

BTI220 - Internet Architecture and Development

**Week 4: Introduction to HTML &
More on JS**

Agenda

- More on JavaScript
 - Date and Math Objects
- What is HTML?
- Document structure/overview
- HTML5 Structural Elements
- Important HTML elements and using them
- Hyperlinks & Images basics

JavaScript Built-in Objects

- JavaScript built-in object list:

- String
- Array
- **Date**
- **Math**
- Number
- Boolean
- RegExp
- JSON

- We'll cover **Date** and **Math** objects in this week.

Date Object

- Enables basic storage and retrieval of dates and times.
- Creates a Date object with current date and time:

```
var myDate = new Date();
```

- date string:

```
alert("The date is " + myDate);
```

Will show the date string:

The date is Mon Mar 10 2014 09:02:37 GMT-0400 (Eastern Standard Time)

The get... Methods of Date Object

➤ getMonth() method

- Returns number of **0** through **11**
 - ▶ Represent month of **January** through **December** correspondingly
- e.g.

```
var myMonth = (new Date()).getMonth();
alert(myMonth); // The myMonth is 6 which is, July
```

➤ getDate() method

- returns number of 1 31
- e.g.

```
var myDay = (new Date()).getDate();
alert(myDay ); //
```

The get... Methods of Date Object

➤ **getDay()** method

- returns number of **0** for **Sunday**, **1** for **Monday**, ...
- e.g.

```
var myDayOfWeek = (new Date()).getDay();
alert(myDayOfWeek );
```

➤ **getFullYear()** method

- returns a 4 digit year
- e.g.

```
var myYear = (new Date()). getFullYear();
alert(myYear ); // 2015
```

The get... Methods of Date Object

- `getHours()` method
 - returns a number of 0 to 23
- `getMinutes()` method
 - returns a number of 0 to 59
- `getSeconds()` method
 - returns a number of 0 to 59
- `getMilliseconds()` method
 - returns a number of 0 to 999
- e.g.

```
var myDate = new Date();
var myHour = myDate.getHours();
var myMinutes = myDate.getMinutes();
var mySeconds = myDate.getSeconds();
alert(myHour + ":" + myMinutes + ":" + mySeconds); // 10:9:35
```

Displaying Dates

➤ Methods

- `toString()`
- `toLocaleString()`
- `toUTCString()`
- `toDateString()`

➤ e.g.

```
var dt = new Date();
alert(dt);                      // Sun Nov 23 2014 23:07:28 GMT-0500 (Eastern Standard Time)
alert(dt.toString());           // Sun Nov 23 2014 23:07:28 GMT-0500 (Eastern Standard Time)
alert(dt.toLocaleString());     // 11/23/2014, 11:10:19 PM
alert(dt.toUTCString());        // Mon, 24 Nov 2014 04:10:19 GMT
alert(dt.toDateString());       // Sun Nov 23 2014
```

Math object - Math functions

- **Math.max(ident_1, ident_2)**
 - the maximum of n numbers
 - e.g. `alert(Math.max(0.52, 1)); // 1`
- **Math.min(ident_1,ident_2)**
 - the minimum of n numbers
 - e.g. `alert(Math.min(0.52, 1)); // 0.52`
- **Math.pow(ident_1, ident2)**
 - ident_1 to the power ident_2
 - e.g. `alert(Math. pow(2, 8)); // 256`
- **Math.sqrt(ident_1)**
 - square root of
 - e.g. `alert(Math. sqrt(9)); // 3`

Rounding floating-point

- **Math.ceil(ident_1)**
 - integer closest to and not less than
 - e.g. `alert(Math.ceil(0.52)); // 1`
 `alert(Math.ceil(0.49)); // 1`
- **Math.floor(ident_1)**
 - integer closest to and not greater than
 - e.g. `alert(Math.floor(0.52)); // 0`
- **Math.round(ident_1)**
 - integer closest to
 - e.g. `alert(Math.round(0.52)); // 1`
 `alert(Math.round(0.49)); // 0`
 `alert(Math.round(0.5)); // 1`

Generating Random Number

- **Math.random()** - pseudorandom number
 - Return a floating point number between 0 (inclusive) and 1 (exclusive)
 - e.g. `alert(Math.random()); // 0.03517110995016992`
- Generating number 1 to 10
`Math.floor((Math.random()*10)+1);`

What is HTML

- HTML (HyperText Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page.
 - **Hypertext** is text with hyperlinks.
 - The **markup** tells the Web browser how to display a Web page's words and images for the user.
- The markup symbols/indicators are often called "**tags**", which are enclosed in angle brackets
 - Most html tags come in pairs e.g. <p> and </p>
 - <p> being the opening of the tag and
 - </p> being the closing of the tag.
 - In between these tags you can add **text-based content**.
 - There are some tags that are not paired – these tags are known as **empty tags**, such as

Basic HTML Document Structure

```
<html>
  <head>
    <title>INT222</title>
  </head>
  <body>
    <h1>Basic HTML Document Structure</h1>
    <p>This is a paragraph</p>
    <p>Here are links to
      <a href="http://www.senecac.on.ca/">Seneca College</a>
      <a href="https://scs.senecac.on.ca/">ICT</a> school
    </p>
    <p><a href="view-source:">View-source</a></p>
  </body>
</html>
```

[Notepad++](#)

[Show in browser](#)

Tags vs Elements

- The terms tag, element & attribute are used throughout the web site. You should note the difference between these terms.
 - HTML elements:
 - An HTML Element is everything from the **start tag** to the **end tag**,
 - Html documents are defined by HTML elements
 - e.g.
 - `<p>Some text</p>` - is referred to as an **element**, including starting tag - content - ending tag
 - `<p>` and `</p>` - are referred to as **tags**.

HTML Element Categories

HTML elements/tags are classified in three different categories:

➤ **Block-level** element:

- A block-level element is a tag that creates large blocks of content like tables (`<table>`) or page divisions (`<div>`).
 - e.g. `<p>`, `<h1>`, ``, `<hr>`, `<dl>`, ...
- By default, block-level elements begin on new lines.
- They can contain other block tags as well as **inline** tags and text.

HTML Element Categories

➤ **Inline-level** element:

- An inline element is a tag that defines the text or data in the document. Using STRONG() makes the enclosed text strongly emphasized.
- e.g. , <a>, , <td>, , , <input>, ...
- Inline elements don't start new lines when they are used.
- they generally only contain other inline tags and text or data.

HTML Element Categories

➤ Empty element

- An empty element does not have closing tags or they are not paired.
- An empty element does not contain any text/content.
- Empty tags are simply used as markers.
 - ▶ In some cases empty tags are used for whatever is contained in their attributes.
- The `
`, ``, `<input />`, `<meta />` tags are a few examples of empty tags.

Attributes

- An attribute is used to define the characteristics of an element, and it is placed inside the opening tag.
- e.g.
`<p id="a3" name="a3" class="highlight">Some text</p>`
 - Id, name and class are examples of **attributes**
- All attribute are made up 2 parts: **name** and **value**.
- Note:
The new HTML standard (HTML5) does not require quotes around attribute values, but we suggest to do so.

HTML Core Attributes

- also called HTML **global attributes**
- can be used on all elements.
- e.g.

```
<h4 title="Hello HTML!">Titled Heading Tag Example</h4>
<div class="className1 className2 className3">
  <p id="html">This para explains what is HTML</p>
  <p style="font-family:arial; color:#FF0000;">Some text...</p>
</div>
```

 [coreAttributes.html](#)

About HTML5

- **HTML5 is the latest standard that defines HTML.**
 - HTML: created in 1990 and standardized as HTML 4 in 1997.
 - xHTML became a [W3C](#) Recommendation in 2000.
 - HTML5 is a candidate recommendation of W3C as of 2012.
- **HTML5 comes with a number of new elements, attributes, and behaviors.**
 - Providing new semantic, graphics, and multimedia elements.
 - designed to deliver rich web content without the need for additional plugins.
- **A larger set of technologies that allows more diverse and powerful Web sites and applications.**
 - New form elements and new API's to make it easier to build web applications.
 - Supporting cross-platform, designed to work on types of hardware (PCs, Tablets, Phones, TVs, etc.)

Basic HTML5 Document Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>INT222</title>
  <link href="css/mystyle.css" rel="stylesheet"/>
  <script src="js/myscript.js"></script>
</head>
<body>
  <h2>Basic HTML5 Document Structure</h2>
  <p>This is a paragraph</p>
  <p>Here are links to
    <a href="https://scs.senecac.on.ca/">ICT</a>
    <a href="http://www.senecac.on.ca/">Seneca College</a>
  </p>
</body>
</html>
```

Template for creating HTML5 file

Document Type Definition (DTD)

- Document type definition (DTD) is a set of markup declarations that define a document type for an Standard Generalized Markup Language (SGML), e.g. XML, HTML.
- Structure Examples - Doctype Declarations List
 - Basic HTML document structure
 - Basic xHTML 1.0 Transitional document structure
 - Basic xHTML 1.0 Strict document structure
 - Basic xHTML 1.0 Frameset document structure
 - Basic HTML5 document structure
 - HTML5 Document Structure

Discussion

- Why html then xhtml and now html5?
- The History of *HTML* - *HTML Source*

HTML5 Structural Elements

html tag	Description
<!DOCTYPE>	Specifies the document type
<html>	Specifies an html document
<head>	Specifies information about the document
<title>	Specifies the document title
<meta>	Specifies meta information
<link>	Specifies a resource reference
<script>	Specifies a script
<style>	Specifies a style definition
<body>	Specifies the body element
<!--...-->	Specifies a comment

Template for Creating HTML5 File

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>INT222</title>
  <link href="css/mystyle.css" rel="stylesheet"/>
  <script src="js/myscript.js">
  </script>
</head>
<body>
  <!-- code for the web page -->
</body>
</html>
```

Note:

- `<script></script>` tags are used for enclosing JavaScript file.
- `<link/>` tag is used to enclose CSS file.

HTML Heading Tags

Heading tags	Description	Example
<code><h1></code>	Specifies a heading level 1	<code><h1>.....</h1></code>
<code><h2></code>	Specifies a heading level 2	<code><h2>.....</h2></code>
<code><h3></code>	Specifies a heading level 3	<code><h3>.....</h3></code>
<code><h4></code>	Specifies a heading level 4	<code><h4>.....</h4></code>
<code><h5></code>	Specifies a heading level 5	<code><h5>.....</h5></code>
<code><h6></code>	Specifies a heading level 6	<code><h6>.....</h6></code>

Heading Tag Examples

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>INT222</title>
</head>
<body>
  <h1>Level 1</h1>
  <h2>Level 2</h2>
  <h3>Level 3</h3>
  <h4>Level 4</h4>
  <h5>Level 5</h5>
  <h6>Level 6</h6>
</body>
</html>
```

Headings

Presentation Tags

Tags	Description	Example
<p>	Specifies a paragraph.	<p>.....</p>
<blockquote>	Specifies a long quotation. It will indent the right and left margins both on the display and in print form.	<blockquote>.....</blockquote>
<pre>	Specifies preformatted text, e.g. keep white space.	<pre>.....</pre>

,
	Inserts a single line break.	
<hr />, <hr>	Specifies a horizontal rule.	<hr />
<mark>	Highlight parts of a text.	<mark>

tags-paragraph+.html

Whitespace & HTML Entities

- Whitespace characters
 - spaces, tabs, and newlines
 - HTML treats them as a single space.
- HTML Entities
 - Reserved characters in HTML must be replaced with character entities.
 - Some useful html character entities:

Entity	Description	Entity Name	Entity #
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
©	copyright	©	©

□ tags-preserveFormatting.html

Presentation Tags

Tags	Description	Example	Equivalent CSS
	Specifies bold text	{ font-weight: bold; }
	Specifies emphasized text	{ font-style: italic; }
<i>	Specifies italic text	<i>.....</i>	{ font-style: italic; }
<u>	Specifies text to be underlined	<u>.....</u>	{ text-decoration: underline; }
<sup>	Specifies superscripted text	^{...}	{ font-size:small; vertical-align:top; }
<sub>	Specifies subscripted text	_{...}	{ font-size: xx-small; vertical-align: bottom; }

tags-presentation.html use CSS instead

HTML Grouping Tags

- The `<div>` and `` elements have no special meaning, but they can group HTML elements into sections.
- You group sections of an HTML page when you want to perform an action on multiple elements.

Tag	Description
-----	-------------

`<div>` Defines a section in a document block-level element

`` Defines a section in a document inline element

- ❑ Note: It should be used only when no other semantic element (such as `<article>`, `<nav>`, `<section>`) is appropriate.

❑ [tags-grouping.html](#)

HTML List Tags

Three types of list tags in HTML:

- Unordered lists
- Ordered lists
- Definition lists

Unordered lists

- The **** tag displays an unordered bulleted list. You can use CSS (**list-style-type** property) to control the bullet style.
- The **** tag is used to designate the individual list items in the list.
- Both the **** and the **** require a closing tag (**** and ****).

Tags	Description	Example
	Specifies an unordered list	<code> </code>
	Specifies a list item	

[tags-list-unordered.html](#)

Ordered lists

- The `` tag displays an ordered list. You can use CSS (`list-style-type` property) to control the sequence style.
- The `` tag is used to designate the individual list items in the list.
- Both the `` and the `` require a closing tag (`` and ``).

Tags	Description	Example
<code></code>	Specifies an ordered list	<code></code> <code> </code> <code> </code> <code> </code> <code></code>
<code></code>	Specifies a list item	

[tags-list-orderd.html](#)

Definition lists

- The `<dl>` encloses a definition list.
- A definition list contains
 - **terms**, which are defined with the `<dt>` tag, and
 - **descriptions**, which are defined with the `<dd>` tag.
- Each `<dl>`, `<dt>` and `<dd>` tag requires a closing tag (`</dl>`, `<dt>` and `</dd>`).
- By default, a browser will align terms on the left and indents each definition on a new line.
- The intent of a definition list is to display lists of terms and their corresponding descriptions, such as in a glossary.

Definition lists

Tags	Description	Example
<dl>	Specifies a definition list	<dl>
<dt>	Specifies a definition term	<dt> </dt>
<dd>	Specifies a definition description	<dd> </dd> <dd> </dd> <dt> </dt> <dd> </dd> </dl>

[tags-list-definition.html](#)

Nested lists

- Ordered lists and Unordered lists can be nested - a combination of the two can also be nested.
- Each level will be indented.
- Nested lists may look complicated however you just need to remember the basic structure for ordered and unordered lists.

[tags-list-nested.html](#)

Example

```
<ol>
  <li> .... </li>
  <li> ....
    <ul>
      <li> .... </li>
      <li> .... </li>
    </ul>
  </li>
  <li> .... </li>
  <li> .... </li>
</ol>
```

Hyperlinks & Anchor

- The HTML `<a>` Element (or the HTML **Anchor Element**) defines a **hyperlink**, the named target destination for a hyperlink, or both.
- A **hyperlink** (or **link**) is a word, group of words, or image that you can click on to jump to another document or another part of the same document.
- Basic HTML link (anchor) format:

```
<a href="URL.....">text</a>
```

Hyperlinks

➤ Absolute link

```
<a href="https://scs.senecac.on.ca/~wei.song">  
Wei Song's Website</a>
```

➤ Relative link

- The links should be relative to the location of the current document. e.g.

```
<a href="xxxxx.html">Text...text</a>  
<a href="..//xxxxx.html">Text...text</a>  
<a href="..//info//xxxxx.html">Text...text</a>
```



Hyperlinks and Anchor

- Link to a particular section of an html page
 - To link to a specific section (Timetable) of a page named int222.html), you need to create a **bookmark (named anchor)** inside the page/document.
 - ▶ e.g.
``
 - Note: `` will also works, but not suggested
 - Then use hyperlinks to link to section/bookmark:
 - The hyperlink is in the same page
`Go to Timetable`
 - The hyperlink is in other pages of the same site
`My Timetable`
 - The hyperlink is from other site (External link)
` My Timetable`

More Hyperlinks

- E-mail link

```
<a href="mailto:wsong@myseneca.on.ca">Email me</a>
```

- Image link

```
<a href="xxxxx.html">  
    
</a>
```

<a> Tag (Anchor) Attributes

- **target** - Specifies where to open the linked document.

```
<a href="xxxxx.html" target="window_name">  
  Text...text</a>
```
- **download** – Specifies that the target will be downloaded when a user clicks on the hyperlink.
- **Name** - Not supported in HTML5. Use the global id attribute instead.

[tags-hyperlinks.html](#)

Resourceful Links

- **Introduction to HTML (MDN)**
<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction>
- **HTML5**
<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
- **HTML element reference (MDN)**
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
- **HTML attribute reference**
<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>
- **Basic Structure of an HTML5 Document**
<http://www.coreservlets.com/html5-tutorial/basic-html5-document.html#>

Thank You!