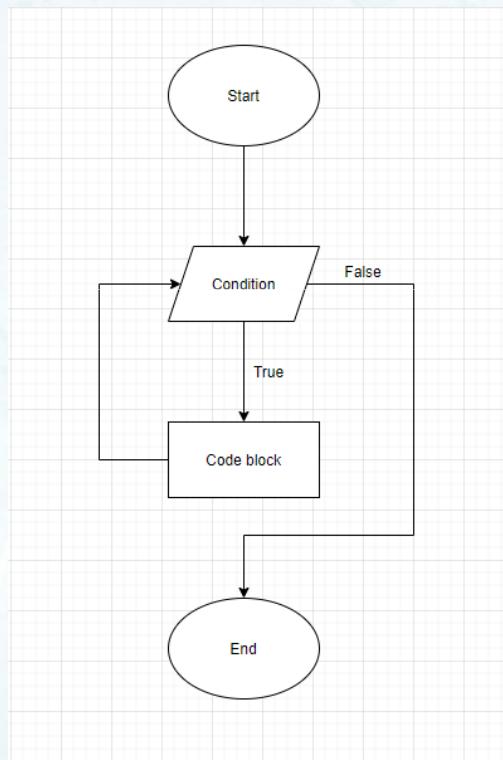


```
var i = 1;

while (i <= 10) {
    console.log(i);
    i++;
}
```

JavaScript: Loops I

while loop flowchart



while loops



What's wrong with the following code? The goal is to print “Hello, World!” five times.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



while loops



```
var i = 0;  
  
while (i < 5) {  
    console.log("Hello, World!");  
}  
}
```

Infinite loop



The value of i never changes, so the while loop will run forever since the condition is always true.

Infinite loop



Trying to execute an infinite loop will probably crash your browser. If you do accidentally create an infinite loop, forcefully close your browser ASAP.

Practice: while loops



Try writing some while loops of your own. Make sure you don't accidentally create an infinite loop.



7.2 do-while loops



do-while loops



do-while loop is very similar to while loop, but the condition is checked after the code is executed.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



do-while loop syntax



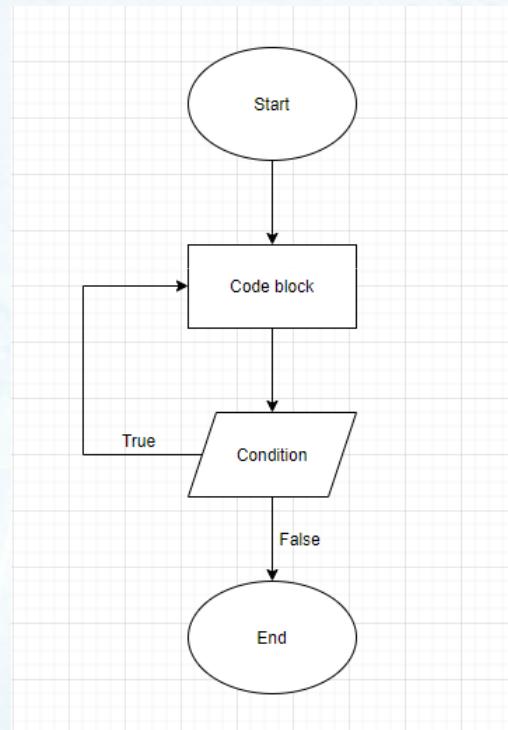
```
do {  
    // code to be run if it is the first iteration or if condition is true  
}  
    while (condition);  
  
    // program then returns to the beginning of the loop
```

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



JavaScript: Loops I

do-while loop flowchart



do-while loops



The code block will always be run at least once in a do-while loop. Unlike a while loop, where it may be skipped all together if the condition starts as False.

do-while loop example



```
do {  
    random = Math.floor(Math.random() * 10) + 1;  
    console.log(random);  
} while (random != 7);
```

do-while loops



do-while loops can be useful when a specific value is required, such as when asking for a user input.

do-while loops



```
do {  
    name = prompt("Enter a number greater than 10")  
} while (name <= 10);
```

do-while loops



In the previous example, the program will prompt for a user inputted number, and will continue to prompt until a number greater than 10 is entered.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



Infinite do-while loops



do-while loops can also be accidentally executed infinitely, so be sure to set the condition as something that will eventually happen.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



Infinite do-while loop example



```
var x = 0;  
do {  
    console.log(x)  
} while (x < 20);
```

Practice: do-while loops



Try thinking of some situations where a do-while loop would be useful and write it out as code.

7.3 break statements



break statements



Remember when we learned about switch statements?

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



break statement example



```
switch (day) {  
    case "Monday":  
    case "Tuesday":  
    case "Wednesday":  
    case "Thursday":  
    case "Friday":  
        weekend = false;  
        break;  
    case "Saturday":  
    case "Sunday":  
        weekend = true;  
}
```

break statements

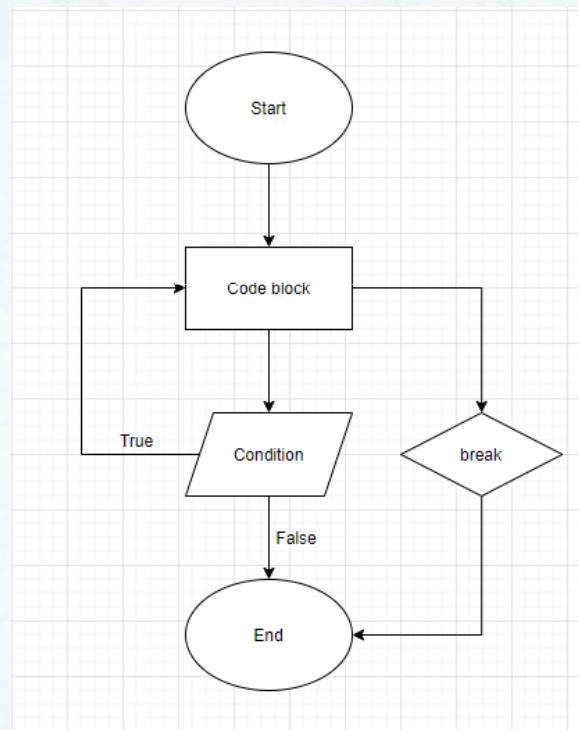


The break statement can be used in a loop to forcefully terminate it.

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



break statement flowchart



break statement example



```
var i = 0
```

```
while (i <= 10) {  
    if (i == 4) {  
        i++;  
        break;  
    }  
    console.log(i);  
    i++;  
}
```

```
// 0, 1, 2, 3
```

Practice: break statements



Try adding break statements to the while and do-while loops you wrote previously.



7.4 continue statements



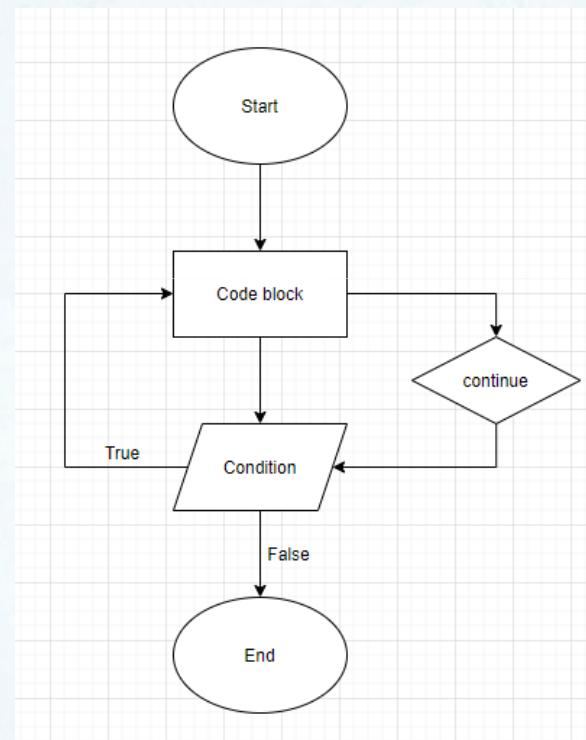
continue statements



The continue statement can be used to skip an iteration of a loop.

JavaScript: Loops I

continue statement flowchart



JavaScript: Loops I

Is this correct?



```
var i = 0

while (i <= 10) {
    if (i == 4) {
        continue;
    }
    console.log(i);
    i++;
}
```



Is this correct?



The continue statement restarts the loop, skipping `i++`. So this is an infinite loop, as `i` is stuck at 4.

JavaScript: Loops I

Is this correct?



```
var i = 0
```

```
while (i <= 10) {  
    if (i == 4) {  
        i++;  
        continue;  
    }  
    console.log(i);  
    i++;  
}
```

```
// 0, 1, 2, 3, 5, 6, 7, 8, 9, 10
```



JavaScript: Loops I

continue statement example



What's the output for the following code?

Copyright © Krystal Institute Ltd 2021
Do not copy or distribute



continue statement example



```
var i = 0

while (i <= 20) {
    if (i % 2 == 0) {
        i++;
        continue;
    }
    console.log(i);
    i++;
}
```

Practice: continue statements



Try adding continue statements to the while and do-while loops you wrote previously. Make sure not to accidentally create an infinite loop.



The End



References



Reference 1: JavaScript.info Loops: while and for <https://javascript.info/while-for>