

Music & the Internet

MUMT 301

Ichiro Fujinaga
Schulich School of Music
McGill University

Plan

- Music Services
- Javascript review: Functions
- Javascript:
 - Functions with arguments
 - Functions with return value
- Javascript: Objects

Music Service websites

- For musicians
- For music companies
- For advertisers
- Metrics

Music Services: Musicians (I)

- Music licensing, marketing
 - <http://awal.com/> (see What We Do)
 - <http://headliner.fm/howitworks.php> (watch the video)
 - <http://www.nimbit.com/> (look at the pricing)
 - <http://www.hellomusic.com/landing.aspx> (sell products to musicians)

Music Services: Musicians (2)

- Facebook app form marketing
 - <http://www.rootmusic.com/> (BandPage)
- DIY music store:
 - <http://members.cdbaby.com/>
 - <http://vibedeck.com>
 - <http://official.fm/> (DYI marketing)
 - <http://routenote.com/> (simple distribution)
- Fan management
 - <http://www.fanbridge.com> (spam facilitator?)
- Copyright management
 - <http://www.mywerx.com>

Music Services: Music Companies

- Large-scale music streaming service
 - <http://www.3gm.hu/>
 - <http://fuga.me/>
- Collect royalties
 - <http://www.bmat.com/products/vericast>
 - <http://www.tunesat.com/> (monitors music on TV)
- Video production, marketing
 - <http://www.dancefoundation.com/>
 - <http://www.mndigital.com/>
- Events management
 - <http://www.eventbrite.com/>

Music Services: Advertisers

- <http://www.targetspot.com/> (find music-related and other website to advertise)
- <http://audiosocket.com/> (see About)

Music Services: Metrics

- <http://bcdash.bigchampagne.com/>
- <http://the.echonest.com/>
- <http://gigulate.com/>
- <http://www.nextbigsound.com/stats>
- <http://www.musicmetric.com/products/>
- <http://trueanthem.com>

!! New Music Business Model !!

- Charge license fees to manufactures of computers and mobile devices but free to consumers
- <http://beyondoblivion.com/>

This is the future!!

See: <http://boinc.com/launch/>

Break

Javascript: Review

- **Functions**

Javascript: Functions

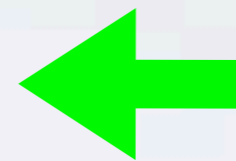
- Also known as procedures, subroutines, methods, etc.
- Purpose: To collect a set of statements to be used repeatedly anywhere in the script
- Four basic parts of a function
 - Key word: **function**
 - A function name
 - An optional comma-separated list of arguments to pass to the function enclosed in parenthesis
 - The statements in the function enclosed in curly braces
- Two sides to every function
 - The definition
 - Defines its behaviour
 - The call
 - Temporary pass control to the start of the function previously defined

JS: Function Definition & Call

```
<html> <head>
```

```
  <script type="text/javascript">
```

```
function print_triangle()  
{  
  var line = "";  
  for (counter = 0; counter < 10; counter++)  
  {  
    line = line + " #";    // append to line, "#"  
    document.write(line + "<br />");  
  }  
}
```



Definition

```
</script>
```

```
</head>
```

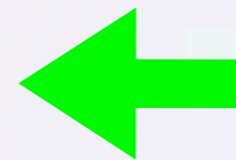
```
<body>
```

```
  <script type="text/javascript">
```

```
  print_triangle();
```

```
</script>
```

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



Call

JS: Function with arguments

```
<html> <head>
  <script type="text/javascript">
function print_triangle(n_lines)
{
  var line = "";
  for (counter = 0; counter < n_lines; counter++)
  {
    line = line + " #";      // append to line, "#"
    document.write(line + "<br />");
  }
}
  </script>
</head>
<body>
  <script type="text/javascript">
print_triangle(12);
  </script>
</body> </html>
```

JS: Function with return statement

```
<html> <head>  
  <script type="text/javascript">  
    function product(a, b)  
    {  
      return a* b;  
    }  
  </script>  
</head>  
<body>  
  <script type="text/javascript">  
    document_write(product(2, 3));  
  </script>  
</body> </html>
```

JS: Built-in functions

- alert()
- prompt()
- confirm()
- An example:

```
<html> <body>
```

```
  <script type="text/javascript">
```

```
    alert("This is an alert box");
```

```
    answer = prompt("Enter your name:");
```

```
    yes_no = confirm("Do you really want to delete this file?");
```

```
  </script>
```

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


JS: Objects

- An object is a special type of data
- It has **properties** and **methods** (functions)
- **Properties** are the details about an object
- **Methods** are things that can be done with the object
- You can create your own objects
- You can use Javascript built-in objects:
 - Strings
 - Math
 - Date
 - etc.
- You can use HTML objects
 - Document
 - Window
 - Navigator
 - etc.

JS: Built-in object examples

- **String object**

- **String property: length**

```
var text = "Hello World";  
document.write("The string length is: " + text.length);
```

- **String method: substr(start, length)**

```
document.write("Substring: " + text.substr(6, 5));
```

- **Math object**

- **Math property: PI**

```
document.write("PI accoring to JS: " + Math.PI);
```

- **Math method: sqrt**

```
var x = 16;  
document.write("Square root of " x " is " + Math.sqrt(x));
```

- **Date object**

- **Need to create it first**

```
var d = new Date();  
document.write(d);
```

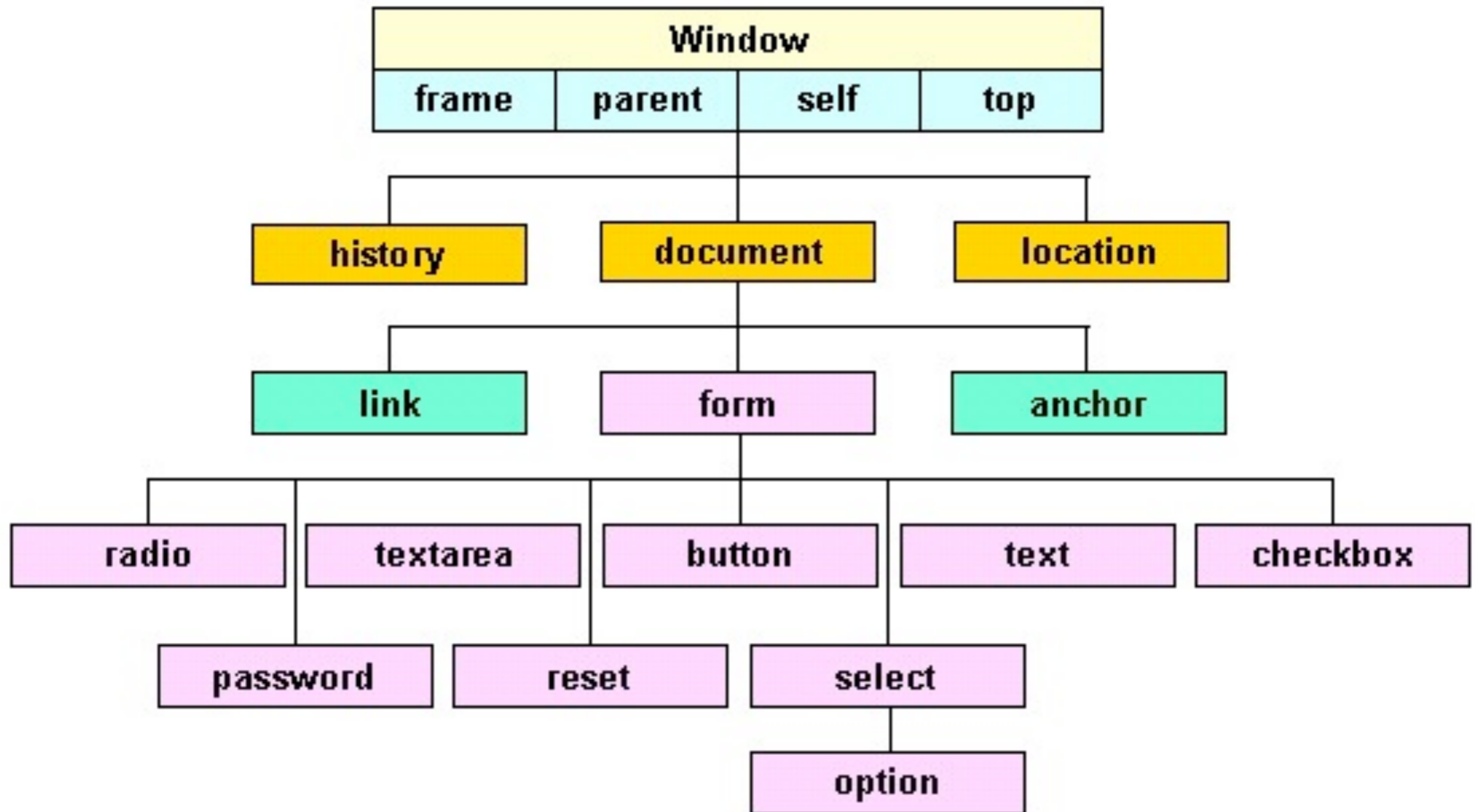
- **Date method: getDay**

```
document.write("Today is the day " + d.getDay() + " of the week.");
```

JS: DOM object examples

- DOM (Document Object Model)
- Examples:
 - **document** object is the current html page in the browser window
 - document property: URL
`document.write(document.URL);`
 - document method: `write()`
`document.write("Writes to this page");`
 - **window** object is the browser window
 - window properties: `screenX` & `screenY`
`document.write("X:" + window.screenX + "Y:" + window.screenY);`
 - **navigator** object contains the browser information
 - navigator properties: `appName` & `appVersion`
`document.write(navigator.appName + navigator.appVersion);`

JS: DOM objects



JS: DOM objects, more examples

- Button
- Event
- Image
- http://www.w3schools.com/js/js_ex_dom.asp