Types, Values, Operators

Part 2

Repetitions . Loops

Loops help to

- repeat some code
- reduce code
- Find a generic solution

Repetitions . Loops

What if you need to print all numbers between 0 and 1000?

While

```
while (condition) {
        code block to be executed
}
```

Do While

```
do {
        code block to be executed
}
while (condition);
```

For

```
for (statement 1; statement 2; statement 3) {
    code block to be executed
}
```

Infinite Loops

Guess how can we create infinite loops

Infinite Loops

1. `for (;;) {}`
2. `while (true) { //your code }`

Breaking Out of a Loop

Sometimes you need to stop loops or terminates execution of the statements in the current iteration of the current or labeled loop, and continues execution of the loop with the next iteration.

For these cases we have ***break*** and ***continue*** operators.

Break

The **break statement** terminates the current loop, switch and transfers program control to the statement following the terminated statement

```
while (i < 5) {
 i++;

 if (i === 3) {
   break ;
 }

 n += i;
}
```

Continue

The **continue statement** terminates execution of the statements in the current iteration of the current or labeled loop, and continues execution of the loop with the next iteration.

```
while (i < n) {
 i++;

 if (i % n === 0) {
   continue;
 }
console.log(i);
 n += i;
}
```