



Posted on Jan 5, 2016 by Kristian Woods

I Don't understand for loops in JavaScript

Hey, I'm not sure how to structure this question, but i'll try.

I'm working through JavaScript for loops.

I'm having an issue wrapping my head around the 'i' variable.

First off, I understand that 'i' starts at 0, checks to see if 0 is less than a specified number, runs the code block then adds 1 to the 'i' variable. Then this continues until 'i' is no longer less than the specified number.

I also know about using the .length property instead of using a specified number. This way the loop runs no matter how many items are inside the array.

What I have an issue understanding is when you use the 'i' variable as an index position, of sorts.

Example:

```
var fruit = ['Apple', 'Orange', 'Banana'];
for (var i = 0; i < fruit.length; i +=1) {
  console.log(fruit[i]);
}
```

I don't understand how the 'i' variable works inside the square brackets.

I tried just putting the 'fruit' variable inside the parentheses by itself, without the [i], and it printed off the array - quotes and comma's included - it printed it off 3 times. I don't understand why.

```
var fruit = ['Apple', 'Orange', 'Banana'];  
for (var i = 0; i < fruit.length; i +=1) {  
    console.log(fruit);  
}
```

When I went back and included the [i] within the parentheses. I checked the console and it had printed off the array as a list - without any quotes or comma's -

I just don't understand what the [i] is doing. Because, you still get the list, even if you don't include the [i]. So why use it?

Please help

Thanks

3 Answers



Anabelle Miacz on Jan 5, 2016

Hey. So as you guessed right. var (i) is set to zero at first then loops over each item in array adding 1 to (i) When i reaches the point where it's equal to (array.length) which is total number of items in array it stops.

Now to answer your question.

For each item in array (i++) You add 1 to i variable.

So behind the scene variable i = 0 i = 1 i = 2 i = 3 And so on

console.log(fruit[i]); ---> that produces (behind the scenes)

```
console.log(fruit[0]);
```

```
console.log(fruit[1]);
```

```
console.log(fruit[2]);
```

And of course. That's how you access stuff in array (By it's index)

```
ARRAY['hey', 'how', 'are', 'you']
```

hey is at index 0

how is at index 1

are is at index 2

you is at index 3

That said: For your array, variable (i) represents (number) which is an index in array

Now your example:

```
var fruit = ['Apple', 'Orange', 'Banana']; for (var i = 0; i < fruit.length; i +=1) { console.log(fruit[i]); }
```

First (i) is set to zero

console.log(fruit[0]) -> which is 'Apple' because it is at index 0 in the array

Second (i) is set to 1 -> which is 'Orange' because it is at index 1 in the array

Third (i) is set to 2 -> which is 'Banana' because it is at index 2 in the array

To give you an example of using (fruit) variable: Here's how you would do that using foreach loop

```
var fruits = ['apple', 'orange', 'banana'];
```

```
foreach(fruits as fruit){ console.log(fruit); }
```

Hope that this makes sense.



Chyno Deluxe on Jan 5, 2016

Hello Kristian,

The variable `i` is the index of the array you are iterating through. When you run your block of code, `i` becomes 0, runs the code, then 1 gets added to it making `i`'s new value = 1.

Each item in an array is given an index. `i` is an integer that cycles through all the items in the array, hence the use of **fruit.length**.

```
var fruit = ['Apple', 'Orange', 'Banana'];

/*
fruit[0] = Apple
fruit[1] = Orange
fruit[2] = Banana
*/
```

This insures the code will work regardless of how many items are in the array. However, in your second example, when you exclude the `[i]`. you are simply logging the array rather than the items in the array, which will repeat as many times as there are items.

I hope this helps. Let me know if you still have questions.



Kristian Woods on Jan 5, 2016

Hey, Anabelle and Chyno!

Thank you so much for your response. It really helped!



Posting to the forum is only allowed for members with active accounts.
Please **sign in** or **sign up** to post.



[Treehouse Blog](#)

[About Treehouse](#)

[Privacy Policy](#)

[Terms & Conditions](#)

© 2017 Treehouse Island, Inc.