JS Objects

Object Purpose

Objects allow you to map keys to values.

In this object, we have keys of "firstName", "lastName", and "isInstructor" and values of "Tim", "Garcia", and true respectively.

```
var firstObj = {
  firstName: "Tim",
  lastName: "Garcia",
  isInstructor: true
};
```

Object Format

```
var firstObj = {
  firstName: "Tim"
  lastName: "Garcia",
  isInstructor: true
};
```

Error

Format of an object.

- a key,
- colon,
- value,
- comma.

The last key and value in the object omits the comma.

Leaving out the comma is an error

Trailing Comma

```
var firstObj = {
  firstName: "Tim",
  lastName: "Garcia",
  isInstructor: true,
};
```

Some Javascript Style Guides recommend always including the trailing comma

https://github.com/airbnb/javascript#commas

Accessing Object Values

```
var firstObj = {
  firstName: "Tim",
  lastName: "Garcia",
  isInstructor: true
};
```

dot notation:

```
firstObj.firstName;
firstObj.lastName;
firstObj.isInstructor;
firstObj.keyDoesntExist;
```

bracket notation:

```
firstObj["firstName"];
firstObj["lastName"];
firstObj["isInstructor"];
firstObj["keyDoesntExist"];
```

returns undefined

Using dot and bracket notation

```
var obj = {
  firstName: "Elie",
  lastName: "Schoppik",
  favoriteColor: "purple",
  job: "instructor",
  isDeveloper: true,
};

obj.firstName; // Elie
obj["lastName"]; // Schoppik
obj[favoriteColor]; // This gives us an error
```

Keys Are Always Strings

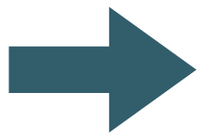
```
var idToName = {
  754: "Tim",
  843: "Matt",
  921: "Janey",
  192: "Elie"
};

idToName.754;  // causes an error
idToName["754"];  // returns “Tim"
```

Adding to objects

Defines object with 2 fields

```
var obj = {
  name: "Jon Snow",
  watchMember: true,
};
```

Append third field



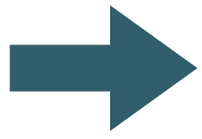
```
obj.gameOfThrones = "awesome";
```

Removing from Objects

Defines object with 2 fields

```
var obj = {
  name: "Elie",
  job: "Instructor"
};
```

Remove one of the fields



```
delete obj.job; // returns true
```

Accessing Object Fields

```
var obj = {
   firstName: "Elie",
   lastName: "Schoppik",
   favoriteColor: "purple",
   job: "instructor",
   isDeveloper: true,
};
```

```
console.log(obj.firstName);
console.log(obj.lastName);
console.log(obj.favoriteColor);
console.log(obj.job);
console.log(obj.developer);
```

Object Iteration

To access the key's value, we must use the bracket notation.

```
var instructor = {
  name: "Matt",
  mathWizard: true,
  dogOwner: true
};

for (var singleKey in instructor) {
  console.log(instructor[singleKey]);
}

// the loop will log:
// "Matt"
// true
// true
```

if...in: Determining If a Key Exists in an Object

true
field exists

```
var obj = {
  favoriteNumber: 33,
  favoriteColor: 'blue'
}

if ("favoriteNumber" in obj) {
  console.log("The favoriteNumber key exists!");
}
// "The favoriteNumber key exists!"
```

false
field does
not exist

```
if ("nothing" in obj) {
  console.log("The nothing key exists!");
}
```