Mouse events

Some of the most interesting things you can do with JavaScript are event-driven. Event-driven means that a particular action causes something to happen. JavaScript has clicks that can detect events such as click, onmouseover, onmouseout. This page illustrates the mouse over and mouse out events.
Mouse events

Some of the first interesting things you can do with JavaScript are event-driven. Event-driven means that a particular action causes something to happen. JavaScript has clicks that can detect events such as onclick, onmouseover, onmouseout. This page illustrates the mouse over and mouse out events.

```html
<!DOCTYPE html>
<html>
<head>
  <title>Event</title>
  <meta charset="utf-8">
</head>
<body>
<h1>Mouse events</h1>
<p>Some of the first interesting things you can do with JavaScript are event-driven means that a particular action causes something to happen. JavaScript has events such as onclick, onmouseover, onmouseout. This page illustrates the mouse over and mouse out events.</p>
<p>
</p>
</body>
</html>
```

Two images - I start out with CISa.gif as an image and give it the name CISimage. The two events onmouseover and onmouseout you bring up different images in the image area named CISimage.
The onmouseover event in this case brings up an alert.

```html
<!DOCTYPE html>
<html>
<head charset="utf-8">
<title>Mouse Event</title>

<style type="text/css">
p {
  margin-left: 500px;
  margin-right: 500px;
}

.fontverylarge
{
  font-size: 36pt;
  text-align: center;
  background-color: black;
  color: red;
}
</style>
</head>
<body>

<p class="fontverylarge" onmouseover = "alert('testing mouse over!');">Mouse over here!</p>
</body>
</html>
```
An alternative used more often a few years ago.
Another alternative. Compare to the example on the top.
An example of the onclick event.
Other examples using `onmouseover` and `onmouseout` - in this case I am using them to change background color.
Using the onfocus event.
This is a header before the function

Hello world

It looks like you haven't started this in a while. Do you want to clean it up for a fresh, new experience? And by the way, welcome back!
I call the `basicHelloWorld` function - it is coded in this case in the `<head>` section.

```
<xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Introduction to functions</title>
<body>
</body>
</html>
```

This example is done using XML.
This is a header before the function
```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Introduction to functions</title>
  </head>
  <body>
    <p>
      { name: 'Ann', color: 'pink' }
    </p>
    <script type="text/javascript">
      function basicHello(passedName)
      { alert('Hello ' + passedName); }
    </script>
  </body>
</html>
```
This is a header before the function
In the script, I set up a variable theName which I fill from the user entry at the prompt. I then send theName to the basicHello function. Note that I send theName and I receive passedName.
Position sends the first field to send to the first field to receive and the second field to send to the second field to receive.
Note the first thing sent is firstName which is received in lastName and so we end up reversing the names.
This is a header before the function

The name is showing last and then first because of position sending/receiving.
```html
<!DOCTYPE html>

<html>
    
    <head>
        <meta charset="utf-8">
        <title>Introduction to Functions</title>
        
        <style type="text/css">
            body {
                background: black;
                color: pink;
            }
        </style>
    
    </head>
    
    <body>
        
        <script type="text/javascript">
            function basicHello(lastName, firstName) {
                alert(lastName + firstName + " Hello");
            }
        </script>
        
        <h1>This is a header before the function</h1>
        
        <!-- This is a comment -->
        
        var firstName;
        var lastName;
        
        firstName = window.prompt("Enter your first name ", ");
        lastName = window.prompt("Enter your last name ", ");
        
        basicHello(firstName, lastName);  
        
        <h1>This is a header after the function</h1>
        
    </body>
</html>
```
I entered Jen and Adams.
This is a header before the function
This is a header before the function

On this one I had Jom on the last name and I had it twice and you can see the output.
```html
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <title>Introduction to functions</title>
</head>
<body>
  
  <style>
    body {
      background: black;
    }
    h1 {
      color: pink;
    }
  </style>

  <script type="text/javascript">
    function basicHello(firstName, lastName) {
      alert("Hello " + firstName + " " + lastName);
    }
  </script>

  <h1>This is a header before the function</h1>
  
  <script type="text/javascript">
    var firstName;
    var lastName
    firstName = window.prompt("Enter your first name ", "");
    lastName = window.prompt("Enter your last name ", "");
    basicHello(firstName, lastName)
  </script>

  <h1>This is a header after the function!</h1>
</body>
</html>
```
Note that lastName was the field that was sent. I need to do some research on this issue.
Without `var` the definition is global.
This is a header before the function

Enter your first name: Ann
Enter your last name: Dwork

Click
This is a header before the function

Enter your first name:  
Enter your last name:  

Help: Ann Brown
When using text boxes I need to work with the value property.
This is a header before the function

Enter your first name: Jane
Enter your last name: Andrews
Click:

It looks like you haven’t started Eudora in a while. Do you want to queue it up for a fresh, new experience? And by the way, welcome back!
This is a header before the function

Enter your first name: [Field]
Enter your last name: [Field]
[Submit]
```html
"utf-8">
<title>ction to functions</title>
{text/css}>
This : black;

"text/javascript">
siHello()

it.info.nameOut.value = "Hello " + document.info.firstIn.value + " " + lastIn.value

header before the function</h1>
" id="info" name="info">
  first name:
  text name="firstIn" size="25" />
</p>
  last name:
  text name="lastIn" size="25" />
</p>
ishment name="toClick" value="Click" onclick="basicHello()" />

"text" name="nameout" size="50" />
</p>
```
You need to get the value property in info. Here I get it with the alert. In the next example I will do it when I call the function.
Here I send the name.
Using a variable ans to hold the value in the info that was sent when I sent aname.
When you use the return, you are returning the result that you created in the function.

When I say `document.write(afunction(info))` I am saying `document.write` what is returned from the function so I wrote

*Your home town is Hingham*
Before the function is called

Your home town is Hingham

After the function is called

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script type="text/javascript">
function afun() { 
   return("Your home town is " + info);
}
</script>
</head>
<body>
<p>Before the function is called</p>
<script type="text/javascript">
var info = prompt("Enter your home town ", "");
document.write(afun(info));
</script>
<p>After the function is called</p>
</body>
</html>
```
Here I sent nameOut which is empty up to the function and then I fill it within the function.
Here I have two functions. The onclick called the basicHello function sent the firstIn, lastIn, nameOut.
When I got there I say that namOut.value will equal a literal concatenated with the results returned from catNames when I sent firstIn, lastIn.
The catNames receives the data and creates the fullName which it returns. So when it returns to basicHello we concatenate Hello with the fullName that was returned.
Here is another variation where onclick I do JavaScript: which takes the values and creates nameOut.value.
The return from basicHello is msg which is assigned to nameOut.value.
This is a header before the function

Enter your first name: John
Enter your last name: Doe
Click
Hello, John Doe

It looks like you haven’t started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
The basicHello function has a return that returns msg and assigns it to nameOut.value.
```html
<!DOCTYPE html>
<html>
<head>
    <title>Introduction to functions</title>
    <style type="text/css">
        body { background: black; }
        p { color: pink; }
    </style>
</head>
<body>
    <script type="text/javascript">
        function basicHello(firstName, lastName)
        {
            var msg = "";
            msg = "Hello " + firstName.value + " " + lastName.value;
            alert(msg);
            return msg;
        }
    </script>

    <h2>This is a header before the function</h2>
    <form>
        <p>Enter your first name:</p>
        <input type="text" id="firstName" size="20" />
        <br>
        <p>Enter your last name:</p>
        <input type="text" id="lastName" size="20" />
        <br>
        <input type="button" id="button" onclick="basicHello(firstName, lastName)"
            value="Click" name="value" />
    </form>
</body>
</html>
```
I call function calc and send it data. The return returns the calculation and assigns it to ans. Then I document.write the result in ans.
```javascript
function calc(first, second, third) {
    return (first * second) / third;
}
```

```javascript
var num1 = 0;
var num2 = 0;
var num3 = 0;
num1 = prompt("Enter first number ", 0);
num2 = prompt("Enter second number ", 0);
num3 = prompt("Enter third number ", 0);
ans = calc(num1, num2, num3);
document.write("The answer is ", ans);
```
I am sending the value of the fields from the text box. The calculation is returned and stored as the value in rslt.
I am setting up these empty divisions with an id.

Now I am putting information into these divisions.

I was able to reserve areas and then fill them later in the code.
The divisions do not show until I get to the DOM where I wrote things into the divisions with the specified id.

The code snippet provided is for multiplying two numbers taken in through prompts. It includes HTML and JavaScript code to achieve this functionality.

```html
<!DOCTYPE html>
<html>
<head>
<title>Multiply</title>
<meta charset="utf-8">
</head>
<body>
<h1>Multiply two numbers taken in through prompts</h1>

<script type="text/javascript">
var ans = 0;
var firstnum = 0;
var secondnum = 0;

firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);

ans = firstnum * secondnum;
document.getElementById('onlyAns').innerHTML = ans + "<br>

document.getElementById('putHere').innerHTML = "The answer is " + ans;
</script>
</body>
</html>
```
Multiply two numbers taken in through prompts.

```html
<!DOCTYPE html>
<html>
<head>
<title>Multiply</title>
<meta charset="utf-8"/>
</head>
<body>
<h1>Multiply two numbers taken in through prompts</h1>
<div id="onlyAns"></div>
<script type="text/javascript">
var ans = 0;
var firstnum = 0;
var secondnum = 0;
firstnum = prompt("Enter the first number", 0);
secondnum = prompt("Enter the second number", 0);
ans = firstnum + secondnum;
document.getElementById('onlyAns').innerHTML = ans + "<br>";
document.getElementById('puthere').innerHTML = "The answer is " + ans;
</script>
</body>
</html>
```
The answer is 20

```
<html>
<head>
<title>Decision</title>
<meta charset="utf-8">
<style type="text/css">
  h1
  {
    text-align:center;
  }
</style>
</head>
<body>
<h1>Results</h1>
<div id="rslt"></div>
<script type="text/javascript">
  var ans = 0;
  var firstnum = 0;
  var secondnum = 0;
  var whattodo;
  firstnum = prompt("Enter the first number",0);
  secondnum = prompt("Enter the second number",0);
  whattodo = prompt("Enter * or /","" );
  if (whattodo == "+")
  {
    ans = firstnum * secondnum;
  }
  else
  {
    if (whattodo == "/")
    {
      ans = firstnum / secondnum;
    }
    else
    {
      ans = "Problem";
    }
  }
  document.getElementById("rslt").innerHTML = "The answer is " + ans;
</script>
</body>
</html>
```
The answer is 12

This is whatever. This is whatever. This is whatever. This is whatever. This is whatever. This is whatever. This is whatever.
Results

The answer is 12

This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever... This is whatever...
Replace the code below with your own JavaScript code:

```javascript
let playerPoints = 0;
let currentPoints = 0;
let totalPoints = 0;

while (true) {
  let inputPoints = parseInt(prompt("Enter the number of points you have earned", 0));
  if (inputPoints < 0) {
    alert("Invalid input!");
  } else {
    playerPoints += inputPoints;
  }
  totalPoints += currentPoints;
  currentPoints = inputPoints;
  if (totalPoints < 50) {
    alert("You can choose a prize from group D");
  } else {
    alert("You can choose a prize from group A");
  }
}
```

This code prompts the user to enter the number of points they have earned, calculates the total points, and checks which group of prizes they qualify for based on their total points.
Loop followed by a decision

This is a JavaScript program that prompts the user for input. I have inserted some alert() commands in here to show the programmer the program's progress through the program. This is an excellent debugging tool. If things are not working the way you want them to, put in an alert() and see what is really happening.

You can choose a prize from group A
Extra credit if after looking at the next few slides you write an explanation of this code.
Note that msg plus other things is assigned to msg so that msg has what was in msg plus the new thing I want to add.
Math Facts

1+1=2

OK
Math Facts

```html
<html>
<head>
<meta charset="utf-8">
<title>This is a math facts program</title>
</head>
<body>
<h2>Math Facts</h2>
<div id="facts"></div>
<script type="text/javascript">
var ct = 1;
var ctin;
var msg;
msg = document.getElementById("facts").innerHTML;
while (ct <= 3) {
    ctin = 1;
    while (ctin <= 3) {
        ans = ct + ctin;
        msg = msg + ct + " + " + ctin + " = " + ans + "<br>";
        alert(msg);
        ctin = ctin + 1;
    }
    ct = ct + 1;
}
</script>
</body>
</html>
```
Math Facts

If I do not have msg = msg + calculation
but instead have msg = calculation.
Math Facts

3 - 6

The end of the math facts

So I end up showing just the last calculation.
ASSIGNMENT: I want you to create this using HTML5/JavaScript. You must use a function.