IS 242 Web Application Development 1

Lecture 9: Introduction to JavaScript (Part 2)

Outlines of today's lecture

- Functions

- Conditional statements

- Loops

- Popup windows

- Events

JavaScript Functions

- A JavaScript function is **a block of code designed to perform a particular task.**
- A function will be executed by an **event or by a call** to the function.
- JavaScript Function Syntax
- A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by **parentheses** ().
- Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
- The parentheses may include parameter names separated by commas: (*parameter1, parameter2, ...*)
- The code to be executed, by the function, is placed inside curly brackets: {}

```
functionName(parameter1, parameter2, parameter3)
{
code to be executed
}
```

The return Statement

- functions that are going to return a value must use the **return** statement.
- The example below returns **the product of two numbers** (a and b):

```html
<html>
  <head>
      <script type="text/javascript">
          function product(a,b)
              {
                return a*b;
              }
      </script>
  </head>

  <body>
      <script type="text/javascript">
          document.write(product(4,3));
      </script>
  </body>
</html>
```

Conditional Statements

- In JavaScript we have the following conditional statements:

- **if statement** – use this statement to execute some code only if a specified condition is true

- **if...else statement** – use this statement to execute some code if the condition is true and another code if the condition is false

- **if...else if....else statement** – use this statement to select one of many blocks of code to be executed

- **switch statement** – use this statement to select one of many blocks of code to be executed

- Note that if is written in lowercase letters. Using uppercase letters (IF) will generate a JavaScript error!

1. if Statement

```
if (condition)
  {
   code to be executed if condition is true
  }
```

Example:
```
<html>
  <head><title>Hello World</title></head>
  <body>
    <script>//Write a "Good morning" greeting if
   //the time is less than 12
   var d=new Date();var time=d.getHours();
     if(time<12)  {
        document.write("<b>Good morning</b>");
     }
    </script>
  </body>
</html>
```
6

2. If else Statement

- Use the if....else statement to execute some code if a condition is true and another code if the condition is not true.

Syntax

```
if (condition)
   {
   code to be executed if condition is true
   }
 else
   {
   code to be executed if condition is not true
   }
```

If..else Statement-Example

```
<script type="text/javascript">
  //If the time is less than 12, you will get a "Good
  morning" greeting.
  //Otherwise you will get a "Good afternoon" greeting.

  var d = new Date();
  var time = d.getHours();
  if (time < 12)
    {
    document.write("Good morning!");
    }
  else
    {
    document.write("Good afternoon!");
    }
</script>
```

3. If…else if…else Statement

- Use the if....else if...else statement to select one of several blocks of code to be executed.

Syntax

```
if (condition1)
  {
  code to be executed if condition1 is true
  }
else if (condition2)
  {
  code to be executed if condition1 is false and condition2 is true
  }
else
  {
  code to be executed if neither condition1 nor condition2 is true
  }
```

If…else if…else Statement–Example

```
<script type="text/javascript">
  var d = new Date()
  var time = d.getHours()
  if (time<12)
      {
        document.write("<b>Good morning</b>");
      }
  else if (time>=12 && time<19)
      {
        document.write("<b>Good afternoon</b>");
      }
  else
      {
        document.write("<b>Good night!</b>");
      }
</script>
```

4. Switch Statement

- Use the switch statement **to select one of many blocks** of code to be executed.

Syntax

```
switch(n)
{
case 1:
            execute code block 1
            break;
case 2:
            execute code block 2
            break;
default:
        code to be executed if n is different from case 1 and 2
}
```

Switch Statement-Example

```
<script type="text/javascript">
  //You will receive a different greeting based
  //on what day it is. Note that Sunday=0,
  //Monday=1, Tuesday=2, etc.

  var d=new Date();
  var theDay=d.getDay();
  switch (theDay)
      {
        case 5:
                  document.write("Finally Friday");
                  break;
        case 6:
                  document.write("Super Saturday");
                  break;
      case 0:
                  document.write("Sleepy Sunday");
                  break;
      default:
                  document.write("I'm looking forward to this weekend!");
  }
</script>
```

For loop

- Loops execute a block of code **a specified number of times**, or while a specified condition is true.

- The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
    code block to be executed
}
```

For Loop-Example

- Loop through the six **different HTML Headings.**

```
<html>
  <body>

        <script type="text/javascript">
         for (i = 1; i <= 6; i++)
         {
          document.write("<h" + i + ">This is heading " + i);
          document.write("</h" + i + ">");
         }
        </script>

    </body>
</html>
```

14

While Loop

```
while (condition)
   {
   code to be executed
   }
```

Example:

```
<html>
  <body>
  <script type="text/javascript">
   var i=0;
   while (i<=5)
   {
   document.write("The number is " + i);
   document.write("<br />");
   i++;
   }
  </script>
  </body>
</html>
```

The do...while Loop

Syntax

```
do
   {
    code to be executed
   } while (condition);
```

Example

```
<html>
  <body>
  <script type="text/javascript">
          var i=0;
          do
             {
               document.write("The number is " +
   i);

               document.write("<br />");
                i++;
             } while (i<=5);

  </script>
  </body>
</html>
```

16

The for...in Statement

- The for...in statement loops through the properties of an object.

Syntax

```
for (variable in object)
    {
    code to be executed
    }
```

Example

```
var person={fname:"John",lname:"Doe",age:25};

for (x in person)
{
document.write(person[x] + " ");
}
```

The result of *text* will be:

John Doe 25

The continue Statement

- The **continue** statement will **break the current loop** and **continue with the next value.**

```
<html>
  <body>
    <script type="text/javascript">
     var i=0
     for (i=0;i<=5;i++)
     {
     if (i==3)
       {
       continue;
       }
     document.write("The number is " + i);
     document.write("<br />");
     }
    </script>
  </body>
</html>
```

The result of *text* will be:

```
The number is 0
The number is 1
The number is 2
The number is 4
```

Popup Boxes

JavaScript has three kinds of popup boxes:

- Alert box

- Confirm box

- Prompt box

Alert Box

- An **alert box** is often used if you want to make sure information comes through to the user.
- When an alert box pops up, the user **will have to click "OK"** to proceed.

Syntax
```
alert("sometext");
```

Example
```html
<html>
  <head>
  <script type="text/javascript">
  function show_alert()
    {
      alert("I am an alert box!");
    }
  </script>
  </head>
  <body>
    <input type="button" onclick="show_alert()" value="Show alert box"
/>
  </body>
  </html>
```
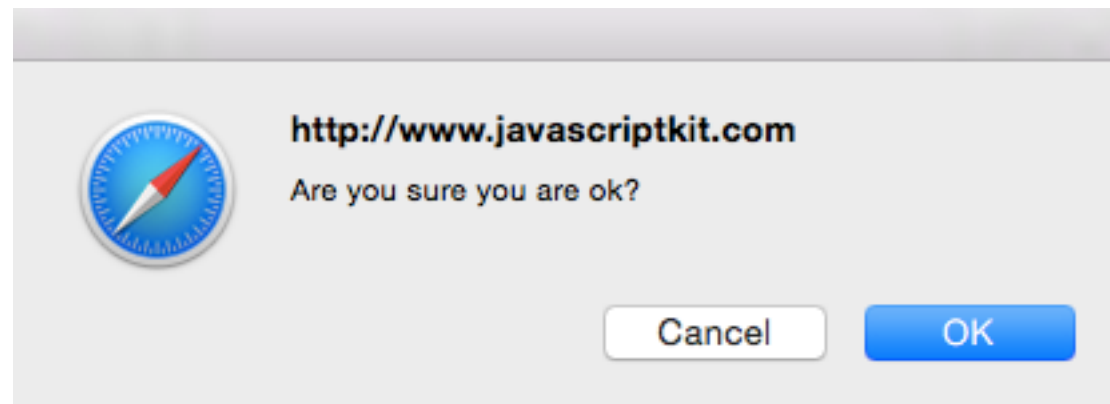

20

Alert Box–Another Example

```html
<html>
<head>
    <script type="text/javascript">
      function disp_alert()
         {
         alert("Hello again! This is how we" + '\n' + "add line breaks to an alert box!");
            }
      </script>
</head>
<body>
    <input type="button" onclick="disp_alert()" value="Display alert box" />
</body>
</html>
```
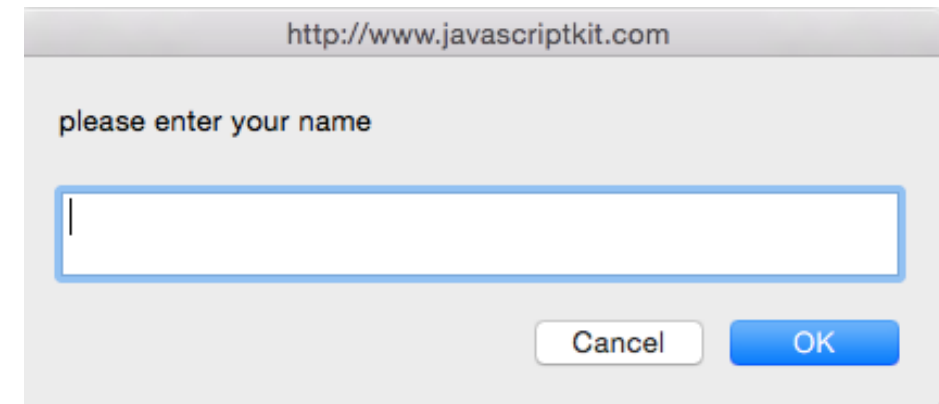
Confirm Box

- A confirm box is often used if you want the user to **verify or accept something.**
- When a confirm box pops up, the user will have to click either **"OK"** or **"Cancel"** to proceed.
- If the user clicks **"OK"**, the box returns **true.** If the user clicks **"Cancel"**, the box returns **false.**

- Syntax
  ```
  confirm("sometext");
  ```

Confirm Box-Example

```html
<html>
  <head>
      <script type="text/javascript">
            function show_confirm()
               {
                     var r=confirm("Press a button");
                     if (r==true)  alert("You pressed OK!");
                     else    alert("You pressed Cancel!");
               }
      </script>
  </head>
  <body>
   <input type="button" onclick="show_confirm()" value="Show confirm box" />
  </body>
  </html>
```

Prompt Box

- A prompt box is often used if you want the user to **input a value before entering a page.**

- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

- If the user clicks **"OK"** the box **returns the input value.** If the user clicks **"Cancel"** the box returns **null.**

- Syntax
    prompt("sometext","defaultvalue");

Prompt Box-Example

```
<html>
    <head>
     <script type="text/javascript">
         function show_prompt()
            {
               var name=prompt("Please enter your name","Harry Potter");
                if (name!=null)
                         document.write("Hello " + name + "! How are you
today?");
            }
     </script>
   </head>
   <body>
           <input type="button" onclick="show_prompt()" value="Show prompt
   box" />
   </body>
</html>
```

JavaScript Events

- HTML events are "things" that **happen** to **HTML elements**.

- Events are actions that can be detected by JavaScript.

- When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.

- For example, we can use the **onClick event** of a button element to indicate that a function will run when a user clicks on the button. We define the events in the HTML tags.

- Examples of events:
  - A mouse click
  - A web page or an image loading
  - Mousing over a spot on the web page
  - Selecting an input field in an HTML form

- Note: Events are normally used in combination with functions, and the function will not be executed before the event occurs!

JavaScript Events−Example

- The example below displays the date when a button is clicked:

```
<html>
<head>
  <script type="text/javascript">
        function displayDate()
          {
                document.getElementById("demo").innerHTML=Date();
          }
  </script>
</head>
<body>
    <h1>My First Web Page</h1>
    <p id="demo"></p>
    <button type="button" onclick="displayDate()">Display
  Date</button>
   </body>
</html>
```

References

- www.w3schools.com
- Deitel & Deitel (2011). *Internet and World Wide Web How to Program, 5th Edition, Harvey & Paul Deitel & Associates.*