JavaScript Objects, Prototypes and Classes

CREATING AND USING JAVASCRIPT OBJECTS

Jim Cooper
DIRECTOR OF ENGINEERING PRACTICES

@jimthecoop

Creating and Using JavaScript Objects

- Object Literals
- Constructor Functions
- Classes
- Object Equality
- Merging Properties
- Immutability

JavaScript Equality Operators

== Should be avoided. Useful only in rare cases.

=== Most common. Should be used in almost all cases.

Object.is() Less common. Like === except for a few mathmatical differences

Object.is(person1, person2)

Equality Operator: ==

Not type-safe

`“42” == 42`

`0 == false`

`null == undefined`

`“ ” == 0`

`[1,2] == “1,2”`

Equality: === vs. Object.is()

===	Object.is()
✓ Type-safe	Type-safe ✓
Convenient / Concise	Verbose
NaN not equal to NaN	NaN equals NaN
1 / “a” = NaN +0 equals -0	+0 does not equal -0

Summary

```
let person = {
  firstName: ‘Jim’,
  lastName: ‘Cooper’
};
```

Summary

```
let person = {
  isAdult: function() {
    ...
  }
};
```

Summary

```
let firstName = ‘Jim’;
let lastName = ‘Cooper’;

let person = {
   firstname,
   lastName
};
```

Summary

```
let propertyNames = Object.keys(Person);


for (let propertyName in person) {
  ...
}
```

Summary

```
function Person(firstName, lastName) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.age = 0;
  this.isAdult = function () { ... }
}

let jim = new Person(‘Jim’, ‘Cooper’);
```

Summary

== Should be avoided

=== and Object.is() are identical except when:

- Comparing NaN
- Comparing -0