Basics of HTML5 and Javascript

Standard HTML document looks like this:

<html>

<head>

<title>Very simple example</title>

</head>

<body>

This will appear as is.

</body>

</html>

The head of the HTML document will consist of the title, which is the text that is displayed in the top left hand corner of the window of the web browser. It will also be displayed as the title of the tab if the browser supports tabs. The text in the body will appear drawn as text from the top left hand corner of the screen, using the standard font of the browser.

To specify which images the browser will load, an image tag is required. This tag is slightly different to the HTML, head and body tags, as it is self-contained within one line. As such, it is called a singleton tag. An example can be seen below:

<img src = “frog.jpg”/>

By default the browser will look in the directory that contains the HTML file for the image files. Width and height variables can also be included to produce a tag like so:

<img src=”frog.jpg” width=”200” height=”300”/>

Hyperlinks are produced in a similar way to images:

<a href=<http://faculty.purchase.edu/jeanine.meyer>>Jeanine Meyer’s Academic Activities </a>

The href is the link that the page will go to when clicked on, while the text after that is what will be displayed in place of the link’s actual name (the blue and underlined text). The href and img tags can be combined to produce an image that is a hyperlink. An example can be seen below:

<a href=<http://faculty.purchase.edu/jeanine.meyer>>  
<br/>  
<img src=”jhome.gif” width=”100”>   
<br/>  
</a>

This produces an image that links to another webpage, tying together an image and a hyperlink. The <br/> produces a line break.

Cascading Style Sheets (CSS)

CSS is a special language just for formatting. One tag, <style>, is used to create rules that specify how particular elements are formatted. It can be placed in a variety of places, and the style that is closest to an element will be the one that is used. The text-align directives are used to specify where the material will align. Other directives include colour, font size, width (as a percentage), borders, padding, margins and background colour. The style tag specifies the general format of a page and can be used to create a large array of pages of a similar look. Because of how the style tag works, the closest style tag to an element will be used, so specific pages can have subtle changes to create slight differences between pages with greater ease.

An example can be seen below:

<html>

<head>

<title>CSS example </title>

<style>

body {

background-color:tan;  
color: #EE015;  
text-align:center;  
font-size:22px;  
}

section {

width:85%;

border:4px #00FF63 solid;

text-align:left;

padding:5px;

margin:10px;

background-color: white;

}

p {

width: 250px;

}

</style>

</head>

<body>

The background here is tan and the text is the totally arbitrary RED GREEN BLUE􀀁

value #EE015. <br/>

<section>Within the section, the background color is white. There is text with􀀁

additional HTML markup, followed by a paragraph with text. Then, outside the􀀁

section there will be text, followed by an image, more text and then a􀀁

hyperlink. <p>The border color of the section matches the color of the􀀁

frog image. </p></section>

<br/>

As you may have noticed, I like origami. The next image represents a frog head.<br/>

<img src="frogface.gif"/> <br/>If you want to learn how to fold it, go to

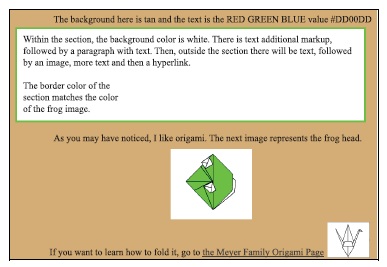
<a href=http://faculty.purchase.edu/jeanine.meyer/origami>the Meyer Family􀀁

Origami Page <img src="crane.png" width="100"/></a>

</body>

</html>

This code above produces the page below:



Javascript is an integral part of accessing features of an HTML document. The HTML document holds Javascript in a <script> element located within the <head> element. A useful method is the document.write() method, which simply writes anything that is held between the circle brackets (usually strings are held between quotation marks). An example can be seen below, which prints out the current date:

Document.write(Date());

Some useful tags are defined below:

|  |  |
| --- | --- |
| Tag Name | Explanation |
| <html> Text </html> | Opening and closing HTML tag |
| <head> Text </head> | Opening and closing head tag |
| <title> Text </title> | Opening and closing title tag |
| <body> Text </body> | Opening and closing body tag |
| <h1> Text </h1> | The h1 tag will make the text appear in a big font |
| <p> Text </p> | The p tag will create a new paragraph |
| <style> Text </style> | The style tag specifies certain parameters to be set |
| Article { } | The binding braces of a style, inside which specific characteristics of the ‘Article’ section is set |

The element that is most important in terms of game development is the canvas. It allows developers to make line drawings and draw images, as well as position them in a free form fashion, quite unlike what previous web developers had to work with.

In order to work with the canvas, it must be included in the body element of the HTML document.

<canvas id = “gameCanvas” width = “400” height = “300”> </canvas>

Another step to getting a hold of the canvas is to declare it as a variable in the Javascript file. The canvas is usually declared and used as a global variable, so is declared outside of any function. Now to assign the variable to the canvas defined in the HTML file:

Var gameContext = document.getElementById(‘gameCanvas’).getContext(‘2d’);

When creating the HTML document, there are some options in terms of where the files are located. For example, it would be quite easy to write all the code to a game inside the HTML file, in between the <script> tags, however, it would get fairly messy. This option would be best for simpler tasks, like setting up the initial rectangle in which the game would be drawn. The other option would be to defer from the HTML file to another, and just leave a reference to the file that the rest of the code is located in. An example of the first option can be seen below:

<html>

<head>

<title>Smile</title>

<script>

function init() {

var ctx =document.getElementById("canvas").getContext('2d');

ctx.beginPath();

ctx.strokeStyle = "rgb(200,0,0)";

ctx.arc(200, 200,50,0,Math.PI, false);

ctx.stroke();

}

</script>

</head>

<body>

<body onLoad="init();">

<canvas id="canvas" width="400" height="300">

Your browser doesn't support the HTML5 element canvas.

</canvas>

</body>

</html>

This code above draws a semi circle starting from (200,200).

**Fundamentals of HTML**

**HTML Headings**

The first major tag in a HTML document is the HTML heading. These range from <h1> to <h6>. The lower the number, the larger the heading size.

**HTML Paragraphs**

Another major tag is the paragraph tag, denoted by <p> </p>

**HTML Links**

These are defined by the <a></a> tag. This flags the text in between the tags as text that can be clicked on (highlighted in blue). To actually link a website to it, however, requires an attribute called href, which provides the address that the text will go to when the user clicks on it.

**HTML Images**

These are defined by the <img> tag. The coder can also specify width and height, as well as where to find the image.

**HTML Elements**

A HTML element is everything between the start (opening tag) and the end (closing tag) tag. An example can be seen below:

1. <p> This is a paragraph </p>  
2. <a href=”google.com”>This is a link.</a>

In the first example, the element content is the text “This is a paragraph”. Similarly, in the second example, the element content is the text “This is a link”.

An important piece of the HTML element is the attribute, which will be explained later on. Most HTML elements can have attributes.

Another important quality elements have is that they can be nested, creating images that link to other pages when clicked on, or paragraphs of varying font sizes. An example can be seen below:

<html>  
<body>  
  
<p>This is the first paragraph</p>  
  
</body>  
</html>

The paragraph element is nested in the body element, which is nested inside the HTML element.

The <body> element defines the body of the HTML document, while the HTML element defines the entire HTML document. There are some HTML elements that are empty, such as the line break <br> element. These can be closed within the start tag.

One major difference between HTML and other programming languages is that HTML is not case sensitive, so an upper case label for a tag is the same as a lower case label for a tag.

**HTML Attributes**

Attributes provide additional information about HTML elements and are always specified in the start tag. They are typically defined in a name-value pair and are always enclosed within quotation marks. Both single and double quotes are allowed. When the attribute value itself includes quotation marks, single style quotes are necessary:

Name=’John “Shotgun” Nelson’

As seen previously, the link tag <a> contains an attribute called href, which directs the webpage to another webpage when clicked on.

**HTML Headings, Rules and Comments**

Headings are useful to get general information to readers of a webpage in a relatively quick manner. Search engines also use the headings of a webpage to index the structure and content of webpages, so it is important to use them wisely and properly.

HTML rules are in reference to lines. The <hr> tag is used to rule a horizontal line across the browser page, providing a visual break from different pieces of information.

HTML comments can be inserted through this syntax:

<!--This comment will not be displayed-->

Unfortunately for HTML there are a large number of devices that can display it. As a result, it is never truly clear how the code that has been written up will be displayed. Careful thought must be placed into writing HTML documents as the browser ignores the layout defined by the writer.

**HTML Text Formatting**

|  |
| --- |
| Tag |
| <b> bold text |
| <strong> strong text |
| <big> big text |
| <em> emphasised text |
| <i> italicised text |
| <small> small text |
| <sub> subscript <sup> superscript |
| <pre> allows preformatted text |



There are also tags that can be used for displaying computer output:

|  |
| --- |
| Tag |
| <code> displays computer code |
| <kbd> displays keyboard input |
| <tt> displays teletype text |
| <samp> displays sample text |
| <var> displays computer variable text |
| <address> used to display addresses |

**Abbreviations and Acronyms**

|  |  |
| --- | --- |
| Tag | Displayed output |
| <abbr title = “United Nations” UN </abbr> | UN |
| <acronym title = “World Wide Web”>WWW</acronym> | WWW |

**Text Direction**

When writing HTML code, the direction that the text is written in can also be influenced and changed. This is subject to support by the browser however. If the browser supports bidirectional override, then the <bdo dir=”rtl”> tag can be used to change the text direction from left to right to right to left.

**Quotations**

Handling of long and short quotations can also be done using the <blockquote> or <q> tag respectively. The <blockquote> tag makes the browser insert line breaks and margins so that the text quote is centred and stands out, while the <q> tag simply places the text inside quotation marks.

**Deleted and Inserted Text**

When users delete or insert text, especially as a result of an edit, browsers can display this text as either stroked out or underscored respectively. The tags are <del> and <ins> respectively:



**HTML Styles**

The style attribute introduces the use of cascading style sheets (CSS) into HTML. It provides a common way to style all HTML elements. Common HTML styles include changing the background colour (tag: background-color), font family, colour and size (tag: font-family, color, font-size), text alignment (tag: text-align). An example can be seen below:

<html>  
<body style = background-color:red; font-family:times new roman; color:black; size:32px>  
</body>  
</html>

**HTML Links (aka anchor element)**

A link is the address to a document located in a place other than your local desktop. The target attribute provides control over how the browser responds when a link is clicked. For example, setting the target attribute to blank will trigger a new window when the link is clicked on:

<a href=”google.com” target=”\_blank”></a>

Other attributes apart of the link element include \_self (opens linked document in same frame), \_parent (opens linked document in the parent frame) and \_top (opens linked document in full body of the window).

In most cases a link tag involves a hyperlink attribute, which is a reference to a resource on the web. The hyperlink attribute is defined by href, which in turn defines the link address.

Another attribute of the anchor element is the name attribute. A named attribute defines a named anchor inside a HTML document. There are not displayed by the browser, as they are meant to be invisible to the reader. One use they have found is when naming chapters within a large document. Each chapter is given a named anchor and links to each of these anchors are put at the top of the document.

An example of the named anchor syntax:

<a name=”tips”>Useful tips for a new writer of HTML</a>

An example of the link syntax to a named anchor:

<a href=”#tips”>Jump to useful tips section</a>

The # in the href attribute defines a link to a named anchor. An example of linking to the useful tips section from another HTML document:

<a href=http://www.w3schools.com/html\_tutorial.htm#tips>Jump to the useful tips section</a>

Another example of linking to other parts of the same document:

<html>

<body>

<p>

<a href="#C4">See also Chapter 4.</a>

</p>

...

<h2><a name="C4">Chapter 4</a></h2>

</body>

</html>

**HTML Images**

Images are defined with the <img> tag. It is an empty tag, meaning that is contains attributes only and has no closing tag. To display an image, the source (src) attribute must be defined. The source is typically a URL of the image, however, for testing purposes, it can also be an image located in the same directory as the HTML document. The image will be displayed where it occurs in the code, so x and y coordinates of images are not used to position externally sourced images in HTML.

Adding a background image to a HTML page is done using the background attribute in the body element. Both PNG and JPG files can be used as HTML backgrounds. If the image is smaller than the page, the image will repeat itself.

Images can also be aligned in regards to lines of text. Images can be aligned to the top, middle or bottom in regards to a line of text. An example of displaying images can be seen below:

<p>This image appears

<img src="hackanm.gif" width="48" height="48" />

exactly where it is placed in the code.</p>

This will place an image in the middle of the sentence. This technique can also be used to create floating images:

<p>

<img src="hackanm.gif" align="left" width="48" height="48"/>

A paragraph with an image. The align attribute of the image

is set to "left". The image will float to the left of this

text.

</p>

This will make the browser display an image on the left hand side of the text

**HTML Lists**

Lists can be ordered or unordered and can contain numbers, words, paragraphs, images, links, other lists and so on. Ordering can also be defined by the writer of the document.

<html>

<body>

<h4>An Unordered List:</h4>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>



Ordered lists:

<h4>Lowercase letters list:</h4>

<ol type="a">

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

</ol>

This creates an ordered list using the alphabet to label each element in the list. Lettered numerals and normal numbers can also be used to label the elements in the list.

Definition lists are a list of items together with a description of each item. A definition list starts with a <dl> tag, each term starts with a <dt> tag and each description starts with a <dd> tag.

<html>

<body>

<h4>A Definition List:</h4>

<dl>

<dt>Coffee</dt>

<dd>Black hot drink</dd>

<dt>Milk</dt>

<dd>White cold drink</dd>

</dl>

</body>

</html>



**HTML Forms and Input**

Forms are used to collect different kinds of user input. A form is an area that can contain form elements, allowing the user to enter information to be collected by the website. The main tag is the input tag, which has attributes such as type, name and width and height.

The type attribute includes things such as text, number, color, date, button and check box. These determine what kind of form is displayed by the browser as opposed to what kind of information is taken in. The name attribute gives the form something that can be referred to in the code later on.

An example of check boxes can be seen below:

