



# **COMP 4021**

## **Internet Computing**

### **Javascript 3**

Dr. Kenneth LEUNG

Slides created by Dr. David ROSSITER

# Revisiting HTML Tables

- Before we look at more JavaScript we need to consider some of the HTML we will need in the next lab

# HTML Tables

```
<table border="1" width="200">
```

*This line of HTML defines one of the boxes in the row*

```
<tr>
```

```
<td height="100" width="100"> One </td>
```

```
<td height="100" width="100"> Two </td>
```

```
</tr>
```

```
<tr>
```

```
<td height="100" width="100"> Three </td>
```

```
<td height="100" width="100"> Four </td>
```

```
</tr>
```

```
</table>
```

*There will be two rows*

This HTML will set up a table with 2 rows, each row having 2 boxes

## Simple 2 \* 2 table structure

One	Two
Three	Four

 Done

 Internet

# HTML Tables

- A table cell can contain anything – not just text
  - A cell can contain another table composed of multiple rows and columns
- For example, on the following slides images are put into the cells

<h2>My favourite characters!</h2>

<table border="10" width="100">

# HTML Tables

<tr>

<td height="100" width="100"> 1.  </td>

</tr>

<tr>

<td height="100" width="100"> 2.  </td>

</tr>

<tr>

<td height="100" width="100"> 3.  </td>

</tr>

</table>

# HTML Tables

- The images have the same width because the width attribute of every row has been specified as 100pt



# HTML Tables

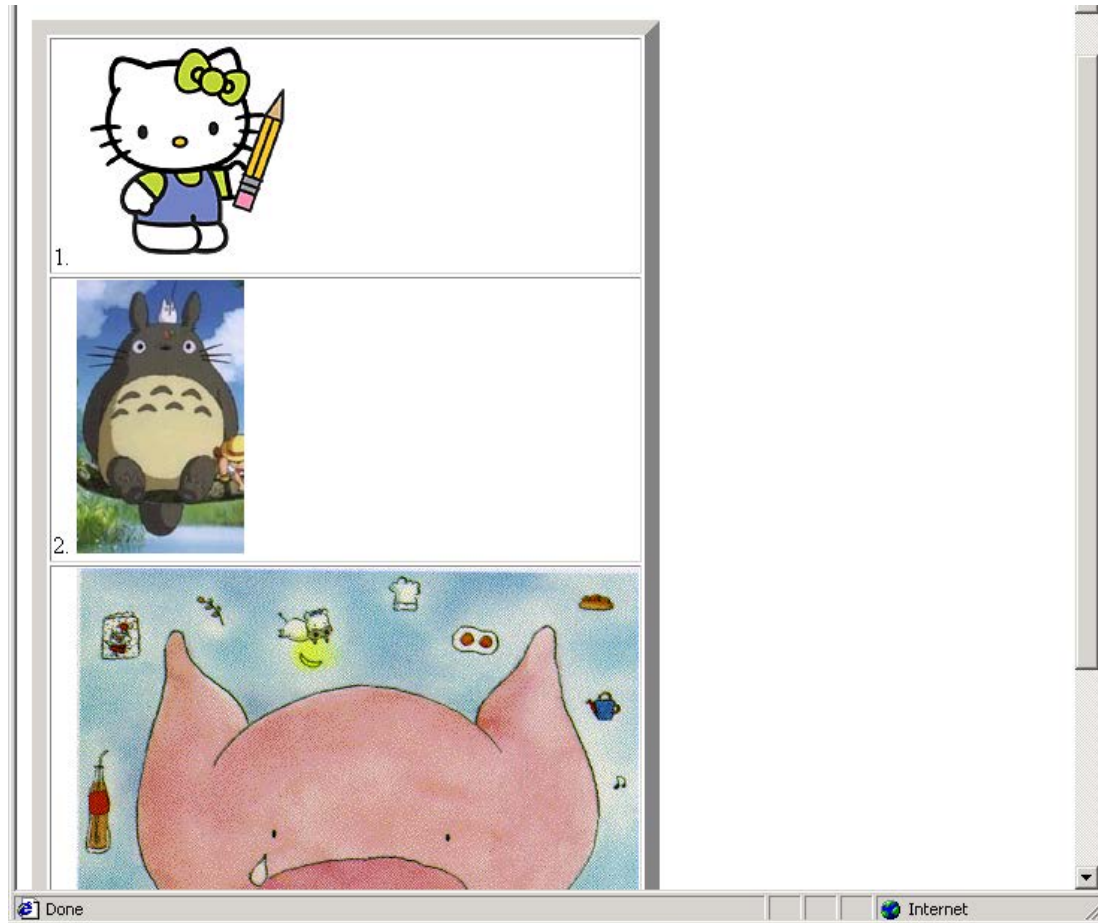
What happens if we don't tell the browser to use a specific width/height?

```
<h2>My favourite characters!</h2>
<table border="10">
  <tr>
    <td> 1.  </td>
  </tr>
  <tr>
    <td> 2.  </td>
  </tr>
  <tr>
    <td> 3.  </td>
  </tr>
</table>
```



# HTML Tables

- The images are shown at their 'natural' size
- Table is automatically expanded to fit them in



# Revisiting Intervals

- Previously we looked at how to use timers/ intervals to tell the browser to do something at a later time
- Multiple timers/ intervals can be used concurrently

# Two setTimeout Timers

```
<script language="JavaScript">  
  
var wait_duration, timer1, timer2;
```

```
function set_things_up() {  
    wait_duration=prompt("How long would you like to sleep?", "");  
    timer1=setTimeout("show_wake_up_message()", wait_duration );  
  
    wait_duration=prompt("How long until your next lecture?", "");  
    timer2=setTimeout("show_lecture_message()", wait_duration );  
}
```

```
function show_wake_up_message() {  
    alert("WAKE UP! WAKE UP! WAKE UP!!");  
}
```

```
function show_lecture_message() {  
    alert("GO TO LECTURE! GO TO LECTURE!");  
}
```

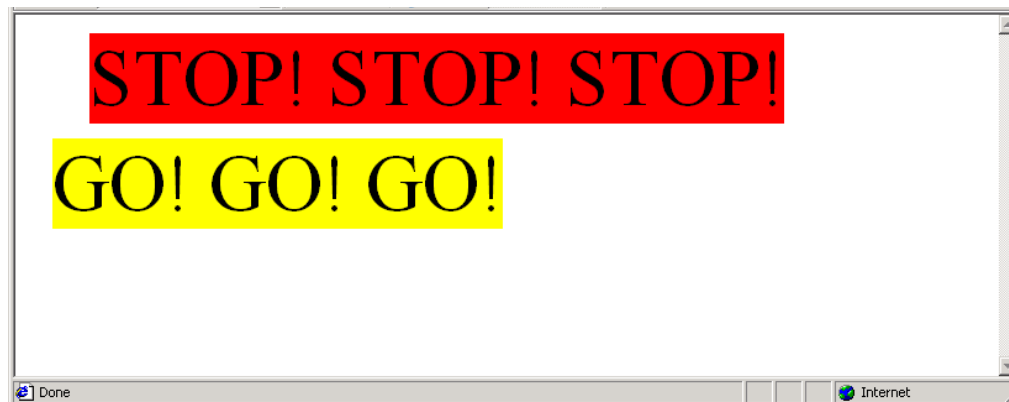
 <body **onload="set\_things\_up()"** >

<h1>Double alarm clock example</h1>

</body>

# Another Two-Timer Example

- See the web site for an example of using two timers, each timer moving a different layer at a different speed



# Key Events



- Previously we learnt about mouse events
- Now we consider key events
- For key events we are usually interested in knowing exactly which key has been pressed
- The way to handle this is a bit different to handling mouse events – for example, a keyboard event can't be applied to one particular object in the web page

# Handling Key Events

- Whenever a key is pressed down when the web page is loaded the JavaScript function `handle_key_press()` will be executed

```
<body onkeydown="handle_key_press(event)">
```

```
...
```

```
</body>
```

# Handling Key Presses

- The following function recognises what key has been pressed and react appropriately

```
function handle_key_press(key_event){  
    var key_pressed_number, key_pressed_letter;  
  
    key_pressed_number=key_event.keyCode;  
    alert("The key you just pressed is key number " +  
        key_pressed_number);  
  
    key_pressed_letter=String.fromCharCode(key_pressed_number);  
    alert("So that means that you pressed the "  
        + key_pressed_letter + " key");  
  
    ... do something depending on which key was pressed ...  
}
```

# The Event Object

- Properties of the event object which are useful for handling key events:
- `event.keyCode` - returns value of key pressed
- `event.shift` - indicates whether “shift” is pressed
- `event.ctrl` - indicates whether “ctrl” is pressed
- `event.alt` - indicates whether “alt” is pressed



# ASCII Table

- Key press values are ASCII

ASCII Hex Symbol	ASCII Hex Symbol	ASCII Hex Symbol	ASCII Hex Symbol
0 0 NUL	16 10 DLE	32 20 (space)	48 30 0
1 1 SOH	17 11 DC1	33 21 !	49 31 1
2 2 STX	18 12 DC2	34 22 "	50 32 2
3 3 ETX	19 13 DC3	35 23 #	51 33 3
4 4 EOT	20 14 DC4	36 24 \$	52 34 4
5 5 ENQ	21 15 NAK	37 25 %	53 35 5
6 6 ACK	22 16 SYN	38 26 &	54 36 6
7 7 BEL	23 17 ETB	39 27 '	55 37 7
8 8 BS	24 18 CAN	40 28 (	56 38 8
9 9 TAB	25 19 EM	41 29 )	57 39 9
10 A LF	26 1A SUB	42 2A *	58 3A :
11 B VT	27 1B ESC	43 2B +	59 3B ;
12 C FF	28 1C FS	44 2C ,	60 3C <
13 D CR	29 1D GS	45 2D -	61 3D =
14 E SO	30 1E RS	46 2E .	62 3E >
15 F SI	31 1F US	47 2F /	63 3F ?

ASCII Hex Symbol	ASCII Hex Symbol	ASCII Hex Symbol	ASCII Hex Symbol
64 40 @	80 50 P	96 60 `	112 70 p
65 41 A	81 51 Q	97 61 a	113 71 q
66 42 B	82 52 R	98 62 b	114 72 r
67 43 C	83 53 S	99 63 c	115 73 s
68 44 D	84 54 T	100 64 d	116 74 t
69 45 E	85 55 U	101 65 e	117 75 u
70 46 F	86 56 V	102 66 f	118 76 v
71 47 G	87 57 W	103 67 g	119 77 w
72 48 H	88 58 X	104 68 h	120 78 x
73 49 I	89 59 Y	105 69 i	121 79 y
74 4A J	90 5A Z	106 6A j	122 7A z
75 4B K	91 5B [	107 6B k	123 7B {
76 4C L	92 5C \	108 6C l	124 7C
77 4D M	93 5D ]	109 6D m	125 7D }
78 4E N	94 5E ^	110 6E n	126 7E ~
79 4F O	95 5F _	111 6F o	127 7F ¯

# Complete Example

```
function handle_key_press(key_event){  
  var letter, para;  
  
  letter= String.fromCharCode(key_event.keyCode); // extract the letter  
  para=document.getElementById("output_paragraph"); // find the paragraph  
  para.innerHTML=letter; // set the content of the paragraph to be the letter  
}
```

```
<body  
  onkeydown="handle_key_press(event)" >  
  <p id="output_paragraph" style="font-size: 96pt">  
    Please type a letter  
  </p>  
</body>
```

# Changing - .innerHTML

- From the previous example:

```
para=document.getElementById("output_paragraph"); // find the element  
para.innerHTML=letter; // change the element
```

- .innerHTML changes the text of something
- You can change the text inside anything that contains text, e.g., paragraph, div, list, header, etc.

# Extending - .innerHTML

- We can also use .innerHTML to find out the text which is already inside an object

```
para=document.getElementById("output_paragraph"); // find the  
element
```

```
para.innerHTML=para.innerHTML + " more text!";  
// add more text to it
```

- The last line of code takes the text that is already there, appends more text to it, and puts the result back

Please type:  
OK I WILL  
TYPE ☐

Done

Internet

# The Image Array

- `document.images[]` is an array containing all images in a web page
  - The first image in the web page is `document.images[0]`
  - The second image is `document.images[1]`, etc.
- Change the `.src` property of the image to change an image. For example:
  - `document.images[0].src="mypicture.jpg"`

# Example Code

```
<body>
<h1>Click on an image to change the .src of that image ...</h1>

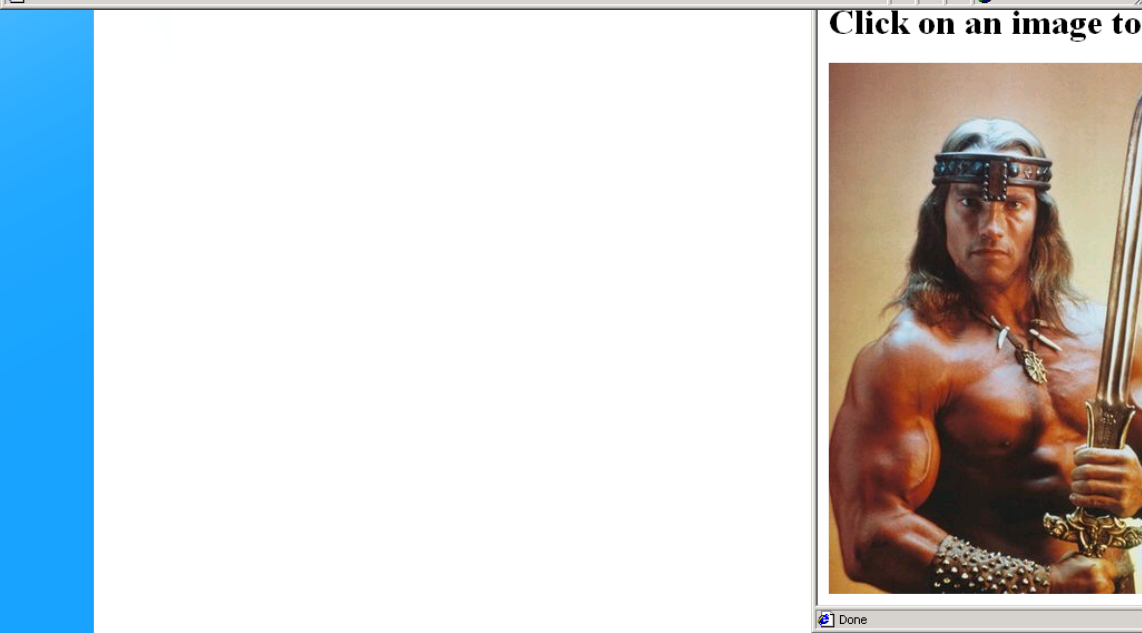




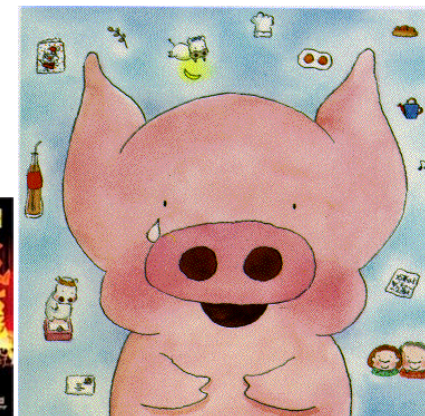

</body>
```



*Click*



**Click on an image to change the .src of that image ...**





# Adding Non-HTML Objects - Embed

- Use **audio** to add audio file to a web page For example:  
`<audio id=" bgmusic" type="audio/mpeg"  
src="bgm.mp3" width="0" height="0"  
autostart="true" ></audio>`
- This tells the browser to load an MP3 file and play it immediately
- However, we usually want exact control using JavaScript

# JavaScript Control over Audio

```
<script language="JavaScript">
```

```
function play_sound() {  
    //stop playing the 'hit' sound file if it is playing  
    document.getElementById( "hit" ).pause();  
  
    //play the 'hit' sound file  
    document.getElementById( "hit" ).play();
```

```
    }  
</script>
```

```
<body onmousedown="play_sound()">
```

```
<h1>Play sound file using JavaScript</h1>
```

If you click on this page, a sound file will be played.

```
<audio id="hit" src="hit.mp3" type="audio/mpeg"  
    width="0" height="0" autostart="false"></audio>
```

```
</body>
```

```
</html>
```