

Javascript Forms

Web Authoring and Design

Benjamin Kenwright

Outline

- Review
- What do we mean Javascript Forms?
- Object Orientated Concepts
- Summary
- Review/Discussion

Defining functions

Function Declarations:

```
function functionName(parameters (optional))  
{  
    FunctionBody  
}
```

Arguments

Function without arguments

```
function greetings()  
{  
    alert("Good morning");  
}
```

Function with arguments

```
function findSum(a,b)  
{  
    alert("Sum is : " + (a + b));  
}
```

Calling a Function

You can call Javascript functions by simply call the function name

```
function greetings()  
{  
    alert("Good Morning");  
}  
greetings(); //just give the function name on.
```

Return Values

```
<script>  
    function findSum(x,y)  
    {  
        var result = x+y;  
        return result;  
    }  
    var sum = findSum(25,30);  
    alert ("Sum is : "+ sum);  
</script>
```

Question

☐ Is Javascript an Object Orientated Language?

a) Yes

b) No

Answer

■ a) Yes

JavaScript is an object oriented programming language

How to create and use objects in Javascript

```
var student = {  
    name: "Jack",  
    age: 12,  
    getName: function()  
    {  
        alert(this.name);  
    }  
};
```

Call Object function

```
var student = {  
    name: "Jack",  
    age: 12,  
    getName: function()  
    {  
        alert(this.name);  
    }  
};
```

```
student.getName();
```

Object constructor

```
function studentConstructor() {  
    this.name = "Jack";  
    this.age = 12;  
    this.getName = function()  
    {  
        alert(this.name);  
    }  
}
```

How to create a New Instance from a Constructor

```
var student = new studentConstructor();  
student.getName(); // will alert "Jack"
```

Literal notation in Javascript Object

```
var student = {  
    name: "Jack",  
    age: 12,  
    getName: function()  
    {  
        alert(this.name);  
    }  
};  
student.getName(); //will alert "Jack"
```

Object.create() in Javascript Object

```
var student = {  
    name: "Jack",  
    age: 12,  
    getName: function()  
    {  
        alert(this.name);  
    }  
};  
var st = Object.create(student);  
st.getName();
```

Accessing Object Properties in JavaScript

We can access Object Properties in JavaScript using either the dot(.) notation or the bracket[] notation

```
var student = {};  
student.name = "Jack"; // access via . notation  
student["age"] = 12; // access via [] brackets  
alert(student["name"]); // will alert "Jack"  
alert(student.age); // will alert 12  
alert(student.name); // will alert "Jack"  
alert(student["age"]); // will alert 12
```

Object contain another Object

```
var student = {  
    name: "Jack",  
    age: 12,  
    classTeacher: {  
        firstname: "John",  
        lastname: "Peter"  
    }  
};  
alert(student.classTeacher.firstname);  
alert(student['classTeacher']['firstname']);  
alert(student.classTeacher['firstname']);  
alert(student['classTeacher'].firstname);
```


Forms

Trigger/Call Javascript

```
<!DOCTYPE html>
<html>
<body>

<p>Enter value and click "Submit" to submit the form:</p>

<form id="frm1">
  Enter value: <input type="text" name="fname"><br>
  <input type="button" onclick="myFunction()" >
</form>

<script>
function myFunction() {
  //.....
}
</script>

</body>
</html>
```

This Week

- Review Slides
- Read Associated Chapters
- Work through Javascript Examples
 - ▷ Update GitHub Account/Webpage
- **Group Project (Christmas Theme)**
 - **Website & Javascript Game**
- **Demonstrate Individual Projects**

Summary

- Overview of Objects in Javascript
- Javascript Forms
- Hands-On/Practical

Questions/Discussion