CodingLab
Powered by RBK

Bootstrap

Iteration with For

For Loop

Duration: 30 minutes
Q&A: 5 minutes by the end of the lecture

While Loop

We used this syntax to create a `while` loop.

```
var x = 0;
while (x < 10) {
  console.log(x);
  x = x + 1;
}
```

Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

Creating for Loops in JavaScript

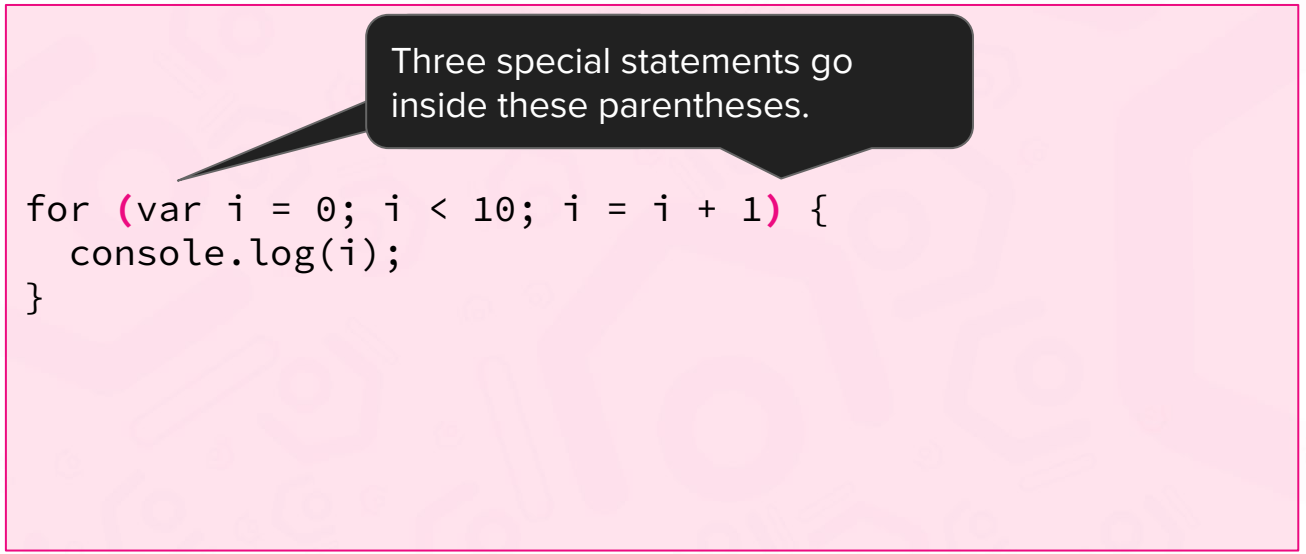
We can use this syntax to create a `for` loop.

The `for` keyword indicates the beginning of our loop.

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.



Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.

The first statement declares our counting variable.

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.

The second statement specifies the condition in which we should repeat as long it's met.

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.

The third statement moves us closer to that condition..

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

Curly braces denote the beginning and end of the code we want to repeat.

Creating for Loops in JavaScript

We can use this syntax to create a `for` loop.

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

This is the statement that will be repeated to print to the console the numbers from 0 to 9

While Loop

```
var x = 0;
while (x < 10) {
  console.log(x);
  x = x + 1;
}
```

For Loop

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

While Loop

```
var x = 0;
while (x < 10) {
  console.log(x);
  x = x + 1;
}
```

Declaration of our counting variable.

For Loop

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

While Loop

```
var x = 0;
while (x < 10) {
  console.log(x);
  x = x + 1;
}
```

The condition in which we should repeat the code as long it's met.

For Loop

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

While Loop

```
var x = 0;
while (x < 10) {
  console.log(x);
  x = x + 1;
}
```

The statement that moves us closer to that condition.

For Loop

```
for (var i = 0; i < 10; i = i + 1) {
  console.log(i);
}
```

```
function power_iter(base, exponent){
  var result = 1;
  while (exponent > 0) {
    result = result * base;
    exponent = exponent - 1;
  }
  return result;
}
```

Earlier we looked at the `while` loop and create a parametric power function using it. Let's see how to change it to a for loop.

```
function power_iter(base, exponent){
  var result = 1;
  while (exponent > 0) {
    result = result * base;
    exponent = exponent - 1;
  }
  return result;
}
```

Let’s change our while loop to increment instead of decrement

```
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (exponent > 0) {
    result = result * base;
    exponent = exponent - 1;
  }
  return result;
}
```

First we'll add a counting variable.

```
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (exponent > 0 i < exponent) {
    result = result * base;
    exponent = exponent - 1;
  }
  return result;
}
```

We'll change our while loop condition to be based on the the counter variable.

```
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (i < exponent) {
    result = result * base;
    exponent = exponent - 1;
    i = i + 1;
  }
  return result;
}
```

Let's change the code that will move us towards the exit condition to be based on increasing the counting variable instead of decreasing the exponent.

```
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (i < exponent) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

We've changed our while loop to use a counting variable, let's see how we can change it to a for loop

```
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (           i < exponent          ) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

```
  for(<declare counter>; <condition>; <adjust counter){
    // do this...
  }
```

Here's the syntax for a **for** loop. In one line, we declare a counter variable to keep track of our iterations, make a condition so our code knows when to stop looping, then an action to adjust our counter variable.

```
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  for   (           i < exponent          ) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

Let's refactor **power_iter** one more time to use a **for** loop in place of a **while** loop.

```
function power_iter(base, exponent){
  var result = 1;

  for   (var i = 0; i < exponent           ) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

Let's use `i` as our counter variable and move it into our `for` loop.

```
function power_iter(base, exponent){
  var result = 1;

  for   (var i = 0; i < exponent;          ) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

We already have our condition in place.

```
function power_iter(base, exponent){
  var result = 1;

  for   (var i = 0; i < exponent; i = i + 1) {
    result = result * base;

  }
  return result;
}
```

Finally we can move our adjustment to `i` into its appropriate place in our `for` loop.

```
//For Loop
function power_iter(base, exponent){
  var result = 1;
  for   (var i = 0; i < exponent; i = i + 1) {
    result = result * base;
  }
  return result;
}

//While Loop
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (i < exponent) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

Here are our two functions

```
//For Loop
function power_iter(base, exponent){
  var result = 1;
  for   (var i = 0; i < exponent; i = i + 1) {
    result = result * base;
  }
  return result;
}

//While Loop
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (i < exponent) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

Declaration of counting variable

```
//For Loop
function power_iter(base, exponent){
  var result = 1;
  for   (var i = 0; i < exponent; i = i + 1) {
    result = result * base;
  }
  return result;
}

//While Loop
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (i < exponent) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

Our stop condition

```
//For Loop
function power_iter(base, exponent){
  var result = 1;
  for   (var i = 0; i < exponent; i = i + 1) {
    result = result * base;
  }
  return result;
}

//While Loop
function power_iter(base, exponent){
  var result = 1;
  var i = 0;
  while (i < exponent) {
    result = result * base;
    i = i + 1;
  }
  return result;
}
```

Our action to reach the stop condition

That’s it