

# MOUSE EVENTS

**PROF. DAVID ROSSITER**

# AFTER THIS PRESENTATION

- You'll know how to use mouse events

# WE WILL LOOK AT

`onclick`

---

`onmousedown`

---

`onmouseup`

---

`onmouseover`

---

`onmouseout`

# MOUSE EVENTS

- The most commonly used mouse events:
  - `onclick` - when the user clicks on an object
  - `onmousedown` - when the user presses down the mouse button
  - `onmouseup` - when the user lets go of the mouse button
- `onclick` = `onmousedown` followed by `onmouseup`

```
<html><body><script>
  function good_choice() { alert("Good choice!"); }
  function bad_choice() { alert("I don't agree!"); }
</script>
<h1>Click on the best social network...</h1>



</body></html>
```

# MORE MOUSE EVENTS

鼠标放在object上

- onmouseover - mouse is moved over an object
- onmouseout - mouse is moved away from an object

```
<html><body><script>
  function change_colour( new_colour ) {
    document.getElementById("myDiv")
      .style.background=new_colour;
  }
</script>
<div id="myDiv"
  style="position:absolute; background:yellow;
  left:300; top:100; width:300; font-size:52pt"
  onmouseover="change_colour('red');"
  onmouseout="change_colour('yellow');">
  Move your mouse over this ...
  then move it out...
</div>
</body></html>
```

# TIMER EVENTS

PROF. DAVID ROSSITER



# AFTER THIS PRESENTATION

- You'll know how to use timer events

# WE WILL LOOK AT

`setTimeout`

`setInterval`

---

`clearTimeout`

`clearInterval`

# TIMERS


- Timers are very useful for dynamic web page behaviour
- Set a timer like this:

```
var the_timer;  
the_timer=setTimeout(do_something, 1000);
```

- do\_something() will be executed 1 second later
- The value 1000 is in milliseconds, so 1000=1 second

```
<html><head><script>
  var wait_duration;
  function set_things_up() {
    wait_duration = prompt("How long do you " +
                          "want to sleep?");
    setTimeout(show_wake_up_message, wait_duration );
  }
  function show_wake_up_message() {
    alert("WAKE UP! WAKE UP! WAKE UP!!");
  }
</script></head>
<body onload="set_things_up()">
  <h1>Alarm clock example</h1>
</body>
</html>
```

no need to change to integer?



# TIMER EXAMPLE - MOVING AN IMAGE

```
<html><head><script>
  var the_timer, x_position = 0, the_image;
  function set_timer() {
    the_image=document.getElementById("stones_image");
    x_position=x_position+1;
    the_image.style.left=x_position;
    the_timer = setTimeout(set_timer, 50);
  }
</script></head>
<body onload="set_timer()">
  
</body></html>
```

recursive

# STOPPING A TIMER

- If a timer is started like this:

```
var the_timer;  
the_timer=setTimeout(do_something, 1000);
```

- Then stop it like this:

```
clearTimeout(the_timer);
```

```
<html><head><script>
  var the_timer, x_position = 0, the_image;
  function set_timer() {
    the_image = document.getElementById("stones_img");
    x_position = x_position + 1;
    the_image.style.left = x_position;
    the_timer = setTimeout(set_timer, 50); }
</script></head>
<body onload="set_timer()">
  
  <button onclick="clearTimeout(the_timer)">
    Stop!</button>
</body></html>
```

# SETINTERVAL

- `setInterval()` repeatedly does something
- Start it like this:

```
var the_timer;  
the_timer=setInterval(do_something, 2000);
```

- `do_something()` will be executed every 2 seconds
- To stop it:

```
clearInterval(the_timer);
```



```
<html><head><script>
  var the_timer, x_position = 0, the_image;
  function do_timer(){
    the_image = document.getElementById("stones_img");
    x_position = x_position + 1;
    the_image.style.left = x_position;
  }
</script></head>
<body onload="the_timer=setInterval(do_timer, 50)">
  
  <button onclick="clearInterval(the_timer)">
    Stop!</button>
</body></html>
```

# ADDING EVENTS USING JAVASCRIPT

PROF. DAVID ROSSITER

# AFTER THIS PRESENTATION

- You'll able to set up events using JavaScript

# WE'LL LOOK AT

`addEventListener()`

---

`removeEventListener()`

# ADDING A HANDLER USING HTML

- Adding an event to an element in HTML:

```
<html><head><script>  
  function do_something() {alert("Page has loaded");}  
</script></head>  
<body onload="do_something()"></body>  
</html>
```

- `do_something()` is the *event handler* for this event

# ADDING A HANDLER USING JAVASCRIPT

- We can also add an event to an element: using javascript

```
<html><body id="theBody">
  <script>
    function do_something() { alert("Page has loaded") }
    window.onload = do_something;
  </script>
</body></html>
```

# ADDING A HANDLER USING JAVASCRIPT

- Another way: using javascript

```
<html><body>
  <script>
    function do_something() { alert("Page has loaded") }
    window.addEventListener("load", do_something);
  </script>
</body></html>
```

加载多个events

# IF YOU HAVE MORE THAN ONE EVENT HANDLER

how to do?

- Event handlers are stored in an array
- When an event happens, all the handlers are executed
- They are executed in the order they are added

events will execute one by one





# REMOVING AN EVENT HANDLER

- To remove an event handler:

```
var theBody = document.getElementById("theBody");  
theBody.removeEventListener("load", do_something);
```

```
<html><body>
  <button id="btn0" onclick=" alert('Hello!') ">
    Click Me!</button><br>
  <button id="btn1">Remove Listener</button>
  <script>
    function do_something() { alert('Clicked'); }

    var btn0 = document.getElementById("btn0");
    btn0.addEventListener("click", do_something);

    var btn1 = document.getElementById("btn1");
    btn1.addEventListener("click", function() {
      btn0.removeEventListener("click", do_something);
    });
  </script></body></html>
```