

🚩 Current Skill Break and Continue

## Break & Continue

We are almost done with loops. There remain only two small constructs: **break** and **continue**.

We can use break or continue inside the loops:

- **Break** exits the closest loop it is in and continues with the next command.
- **Continue** skips and jumps out from the loop body, and jumps back to the condition of the loop.

*// Let's sum every second element of the numbers array:*

```
var numbers = [19, 65, 1, 2, 6, 1, 9, 9, 2, 1];
```

```
var sum = 0;
```

```
for ( var i in numbers ) {
```

```
    if ( i % 2 == 0 ) continue;
```

```
    sum += numbers[i]
```

```
}
```

```
console.log(sum);
```

```
// prints 78
```

```
/*
```

*PS: Notice that we could've written `i += 2` in a simple for loop to jump to every second element.*

*We're using continue here just for the sake of the example.*

*Whenever i is even, continue moves execution back to the next iteration of i in numbers.*

```
*/
```

```
/*
```

*You already know what a break statement looks like, because we learned it when dealing with if statements.*

*It is doing the same thing in loops. Suppose we want to break out from the loop when we find a number greater than 10.*

```
*/
```

```
var numbers = [19, 65, 1, 2, 6, 1, 9, 9, 2, 1];
```

```
var sum = 0;
```



```
for ( var i in numbers) {  
    sum += numbers[i];  
    if ( sum >= 100 ) {  
        break;  
    }  
}  
  
console.log(sum)  
  
// prints 103
```



[< Previous](#)

[next >](#)