

for-in loops

The for-in loop is specifically designed to work with objects. Learn how to loop through an object's keys in order to process each of them.

We've seen that for-loops are great for looping through arrays. There's another type of loop for objects, the for-in loop.

```
1 let jon = {  
2   firstName: 'Jon',  
3   lastName: 'Smith',  
4   age: 34,  
5   profession: 'Developer',  
6   married: true  
7 };  
8  
9 for(let key in jon) {  
10   console.log(key, ':', jon[key]);  
11 }
```



We start it out as a standard for-loop.

```
for() {  
    // body  
}
```

In the parentheses, we declare a variable using `let key`. After we declare the variable, we write the `in` keyword followed by the name of the object we want to work with.

Inside the loop, our variable `key` will be equal to the keys of the object `jon`. It'll go through each key, running our loop repeatedly until every property has been processed.

Note that we have to use bracket notation here because `key` is a variable and not a string. If we were to write `jon.key`, the loop would print `undefined` over and over

because it would be looking for a `key` property.



By using bracket notation, we can give it a variable instead.

Quiz#

for-in loops are often unintuitive for beginners. Practice will help it sink in. Feel free to test your understanding.

1 What will the following code print?

```
let jon = {  
  firstName: 'Jon',  
  lastName: 'Smith',  
  age: 34,  
  profession: 'Developer',  
  married: true  
};  
  
for(let key in jon) {  
  console.log(key + jon[key]);  
}
```



A)

```
firstName : Jon  
lastName : Smith  
age : 34  
profession : Developer  
married : true
```



B)

```
firstName Jon  
lastName Smith  
age 34  
profession Developer  
married true
```



C)



```
firstNameJon  
lastNameSmith  
age34  
professionDeveloper  
marriedtrue
```

Submit Answer



Question 1 of 2
0 attempted



Reset Quiz ↺

← Back

while-loops

Next →

Putting it all together



Mark as Completed



Report an Issue