

CSC435: Web Programming

Lecture 10: JavaScript: Objects

Bei Xiao

American University

Feb 20, 2014

Last Lecture

- Functions: `function myFunct(a,b){}`
- DOM Objects: `getDocumentByID(ID,param);`

Overview

- Form Validation
- Example: Making a div to move as response to user's specification.
- Objects: construction, retrieving, prototypes

Form Validation

- JavaScript can be used to validate data in HTML forms before it is sent to the data.
- Form data that typically are checked by JavaScript could be:
 1. Has the user left required fields empty?
 2. Has the user entered a valid email address?
 3. Is the password achieve certain length and have certain characters?
 4. Has the user entered text in a numeric field?

Form Validation

- Why does JavaScript validate user input before sending it to the user?

Form Validation: Syntax

JavaScript

```
Function validateForm() {  
  // code that checks the form  
  If( field is empty){  
    Return: false;}  
  Else{  
    Return:true;  
  }  
}
```

Form

```
<form name="myForm" action="FirstNameValidation.asp"  
  onsubmit="return validateForm()" method="post">
```

Form Validation: Example 1

- Check whether a field is filled

Form Validation: Example 2

- Check whether an email has the general syntax of an email such as:
 1. Whether it has “.” and “@” characters in the string.
 2. @ must not be the first character of the string.
 3. The last “.” must be present after the @.

Form Exercise 1: put them together

- Put the first name, last name, and email together in a form.
- Validate that all the fields are not empty and the email fulfill the requirement of the previous examples.

Form Validation: Exercise 2

Write a form validation for password, if the user input a password that has length smaller than 5, alert them” password must have at least 5 characters”.

Form Validation: Submit Disabled

Sometimes, we want to disabled a submit button before we click on agreement.

Fun part: Make a div that moves

- Step 1: Create a basic html that contain a div that has the id “tcp”.
- Step 2: Make it has the following properties or whatever you wish:

```
body {
    color:white;
    font-family:verdana;
}
#tcp {
    background-color:#003366;
    position:absolute;
    width:300px;
    height:250px;
    text-align:center;
    padding:20px;
    top:10%;
    left:37%;
    border-radius:4px;
}
```

Fun part: Make a div that moves

- Step 3: Make two input boxes, one to specify the top pixels to move and the another to specify the left pixels to move.

Move Top: `<input type="text" maxlength="3" id="mtop" placeholder="Add pixel to move top" />
`

- Step 4: Create another button called “Move” with the id “ done”.

Fun part: Make a div that moves

- Step 5: Write a function called “ Moveit”, to move the div according to user input. The function takes three values, the div name, the top moving pixels, the left moving pixels.

```
function moveit(dmove, gtop, gleft){  
    dmove.style.top = gtop.value + 'px';  
    dmove.style.left = gleft.value + 'px';  
}
```

Fun part: Make a div that moves

- Step 6: Calling the function by click on the button “Move”. You can use `getElementById` to access to the div name, the two pixel input boxes.

```
OnClick="moveit(document.getElementById('tcp'),  
document.getElementById('mtop'),  
document.getElementById('mleft'))"
```

- Step 7: Write a paragraph such as `<p>To make the box go bottom or right you can simply add greater digits like 360</p>`
- Step 8: Stylize the page using CSS such as the button style, hover, etc.

Objects: construction

- An object is a container of properties, where a property has a name and a value.

- Construction:

```
var myCar = {"make": "Ford";  
"model": "Mustang";  
"year": 1969;  
}
```

```
myCar["make"] = "Ford";  
myCar["model"] = "Mustang";  
myCar["year"] = 1969;
```


Objects: construction by using functions

- One can use function to construct an object :

```
function person(firstname,lastname,age,eyecolor)
{
this.firstname=firstname;
this.lastname=lastname;
this.age=age;
this.eyecolor=eyecolor;
}
myFather=new person("John","Doe",50,"blue");
```

- New instances can be created

```
myMother = new person("Mary", "Smith", "45", "brown");
```

Objects: instance, method, and “this”

```
<!DOCTYPE html>
<html>
<head>
  <title> JavaScript Create an Object </title>
<script type="text/javascript">
  myNewObject = new Object();
  myNewObject.info = 'I am a shinny new object';
  function myFunc(){
    alert(this.info);
  }
  myNewObject.showInfo = myFunc;
</script>
</head>
<body>
  <input type = "button" value="Good showInfo Call" onclick ="myNewObject.showInfo()"/>
  <input type = "button" value="myFunc Call" onclick ="myFunc()"/>
  <input type = "button" value="Bad showInfo Call" onclick ="showInfo()"/>
</body>
</html>
```

Objects: Quiz

Which of the following is a valid way to create a direct instance of an object?

- a. `myObject.create ();`
- b. `myObject = new Object;`
- c. `myObject = new Object();`

Objects: nest

- A property's value can be obtained from any expression, including another object.

```
var flight = {  
  airline: "United",  
  number: 815,  
  departure:{  
    IATA:"SYD",  
    Time:"2004-09-22 14:55".  
    City:"Sydney"  
  }  
};
```

Objects: retrieval

- Retrieval

```
flight.departure.IATA // "SYD"  
myCar.model           // "Ford"  
flight.equipment      // undefined  
var equipment = flight.equipment || "unknown";
```

Objects: Update

- Update: a value in an object can be updated by assignment

```
flight.equipment = {  
  Model: 'Boeing 777'  
};  
Flight.status = 'overdue';  
  
myCar.mileage = "50K";
```

Objects: Reference

- Reference:

```
var Stooge = {  
  "first-name": "Jeremy",  
  "second-name": "Howard"  
}
```

```
var x = stooge;
```

```
x.nickname = 'Curly';
```

```
var nick = stooge.nickname; //nick is 'Curly' because x and //  
  stooge are references to the same object
```

Quick Quiz

- What is the the output of the following code after “alert”?
function person (firstname,lastname,age,eyecolor)
{
this.firstname=firstname;
this.lastname=lastname;
this.age=age;
this.eyecolor=eyecolor;
}
myFather = new person("John","Doe",50,"blue");
var x =myFather;
x.job = “Teacher”;
Var profession = myFather.job;
alert(profession);

Homework & Readings

Sams teach you JavaScript in 24 hours, Hours 6 and 7.

JavaScript: The good parts, Chapter 3. Objects

Code Academy, JavaScript (tracks 5-6).

<http://www.codecademy.com/tracks/javascript>

Objects: Prototype

Objects: Reflection

Objects: Enumeration

Summary of what we have learned