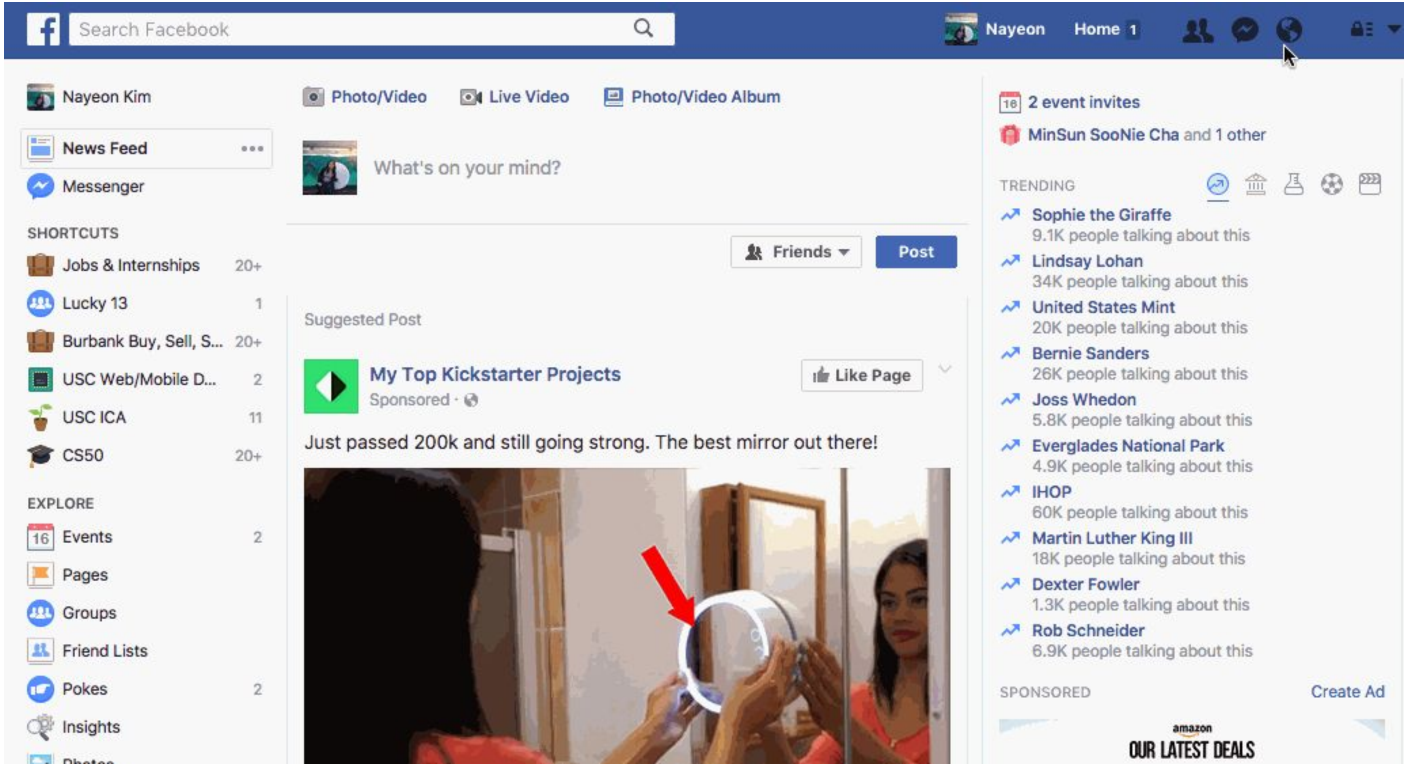
Lecture 09: JS Objects, DOM Manipulation

ITP 303 Full-Stack Web Development

JavaScript in live action



What's on your mind?

Suggested Post

My Top Kickstarter Projects
Sponsored ·

Just passed 200k and still going strong. The best mirror out there!

2 event invites

MinSun SooNie Cha and 1 other

TRENDING

Sophie the Giraffe
9.1K people talking about this

Lindsay Lohan
34K people talking about this

United States Mint
20K people talking about this

Bernie Sanders
26K people talking about this

Joss Whedon
5.8K people talking about this

Everglades National Park
4.9K people talking about this

IHOP
60K people talking about this

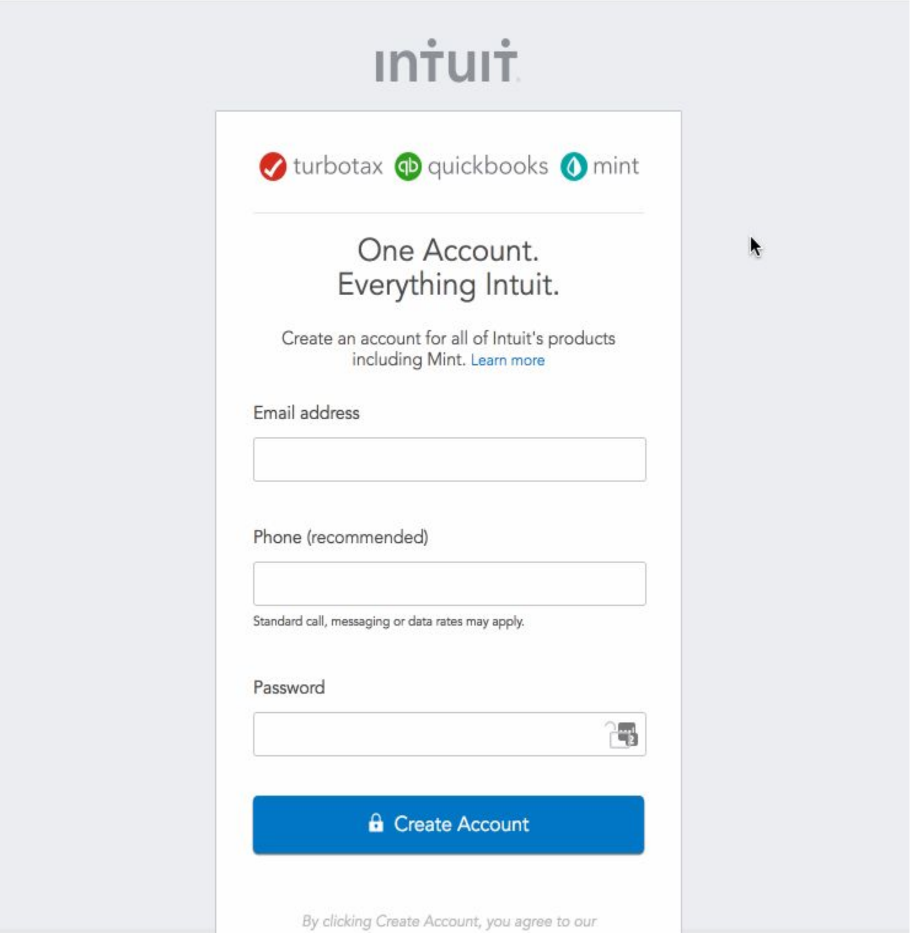
Martin Luther King III
18K people talking about this

Dexter Fowler
1.3K people talking about this

Rob Schneider
6.9K people talking about this

SPONSORED

JavaScript in live action



One Account.
Everything Intuit.

Create an account for all of Intuit's products including Mint. Learn more

Email address

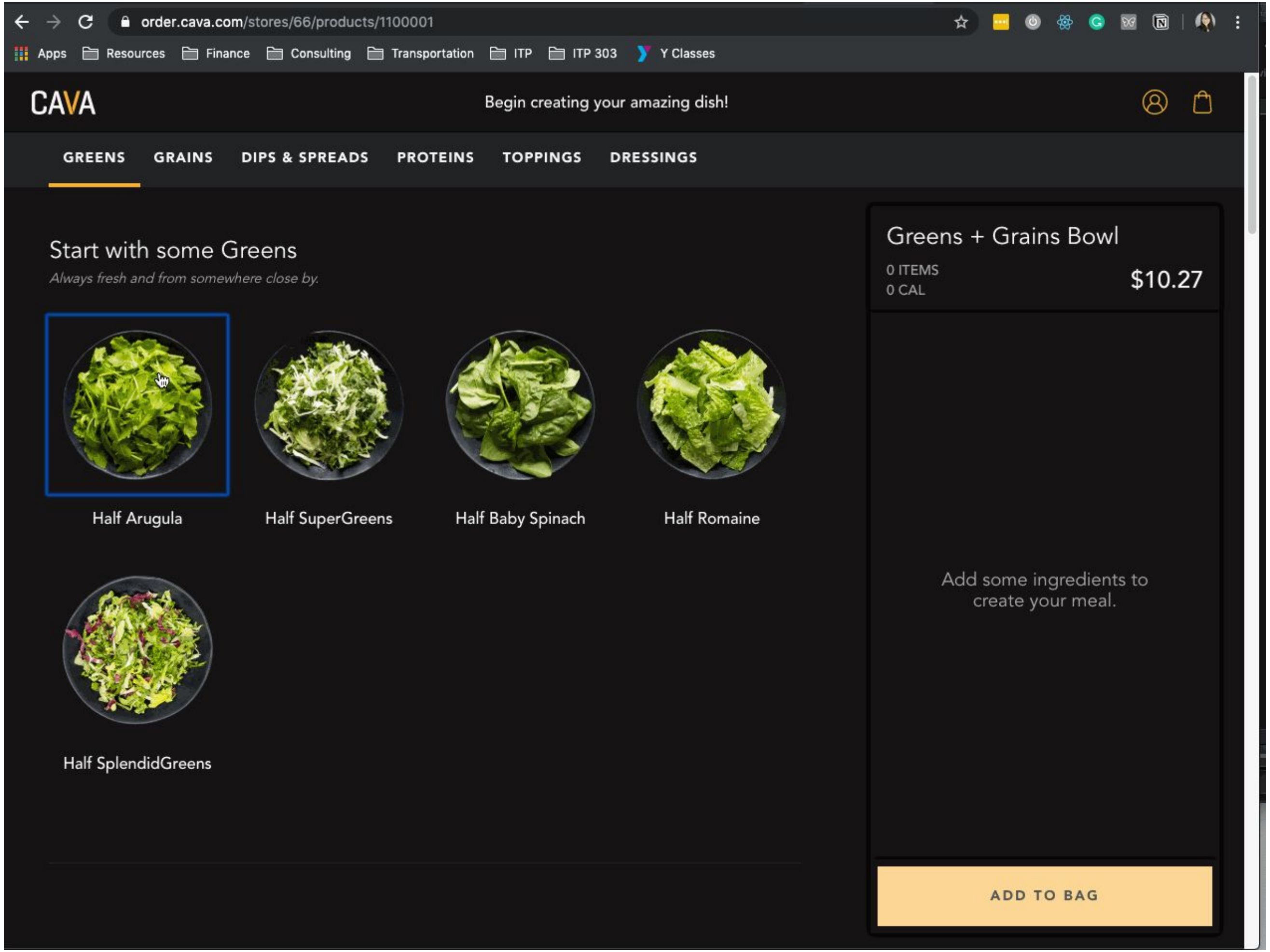
Phone (recommended)

Standard call, messaging or data rates may apply.

Password

By clicking Create Account, you agree to our

JavaScript can also do this!



Begin creating your amazing dish!

Start with some Greens

Always fresh and from somewhere close by.

Half Arugula

Half SuperGreens

Half Baby Spinach

Half Romaine

Half SplendidGreens

Greens + Grains Bowl

0 ITEMS

0 CAL

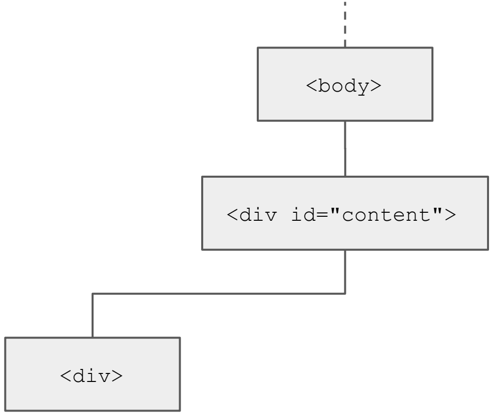
$10.27

Add some ingredients to create your meal.

Creating DOM nodes

Creating DOM Nodes

`document.createElement( `*`tag`*`)`	Create a new HTML *tag*.
`Parent``.appendChild( `*`newNode`*`)`	Add *newNode* do the DOM as the last child of *Parent*.



<body>

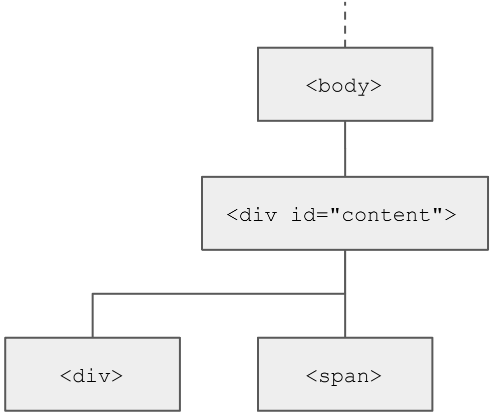
<div id="content">

<div>

```
// Create a new <div> tag.
var div = document.createElement('div');

// Add 'div' as last child of #content.
document.querySelector('#content').appendChild(div);
```

Creating DOM Nodes

`document.createElement( tag)`	Create a new HTML *tag*.
`Parent.appendChild( newNode)`	Add *newNode* do the DOM as the last child of *Parent*.



<body>

<div id="content">

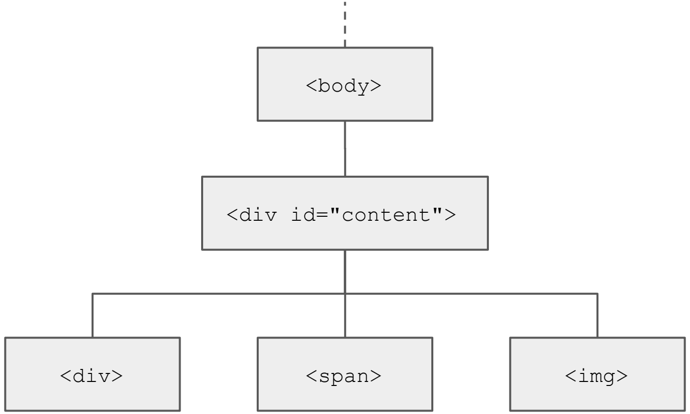
<div>

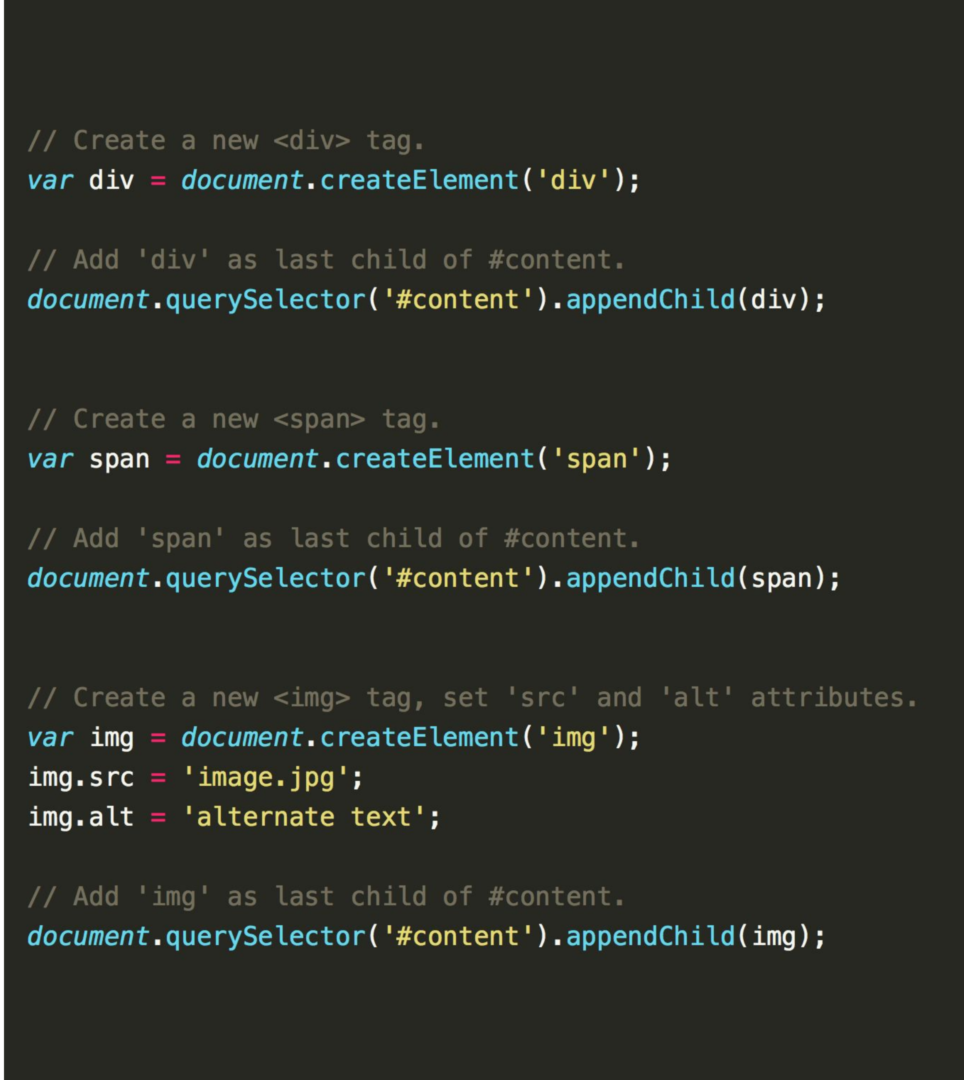
<span>

```
// Create a new <div> tag.
var div = document.createElement('div');

// Add 'div' as last child of #content.
document.querySelector('#content').appendChild(div);


// Create a new <span> tag.
var span = document.createElement('span');

// Add 'span' as last child of #content.
document.querySelector('#content').appendChild(span);
```

Creating DOM Nodes

`document.createElement( tag)`	Create a new HTML *tag*.
`Parent.appendChild( newNode)`	Add *newNode* do the DOM as the last child of *Parent*.



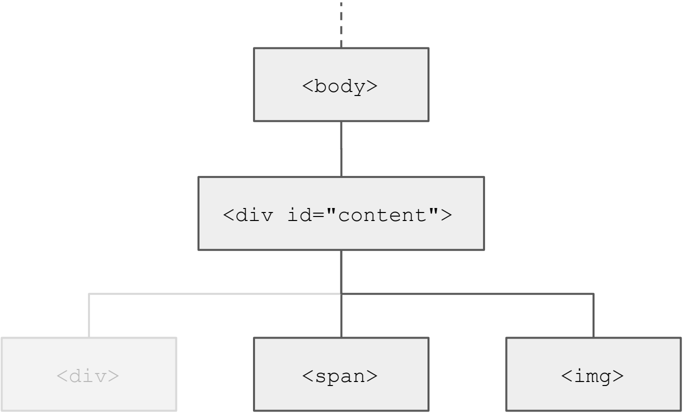


```
// Create a new <div> tag.
var div = document.createElement('div');

// Add 'div' as last child of #content.
document.querySelector('#content').appendChild(div);


// Create a new <span> tag.
var span = document.createElement('span');

// Add 'span' as last child of #content.
document.querySelector('#content').appendChild(span);


// Create a new <img> tag, set 'src' and 'alt' attributes.
var img = document.createElement('img');
img.src = 'image.jpg';
img.alt = 'alternate text';

// Add 'img' as last child of #content.
document.querySelector('#content').appendChild(img);
```

Removing DOM Nodes

Parent.removeChild(*Node*)

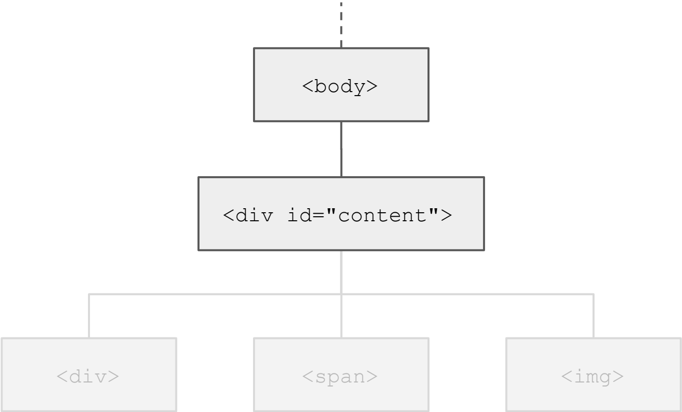
Create a new HTML *tag*.



```
var parentNode = document.querySelector('#content');


// Remove first child from 'parentNode' element.
parentNode.removeChild( parentNode.children[0] );
```

Removing DOM Nodes

`Parent.removeChild( Node)` — Create a new HTML *tag*.



```
var parentNode = document.querySelector('#content');


// Remove first child from 'parentNode' element.
parentNode.removeChild( parentNode.children[0] );


// Remove all children from 'parentNode' element.
parentNode.innerHTML = '';
```

JavaScript Objects

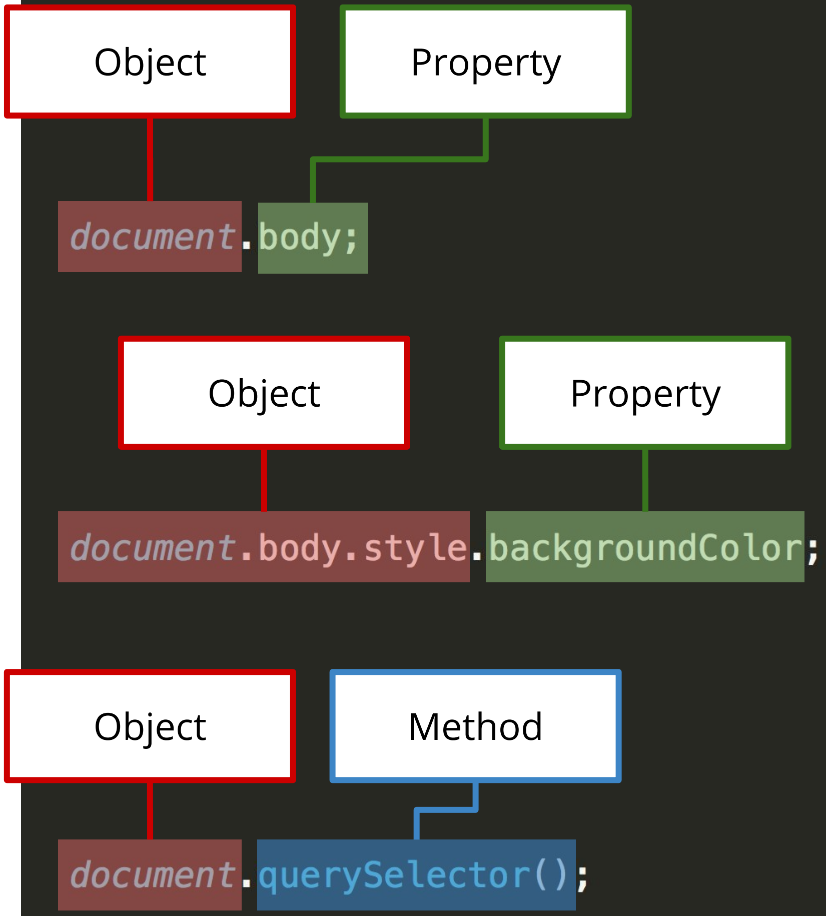
JavaScript Objects

JavaScript is an Object-Oriented Programming (OOP) Language.

Object – collection of closely related variables (**properties**) and functions (**methods**).

Why OOP?

- Modeled after the real world,
- Each object keeps track of its own behavior and state,
- They help break down complex problems into independent entities.



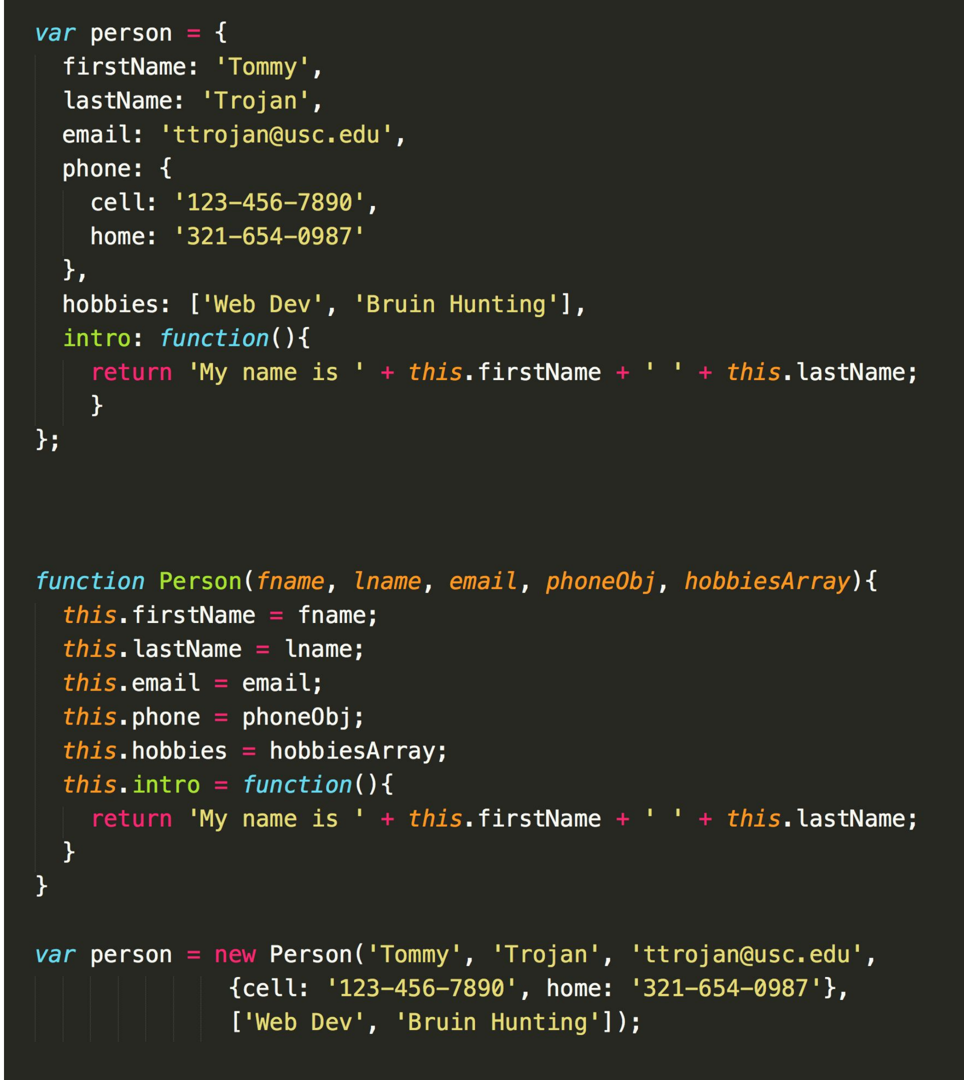
```
document.body;
document.body.style.backgroundColor;
document.querySelector();
```

Creating Objects

Each object has properties and methods stored in `key: value` pairs.

Two ways to create objects:

1. Literal Syntax
2. Constructor Function Syntax

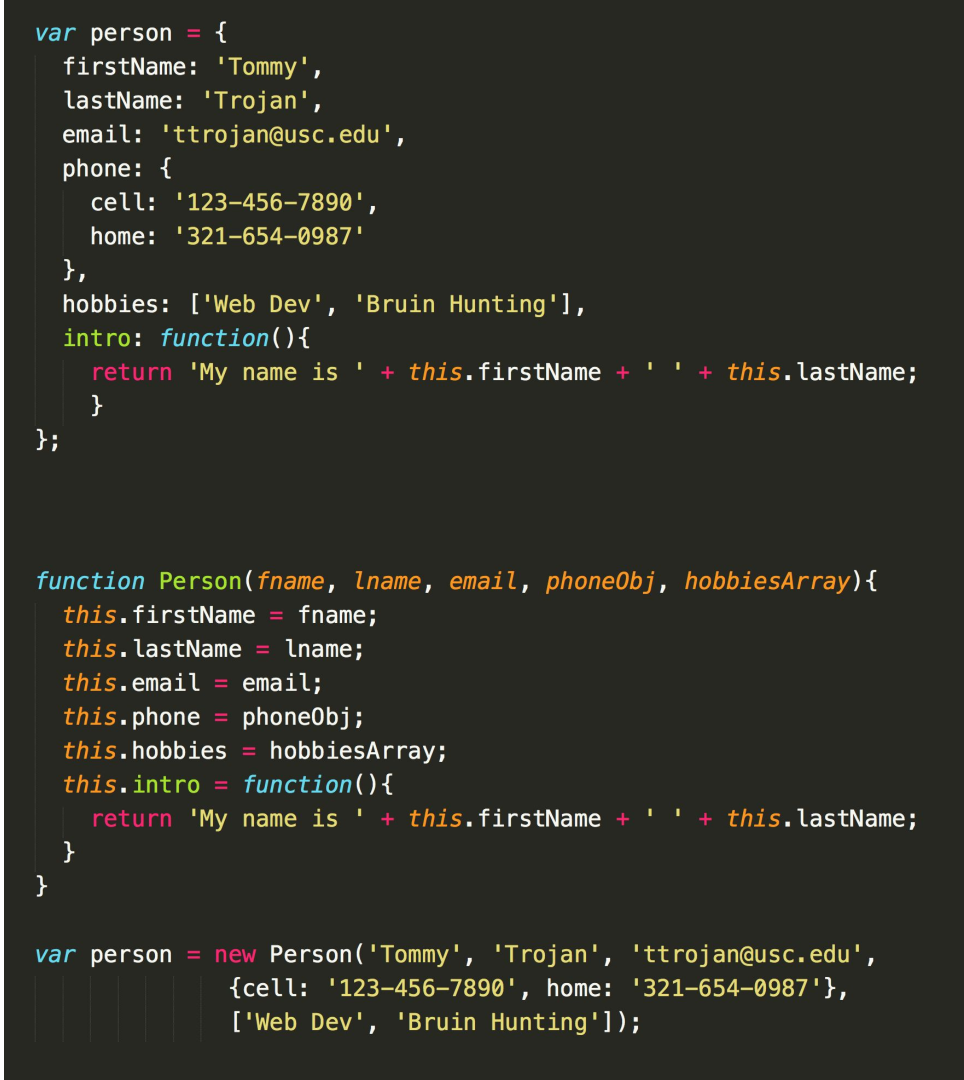


```
var person = {
  firstName: 'Tommy',
  lastName: 'Trojan',
  email: 'ttrojan@usc.edu',
  phone: {
    cell: '123-456-7890',
    home: '321-654-0987'
  },
  hobbies: ['Web Dev', 'Bruin Hunting'],
  intro: function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
    }
};
```

```
function Person(fname, lname, email, phoneObj, hobbiesArray){
  this.firstName = fname;
  this.lastName = lname;
  this.email = email;
  this.phone = phoneObj;
  this.hobbies = hobbiesArray;
  this.intro = function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
  }
}

var person = new Person('Tommy', 'Trojan', 'ttrojan@usc.edu',
                         {cell: '123-456-7890', home: '321-654-0987'},
                         ['Web Dev', 'Bruin Hunting']);
```

Creating Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table><tr><th>Key</th><th>Value</th></tr><tr><td>cell</td><td>123-456-7890</td></tr><tr><td>home</td><td>321-654-0987</td></tr></table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>



```
var person = {
  firstName: 'Tommy',
  lastName: 'Trojan',
  email: 'ttrojan@usc.edu',
  phone: {
    cell: '123-456-7890',
    home: '321-654-0987'
  },
  hobbies: ['Web Dev', 'Bruin Hunting'],
  intro: function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
    }
};



function Person(fname, lname, email, phoneObj, hobbiesArray){
  this.firstName = fname;
  this.lastName = lname;
  this.email = email;
  this.phone = phoneObj;
  this.hobbies = hobbiesArray;
  this.intro = function(){
    return 'My name is ' + this.firstName + ' ' + this.lastName;
  }
}

var person = new Person('Tommy', 'Trojan', 'ttrojan@usc.edu',
            {cell: '123-456-7890', home: '321-654-0987'},
            ['Web Dev', 'Bruin Hunting']);
```

Accessing Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>cell</td><td>123-456-7890</td></tr>
<tr><td>home</td><td>321-654-0987</td></tr>
</table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>

```
person.firstName
```

Accessing Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table><tr><th>Key</th><th>Value</th></tr><tr><td>cell</td><td>123-456-7890</td></tr><tr><td>home</td><td>321-654-0987</td></tr></table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>

```
person.firstName

person.phone
```

Accessing Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table><tr><th>Key</th><th>Value</th></tr><tr><td>cell</td><td>123-456-7890</td></tr><tr><td>home</td><td>321-654-0987</td></tr></table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>

```
person.firstName

person.phone

person.phone.cell
```

Accessing Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>cell</td><td>123-456-7890</td></tr>
<tr><td>home</td><td>321-654-0987</td></tr>
</table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>

```
person.firstName

person.phone

person.phone.cell

person.hobbies
```

Accessing Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table><tr><th>Key</th><th>Value</th></tr><tr><td>cell</td><td>123-456-7890</td></tr><tr><td>home</td><td>321-654-0987</td></tr></table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>

```
person.firstName

person.phone

person.phone.cell

person.hobbies

person.hobbies[0]
```

Accessing Objects

<table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>firstName</td><td>Tommy</td></tr>
<tr><td>lastName</td><td>Trojan</td></tr>
<tr><td>email</td><td>ttrojan@usc.edu</td></tr>
<tr><td>phone</td><td><table>
<tr><th>Key</th><th>Value</th></tr>
<tr><td>cell</td><td>123-456-7890</td></tr>
<tr><td>home</td><td>321-654-0987</td></tr>
</table></td></tr>
<tr><td>hobbies</td><td><table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table></td></tr>
<tr><td>intro</td><td>function(){...}</td></tr>
</table>

```
person.firstName

person.phone

person.phone.cell

person.hobbies

person.hobbies[0]

person.intro()
```