JAVASCRIPT

OBJECTS

WHAT ARE OBJECTS?

- Collections of key value pairs
- Similar to arrays, the main difference being that instead of accessing our values by index, we access them by a key.
- Nearly universal across programming languages, although they may have different names in different languages

HOW DO WE DEFINE OBJECTS?

```
var shirt = {
  size: ‘medium’,
  color: ‘blue’,
  price: 60
}
```

- Between curly braces { }
- Keys and values
- Separated with commas

HOW DO WE DEFINE OBJECTS?

```
var shirt = {
  ‘size’: ‘medium’,
  ‘color’: ‘blue',
  ‘price’: 60
}
```

- Keys are actually strings, but we do not need the quotes when defining objects
- Values can be any datatype

HOW DO WE ACCESS VALUES?

```
var shirt = {
  size: ‘medium’,
  color: ‘blue’,
  price: 60
}

console.log(shirt.size);
  => ‘medium’

console.log(shirt[‘size’]);
  => ‘medium’
```

- Dot notation
  objectName.keyName
- Bracket notation
  objectName[‘keyName’]

DOT NOTATION VS. BRACKET NOTATION

```
var shirt = {
  size: 'medium',
  color: 'blue',
  price: 60
}

var property = 'size';
console.log(shirt[property]);
  => 'medium'

property = 'price';
console.log(shirt[property]);
  => 60
```

- Dot notation
  - You have to know what the key is
- Bracket notation
  - you can use a variable as the key

HOW DO WE SET VALUES?

```
var shirt = {
  size: ‘medium’,
  color: ‘blue’,
  price: 60
}

shirt.size = ‘large’;

shirt[‘size’] = ‘large’;
```

- Dot notation assignment
  objectName.key = value
- Bracket notation assignment
  objectName[‘key’] = value

ARRAYS IN OBJECTS

```
var me = {
  name: 'Jackie Casper',
  hometown: 'New York',
  interests: [
    'traveling',
    'SCUBA diving',
    'Game of Thrones'
  ]
}

console.log(me.interests[2]);
  => 'Game of Thrones'
```

- Indices can be used on arrays in objects
- Access the array from the object
- Use an index to access the desired value

OBJECTS IN OBJECTS

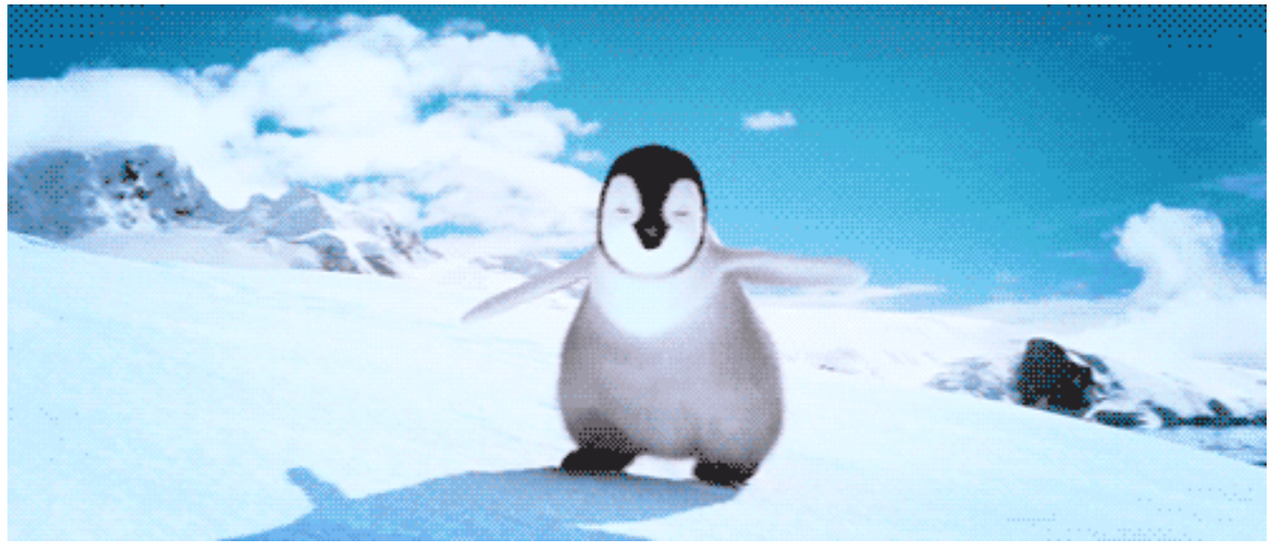
```
var lesson = {
  name: 'JS Objects',
  cohort: 'Penguins',
  instructor: {
    name: 'Jackie Casper',
    slackName: 'jackie'
  }
}

console.log(lesson.instructor.name);
  => 'Jackie Casper'

console.log(lesson['instructor'].name);
  => 'Jackie Casper'
```

- You can access objects within objects by chaining either dot notation, bracket notation, or a mix of both!

REVIEW



- What is an object?
- What is a Key? Value?
- What is the difference between dot and bracket notation?
- How can we access an array inside of an object? An object inside of an object?

CODEALONG

Assignment 1

ARRAYS OF OBJECTS

```
var shirts = [
  {
    size: ‘medium’,
    color: ‘blue’,
    price: 60
  },
  {
    size: ‘large’,
    color: ‘black’,
    price: 75
  }
]

console.log(shirt[0].color)
  => ‘blue’
```

- Access the object in the array with an index
- Access properties on the object with dot or bracket notation

LOOPING ARRAYS OF OBJECTS

```
var shirts = [
  {
    size: 'medium',
    color: 'blue',
    price: 60
  },
  {
    size: 'large',
    color: 'black',
    price: 75
  }
];
```

```
for(var i=0; i<shirts.length; i++) {
  console.log(shirts[i].color);
}
```

- Loop through arrays in the same way we did yesterday
- Can then access properties on each object in the array

LOOPING ARRAYS OF OBJECTS

```
var shirts = [
  {
    size: 'medium',
    color: 'blue',
    price: 60
  },
  {
    size: 'large',
    color: 'black',
    price: 75
  },
  {
    size: 'medium',
    color: 'black',
    price: 55
  }
];
```

```
for(var i=0; i<shirts.length; i++) {
  shirts.price = shirts.price * .75;
}
```

- What would the code above do?

LOOPING ARRAYS OF OBJECTS

```
var shirts = [
  {
    size: 'medium',
    color: 'blue',
    price: 60
  },
  {
    size: 'large',
    color: 'black',
    price: 75
  },
  {
    size: 'medium',
    color: 'black',
    price: 55
  }
];
```

```
var blackShirts = [];

for(var i=0; i<shirts.length; i++) {
  if(shirts[i].color === 'black') {
    blackShirts.push(shirts[i]);
  }
}
```

- What would be in blackShirts?

LOOPING ARRAYS OF OBJECTS

```
var shirts = [
  {
    size: 'medium',
    color: 'blue',
    price: 60
  },
  {
    size: 'large',
    color: 'black',
    price: 75
  },
  {
    size: 'medium',
    color: 'black',
    price: 55
  }
];
```

```
var shirtCounts = {};

for(var i=0; i<shirts.length; i++) {
  shirtColor = shirts[i].color;
  if(shirtCounts[shirtColor] === undefined) {
    shirtCounts[shirtColor] = 1;
  } else {
    shirtCounts[shirtColor] += 1;
  }
}
```

- What would shirtCounts look like?

YOUR TURN

Assignment 2

FACTORY FUNCTIONS

- A function that returns an object.
- Can be used to programmatically create objects!

```
function userFactory(name, email, phoneNumber){
  return {
    name: name,
    email: email.toLowerCase(),
    phoneNumber: phoneNumber
  }
}

var jackie = userFactory(‘Jackie’, ‘jaclyn.casper@generalassemb.ly', ‘555-938-9312’)
```

FACTORY FUNCTIONS

```
function markerFactory(color){
  return {
    item: ‘Marker’,
    color: color
  }
}

colors = [‘blue’, ‘black’, ‘pink’, ‘red’];
markers = [];

for(var i=0; i<colors.length; i++){
  var newMarker = markerFactory(colors[i]);
  markers.push(newMarker);
}
```

- What will be in the markers array?

YOUR TURN

Assignment 3

REVIEW

- What is an object?
- What is a Key? Value?
- What is the difference between dot and bracket notation?
- How can we access an array inside of an object? An object inside of an object?
- What is a factory function?