COMP 4021
Internet Computing

JavaScript

Part 3

David Rossiter

Revisiting setTimeout()

- setTimeout is useful to delay an operation, e.g., in game
- However, here, we use setTimeout to illustrate JavaScript asynchrony and how JavaScript engine interacts with the browser via Web API

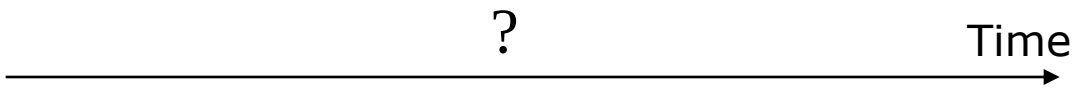
Two setTimeout Timers

```
<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!");   }
</script>

<body onload="set_things_up()“ >
        <h1>Double alarm clock example</h1>
</body>
```

Fill in the Timeline

```
function set_things_up() {
        wait_duration=prompt("How long would you like to sleep?", "");        //  (1)
        timer1=setTimeout("show_wake_up_message()", wait_duration );  //  (2)

        wait_duration=prompt("How long until your next lecture?", "");        //  (3)
        timer2=setTimeout("show_lecture_message()", wait_duration );  }  // (4)

function show_wake_up_message() {
        alert("WAKE UP! WAKE UP! WAKE UP!!");   }     // (5)
function show_lecture_message() {
        alert("GO TO LECTURE! GO TO LECTURE!");   }  // (6)
```

- Timeouts are “multi-threaded”
- The first timeout triggers an alert() which blocks execution until “OK” on the alert box is clicked
- Question: If timer1 is 5 sec and timer2 is 6 sec, user responds to a prompt in 2 sec, can you put 1-6 above on the following timeline?
- See discussion in separate slides



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

Demo

- This example does not have alert messages to block execution of the timers, so the two messages move simultaneously without interference

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 from 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 …
}
```

``Usually'' the ASCII code but it is browser dependent

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>
```

Demo

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.

Demo

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

Demo

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"

- It is not advised to use images[] since the order of the images is defined by the browser and changes to the HTML page can make your program non-functional

Example Code

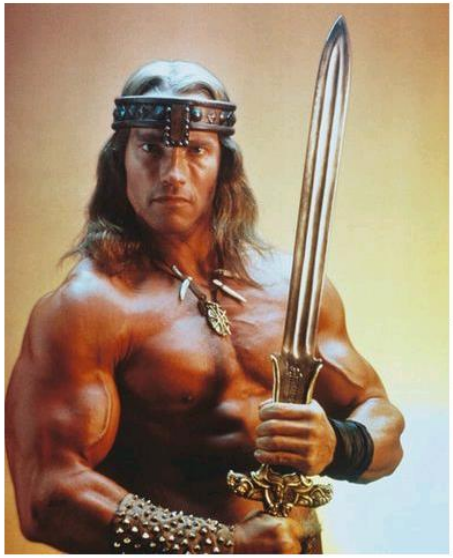
```
<body>
<h1>Click on an image to change the .src of that
   image ...</h1>

<img src="arnold.jpg" onclick="document.images[0].src =
   'totoro.jpg' ">

<img src="jackie_chan.jpg" onclick="document.images[1].src
   = 'hello_kitty.gif' ">

<img src="jet_li.jpg" onclick="document.images[2].src =
   'mcmug.gif' ">
</body>
```

http://course.cs.ust.hk/comp303/hkust/lectures/f2005_javascript_examples_for_lab3/09_image_arra - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://course.cs.ust.hk/comp303/hkust/lectures/f2005_javascript_examples_for_lab3/09_image_array.html

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

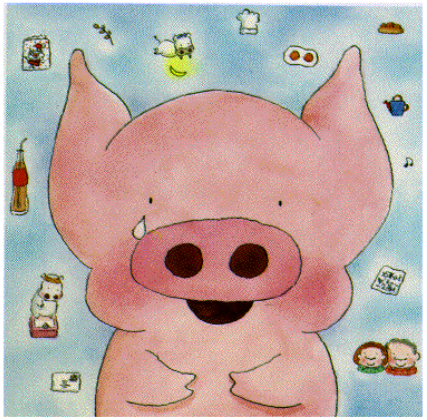


Click

f2005_javascript_examples_for_lab3/09_image_arra - Microsoft Internet Explorer

/f2005_javascript_examples_for_lab3/09_image_array.html

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



Multimedia in HTML5: <video>

- HTML5 supports <video> and <audio>
- Browsers supporting HTML5 (i.e., most browsers) have native support of video and audio objects without using external software

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Try it out

- controls vs autoplay
- First <source> tag with recognized format is played

Multimedia in HTML5: <audio>

- “controls” and multiple <source> tags are similar to <video> :

```
<audio controls>
    <source src="horse.mp3" type="audio/mpeg">
    <source src="horse.ogg" type="audio/ogg">
Your browser does not support the audio element.
</audio>
```

Try it out

Before HTML5: <embed>

- **embed** a video/audio object played by plugins:

```
<embed id="bgmusic" type="application/x-mplayer2"
src="12_fun_music.mid" hidden="true" autostart="true"
loop="true"></embed>
```

- Browser loads the MIDI file and use MediaPlayer to play it immediately (autostart="true")
- Media is played by a plugin, not natively by browser
- <embed> is not supported in HTML5; this slide is a salute to a technology that served us for many years

Take Home Message

- We look at more examples of HTML and JavaScript
  - Events and user interactions
  - Dynamic changes to web page elements using JavaScript
  - Handling of Non-text objects: <video>, <audio>
- Difference between playing a video natively by the browser and by a plugin