

Objects



Overview

- 1 /*
 - what is an object?
 - why are objects useful?
- 2
 - typeof object
 - accessing, adding, changing, deleting values
 - in operator
- 3
 - for...in loop
 - Object.keys()
- 4
 - nested arrays and objects
- */

5

6

7

8

9

10

11



value1
value1

What is an object?

```
1  /* an object is a collection of key-value pairs */
2
3  /* like arrays, objects store values, but instead of storing them in
4     numeric "indices", objects store values in string "keys" */
5
6  let myArray = ['value1', 'value2'];
7
8  let myObject = {
9    'key1': 'value1',
10   'key2': 'value2'
11  };
12
13  console.log(myArray[0]);
14  console.log(myObject['key1']);
```



Why are objects useful?

```
1  /* consider representing Pusheen the Cat as an array */
2  let pusheen = ['Pusheen', 7, 'gray and tabby'];
3
4  /* an array is a good place to hold an ordered list of values, but it
5     doesn't store any information about what those values represent */
6
7  /* an object's string keys allows objects to store more information about
8     the values within it */
9
10 let pusheen = {
11   'name': 'Pusheen',
12   'age': 7,
13   'color': 'gray and tabby'
14 };
```



typeof object

```
1 console.log(typeof {});
```

2

3

4

5

6

7

8

9

10

11

12

13

14



Creating an object

```
1  /* create a new object using curly braces */
2
3  /* an object's keys are always strings; you can omit the quotation
4     marks */
5
6  let pusheen = {
7     name: 'Pusheen',
8     age: 7,
9     color: 'gray and tabby'
10 };
11
12
13
14
```



Pusheen
7
gray and tabby
undefined

Accessing a value

```
1  /* use bracket notation to access a value */
2
3  /* pass a string into the brackets that corresponds with a key in the object */
4  let pusheen = {
5    name: 'Pusheen',
6    age: 7,
7    color: 'gray and tabby'
8  };
9
10 console.log(pusheen['name']);
11 console.log(pusheen['age']);
12 console.log(pusheen['color']);
13 console.log(pusheen['notAKeyInTheObject']);
14
```



Accessing a value

```
1  /* any variable or expression that evaluates to a string can be passed
2     into the brackets */
3
4  let pusheen = {
5     name: 'Pusheen',
6     age: 7,
7     color: 'gray and tabby'
8  };
9
10 let keyToCheck = 'name';
11
12 console.log(pusheen[keyToCheck]);
13 console.log(pusheen['col' + 'or']);
14
```




Pusheen
7
gray and tabby
undefined

Accessing a value

```
1  /* you can also use dot notation to access values */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  let keyToCheck = 'name';
10
11 console.log(pusheen.name); // no quotes needed with dot notation
12 console.log(pusheen.age);
13 console.log(pusheen.color);
14 console.log(pusheen.keyToCheck);
```



Adding a key/value pair

```
1  /* use bracket notation or dot notation to add a
2     key/value pair */
3
4  let pusheen = {
5     name: 'Pusheen',
6     age: 7,
7     color: 'gray and tabby'
8  };
9
10 pusheen['sister'] = 'Stormy';
11 pusheen.brother = 'Pip';
12
13 console.log(pusheen);
14
```

```
{
  name: Pusheen,
  age: 7,
  color: gray and tabby,
  sister: Stormy,
  brother: Pip
}
```



Changing a value

```
1  /* use bracket notation or dot notation to change a value */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  pusheen['age'] = 8;
10 pusheen.age++
11
12 console.log(pusheen.age);
13
14
```



```
{ name: Pusheen }
```

Deleting a key/value pair

```
1  /* use the delete keyword to delete a key/value pair */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  delete pusheen['age'];
10 delete pusheen.color;
11
12 console.log(pusheen);
13
14
```



true
false

in operator

```
1  /* use the in operator to check if a key is in the object */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  console.log('name' in pusheen);
10 console.log('sadness' in pusheen);
11
12
13
14
```



name
age
color

for...in loop

```
1  /* use the for...in loop to loop through all of the keys in an object */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  for (let key in pusheen) {
10    console.log(key);
11  }
12
13
14
```



Pusheen's name is Pusheen
Pusheen's age is 7
Pusheen's color is gray and tabby

for...in loop

```
1  /* use the for...in loop to loop through all of the keys in an object */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  for (let key in pusheen) {
10    console.log("Pusheen's", key, 'is', pusheen[key]);
11  }
12
13
14
```



for...in loop

Pusheen's name is undefined
Pusheen's age is undefined
Pusheen's color is undefined

```
1  /* use the for...in loop to loop through all of the keys in an object */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  for (let key in pusheen) {
10    console.log("Pusheen's", key, 'is', pusheen[key]);
11  }
12
13
14
```




[name, age, color]

Object.keys()

```
1  /* use Object.keys() to get an array of the keys in the object */
2
3  let pusheen = {
4    name: 'Pusheen',
5    age: 7,
6    color: 'gray and tabby'
7  };
8
9  console.log(Object.keys(pusheen));
10
11
12
13
14
```



nested arrays

```
1  /* objects can store any type of value, including arrays and other
2     objects */
3
4  let pusheen = {
5     name: 'Pusheen',
6     age: 7,
7     colors: ['gray', 'tabby']
8  };
9
10 console.log(pusheen.colors[0]);
11 console.log(pusheen.colors[1]);
12
13
14
```



nested objects

```
1  /* objects can store any type of value, including arrays and other
2     objects */
3
4  let pusheen = {
5     name: 'Pusheen',
6     age: 7,
7     siblings: {
8       sister: 'Stormy',
9       brother: 'Pip'
10    }
11  };
12
13  console.log(pusheen.siblings.sister);
14  console.log(pusheen.siblings.brother);
```



Recap

```
1  /*
2    - what is an object?
3    - why are objects useful?
4    - typeof object
5      - accessing, adding, changing, deleting values
6    - in operator
7    - for...in loop
8    - Object.keys()
9    - nested arrays and objects
10 */
11
12
13
14
```