Lecture 08: JS Numbers, Strings, Objects, User Input

ITP 303 Full-Stack Web Development

More DOM Events

Keyboard Events

`onkeydown`	Key pressed.
`onkeyup`	Key released.
`onkeypress`	Fired immediately after `onkeydown`. Only works with keys used for “typing” text. Not fired by Ctrl, Alt, arrows, etc.

Form Events

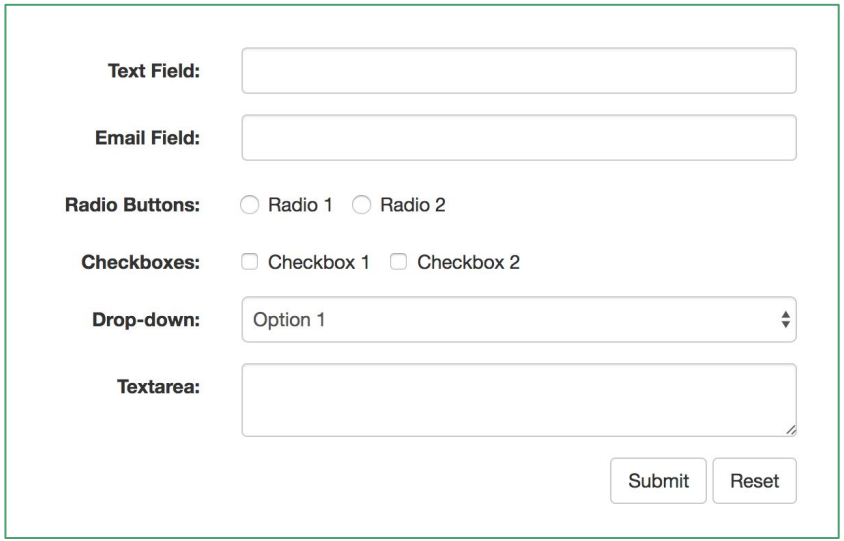
`oninput`	Form element value changed.
`onchange`	Form element state changed. Text fields fired after losing focus.
`onsubmit`	Form is submitted.

Forms & User Input

HTML Forms

Allow for user interaction and data collection.

Result (with additional styling):




```html
<form action="" method="">
  <label for="text-id">Text Field:</label>
  <input type="text" id="text-id" name="text-name">

  <label for="email-id">Email Field:</label>
  <input type="email" id="email-id" name="text-name">

  <label for="">Radio Buttons:</label>
  <label>
    <input type="radio" id="radio-1" name="radio-name" value="val-1">
    Radio 1
  </label>
  <label>
    <input type="radio" id="radio-2" name="radio-name" value="val-2">
    Radio 2
  </label>

  <label for="">Checkboxes:</label>
  <label>
    <input type="checkbox" id="checkbox-1" name="checkbox-name" value="val-1">
    Checkbox 1
  </label>
  <label>
    <input type="checkbox" id="checkbox-2" name="checkbox-name" value="val-2">
    Checkbox 2
  </label>

  <label for="select-id">Drop-down:</label>
  <select id="select-id" name="select-name">
    <option value="val1">Option 1</option>
    <option value="val2">Option 2</option>
  </select>

  <label for="textarea-id">Textarea:</label>
  <textarea id="textarea-id" name="textarea-name"></textarea>

  <button type="submit">Submit</button>
  <button type="reset">Reset</button>
</form>
```

Form Tags

Define HTML form.

`action`	Specifies where form data is sent upon submission.
`method`	Specifies HTTP method used to send data: `POST` or `GET`.

```
<form action="" method="">

    ...

</form>
```

Text Fields

Single-line text fields.

Result (with additional styling):


```
Text Field:
<input type="text" id="text-id" name="text-name">
```

Field name submitted along with data.

Email Fields

Text fields with basic email validation.

Result (with additional styling):


```
Email Field:
<input type="email" id="email-id" name="text-name">
```

Radio Buttons

Group of mutually exclusive fields.

Result (with additional styling):

Group radio buttons by setting same `name` attribute value.

```
<input type="radio" id="radio-1" name="radio-name" value="val-1">
Radio 1

<input type="radio" id="radio-2" name="radio-name" value="val-2">
Radio 2
```

Value submitted when radio button is selected.

Checkboxes

Group of independent fields.

Result (with additional styling):

Group checkboxes by setting same `name` attribute value.

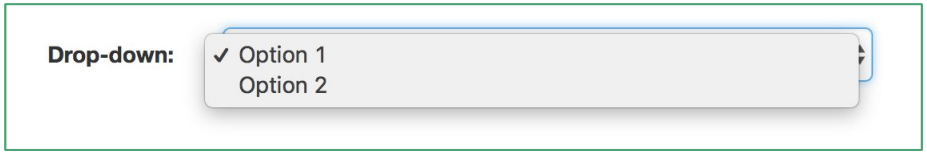
```
<input type="checkbox" id="checkbox-1" name="checkbox-name" value="val-1">
Checkbox 1

<input type="checkbox" id="checkbox-2" name="checkbox-name" value="val-2">
Checkbox 2
```

Value submitted when checkbox is selected.

Drop-Down Lists

Pop-up menu that allows users to choose a single option from the list.

Result (with additional styling):




```
Drop-down:
<select id="select-id" name="select-name">
  <option value="val1">Option 1</option>
  <option value="val2">Option 2</option>
</select>
```

Textareas

Multi-line text fields.

Result (with additional styling):

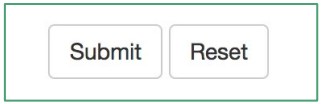
Textarea:

```
<textarea id="textarea-id" name="textarea-name"></textarea>
```

Buttons

Allow users to submit or reset forms.

Result (with additional styling):




```
<button type="submit">Submit</button>
<button type="reset">Reset</button>
```

Form Labels

Captions for form elements.

Clicking a label triggers assigned form field.

Usage 1 Set `for` attribute to `id` of corresponding form field.

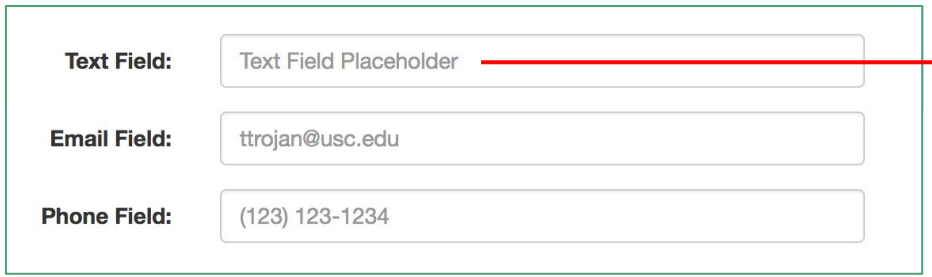
Usage 2 Wrap label around the form field.

Placeholder Attribute

Text displayed within form field before user input.

Usually used to specify hints or expected formats.

Result (with additional styling):




```
<label for="text-id">Text Field:</label>
<input type="text" id="text-id" name="text-name" placeholder="Text Field Placeholder">

<label for="email-id">Email Field:</label>
<input type="email" id="email-id" name="text-name" placeholder="ttrojan@usc.edu">

<label for="phone-id">Phone Field:</label>
<input type="text" id="phone-id" name="text-name" placeholder="(123) 123-1234">
```

Strings

Strings

Strings are series of characters.

- Sometimes treated like arrays with letters in each slot.

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		W	o	r	l	d	!

`.replace(find, replace)`	Find first occurrence of *find* and replace it with *replace*.
`.substr(start, [length])`	Return substring starting at index *start*. Length of substring is optional.

```
var myString = "Hello World!";


console.log( myString.replace('Hello', 'Hi') );
// Returns "Hi World!"


console.log( myString.substr(6) );
// Returns "World!"

console.log( myString.substr(0, 5) );
// Returns "Hello"
```

String Operations

`.length`	Length of the string.
`.indexOf(` *sub*`)`	Return index of first occurrence of *sub* within the string. Returns `-1` if *sub* is not found.
`.toLowerCase(` *str*`)`	Convert *str* to lowercase.
`.toUpperCase(` *str*`)`	Convert *str* to uppercase.

Resource: String Reference.

```
console.log( str.length );

if (str.indexOf("olo") > -1 ) {
  console.log('Found');
}

if ( str.toLowerCase() == str2.toLowerCase() ) {
  console.log('Same strings (case-insensitive');
}

if( str.toUpperCase().indexOf('LOREM') == -1 ) {
  console.log('Not Found');
}
```

Substring methods

`.substr()`	Returns the characters in a string beginning at the specified location through the specified number of characters.
`.substring()`	Returns the characters in a string between two indexes into the string.
`.slice()`	Extracts a section of a string and returns a new string.
`.split()`	Splits a String object into an array of strings by separating the string into substrings.

Full list here: https://www.w3schools.com/js/js_string_methods.asp

JavaScript Objects

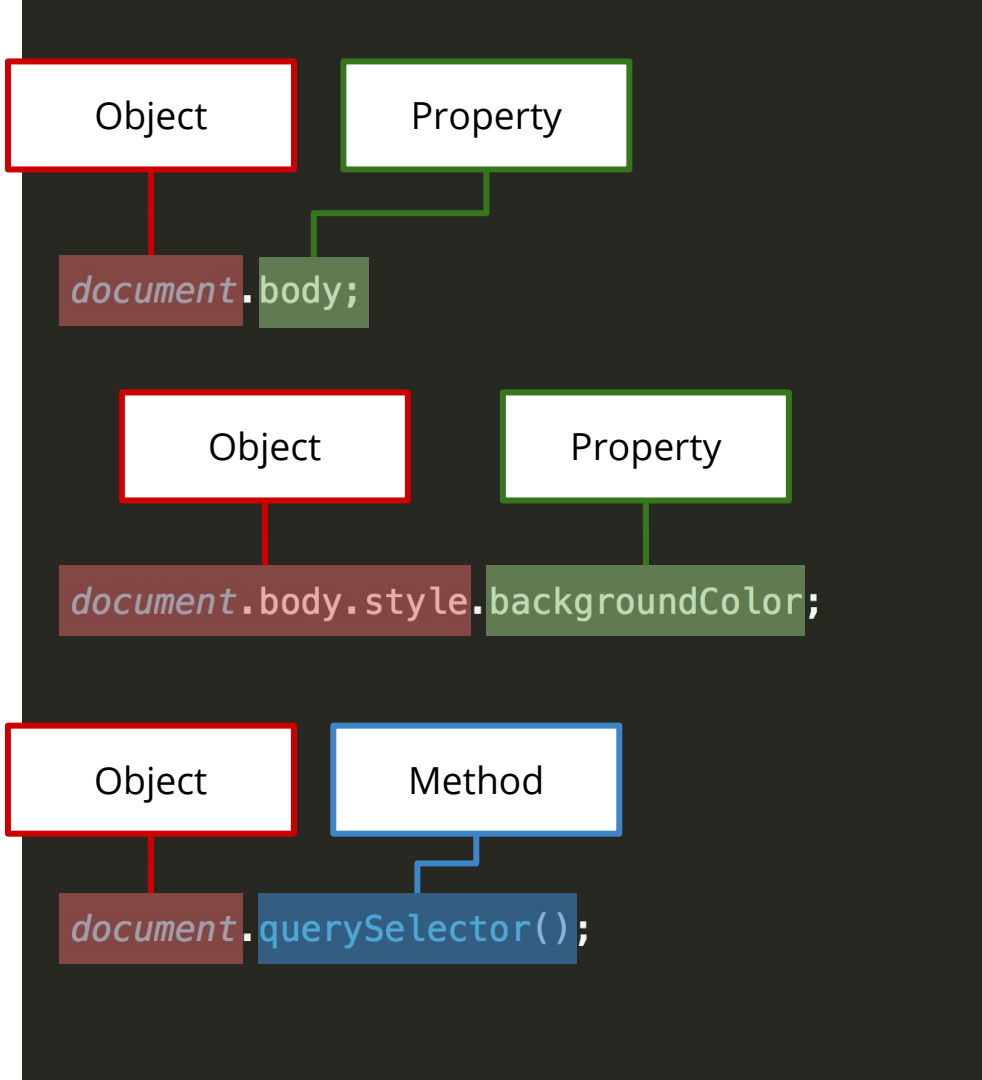
JavaScript Objects

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.



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

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<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>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>
intro	function(){...}

```
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

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<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>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>
intro	function(){...}

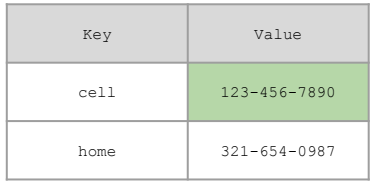
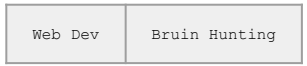
```
person.firstName
```

Accessing Objects

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<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>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>
intro	function(){...}

```
person.firstName

person.phone
```

Accessing Objects

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	
hobbies	
intro	function(){...}


```
person.firstName

person.phone

person.phone.cell
```

Accessing Objects

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<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>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>
intro	function(){...}

```
person.firstName

person.phone

person.phone.cell

person.hobbies
```

Accessing Objects

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<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>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>
intro	function(){...}

```
person.firstName

person.phone

person.phone.cell

person.hobbies

person.hobbies[0]
```

Accessing Objects

Key	Value
firstName	Tommy
lastName	Trojan
email	ttrojan@usc.edu
phone	<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>
hobbies	<table><tr><td>Web Dev</td><td>Bruin Hunting</td></tr></table>
intro	function(){...}

```
person.firstName

person.phone

person.phone.cell

person.hobbies

person.hobbies[0]

person.intro()
```