Objects

Overview

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

```
value1
value1
```

What is an object?

```
/* an object is a collection of key-value pairs */

/* like arrays, objects store values, but instead of storing them in
   numeric "indices", objects store values in string "keys" */

let myArray = ['value1', 'value2'];

let myObject = {
  'aardvark': ‘an aardvark lives in such',
  'key2': 'value2'
};

console.log(myArray[0]);
console.log(myObject[‘aardvark']); //‘an aardvark lives in such',
```

Why are objects useful?

```
/* consider representing Pusheen the Cat as an array */
let pusheen = ['Pusheen', 7, 'gray and tabby'];

/* an array is a good place to hold an ordered list of values, but it
   doesn't store any information about what those values represent */

/* an object's string keys allows objects to store more information about
   the values within it */

let pusheen = {
  'name': 'Pusheen',
  'age': 7,
  'color': 'gray and tabby'
};
```

typeof object

```
console.log(typeof {}); ==> ‘object’
```

Creating an object

```
/* create a new object using curly braces */

/* an object's keys are always strings; you can omit the quotation
   marks */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};
```

Accessing a value

```
Pusheen
7
gray and tabby
undefined
```

```
/* use bracket notation to access a value */

/* pass a string into the brackets that corresponds with a key in the
object */
let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

console.log(pusheen['name']);
console.log(pusheen['age']);
console.log(pusheen['color']);
console.log(pusheen['notAKeyInTheObject']);
```

```
Pusheen
gray and tabby
```

Accessing a value

```
/* any variable or expression that evaluates to a string can be passed
   into the brackets */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

let keyToCheck = 'name';

console.log(pusheen[keyToCheck]);
console.log(pusheen['col' + 'or']);

```

Accessing a value

```
/* you can also use dot notation to access values */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

let keyToCheck = 'name';

console.log(pusheen.name); // no quotes needed with dot notation
console.log(pusheen.age);
console.log(pusheen.color);
console.log(pusheen.keyToCheck);
```

```
Pusheen
7
gray and tabby
undefined
```

Adding a key/value pair

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

```
/* use bracket notation or dot notation to add a
   key/value pair */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

pusheen['sister'] = 'Stormy';
pusheen.brother = 'Pip';

console.log(pusheen);

```

Changing a value

```
/* use bracket notation or dot notation to change a value */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

pusheen['age'] = 8;
pusheen.age = pusheen.age + 1

console.log(pusheen.age);


```

Deleting a key/value pair

```
/* use the delete keyword to delete a key/value pair */

let pusheen = {
  ‘Cat is cool’: 'Pusheen',
  ageOfTheCat: 7,
  color: 'gray and tabby'
};

delete pusheen[‘Cat is Cool’];
delete pusheen.cat is ;

console.log(pusheen);
```

in operator

```
/* use the in operator to check if a key is in the object */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

console.log('name' in pusheen);
console.log('sadness' in pusheen);
```

```
true
false
```

```
name
age
color
```

for...in loop

```
/* use the for…in loop to loop through all of the keys in an object */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

for (let key in pusheen) {. //name
 console.log(key, Pusheen[key])
}
```

for...in loop

```javascript
/* use the for…in loop to loop through all of the keys in an object */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

for (let key in pusheen) {
  console.log("Pusheen's", key, 'is', pusheen[key]);
}
```

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

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

for...in loop

```
/* use the for…in loop to loop through all of the keys in an object */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

for (let key in pusheen) {
  console.log("Pusheen's", key, 'is', pusheen.key);
}
```

```
[ name, age, color ]
```

Object.keys()

```
/* use Object.keys() to get an array of the keys in the object */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  color: 'gray and tabby'
};

Object.keys(pusheen)
Object.values(pusheen) ===> [‘Pusheen’, 7, 'gray and tabby']

console.log(Object.keys(pusheen));


```

```
gray
tabby
```

nested arrays

```
/* objects can store any type of value, including arrays and other
   objects */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  colors: [‘gray’, 'tabby']
};

Pusheen.colors.shift();


console.log(pusheen.colors[0]);
console.log(pusheen.colors[1]);
```

```
Stormy
Pip
```

nested objects

```
/* objects can store any type of value, including arrays and other
   objects */

let pusheen = {
  name: 'Pusheen',
  age: 7,
  siblings: {
    sister: 'Stormy',
    brother: 'Pip'
  }
};

console.log(pusheen[‘siblings’].sister);
console.log(pusheen.siblings.brother);
```

```
something was logged
something else was logged
```

```
Let obj = [{num:1}, {num:2}, {num:3}]
For (let I = 0; I < obj.length; I++){
 obj[I].num
}
```

Recap

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