

CNIT 131H – 4-day course

Day 1

- Introductions
- What tools to use: Adobe Brackets or Sublime, World Wide Web
- Basic HTML – some parts might be a review to some students

- Open web editor, blank file
- Write `<!DOCTYPE html>`
- Why we need to include this tag? What is quirks mode?
- Write the following:

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

- Based on the code above: how many parts make an HTML document? What do you put in the head and what do you put in the body?
- Between the `<head>` and `</head>` tags, write the following:

```
<meta charset="utf-8">
<meta name="description" content="HTML is fun">
<title>My web page</title>
```

- What is the use of meta charset? What is the use of meta description? Other meta tags to mention (keywords, author). What is the importance of title?
- Save your file as index.html
- Between the `<body>` and `</body>` tags, write the following:

```
<div>
  <h1>My web page</h1>
</div>
<div>
  <h3>Who I am?</h3>
  <p>My name is .....</p>
</div>
<div>
  <p>This belongs to .....<br>
  Copyright 2016</p>
</div>
```

- Save the file again and now let's open Chrome browser and drag the index.html file you just saved to be inside the browser. Let's answer these questions: How many elements you have created in this code? Are those block or inline elements? What is the importance of having headings in the content? What is the biggest heading? What are important points to remember when writing content to the web to a certain web page? What is the `
` tag doing in the last paragraph? What happens if you have more content than the width of the browser window?
- Let's now change the code we have just typed and we will substitute the div tags so the code will become:

```

<header>
  <h1>My web page</h1>
</header>
<section>
  <h3>Who I am?</h3>
  <p>My name is .....</p>
</section>
<footer>
  <p>This belongs to .....<br>
  Copyright 2016</p>
</footer>

```

- Save the file and then open again (or refresh) in the browser. Any difference? What are those tags we inserted (footer, header, section)? Other HTML5 tags – semantics.
- Let's modify the code to include the nav tag. You will include the following code one line below the </header> tag (give some spaces between the words Home, Education, Experience):

```

<nav>
  Home   Education   Experience
</nav>

```

- Save the file and then let's now include the aside tag. You will include the following code one line below the </section> tag:

```

<aside>
  <h4>Favorite Websites</h4>
  <p>Site 1</p>
  <p>Site 2</p>
</aside>

```

- Save the file and reopen in the browser. Observations and questions: Were the spaces you put manually between the words Home, Education, Experience, followed by the browser? Could we then type the whole code in one single line without the indentation we are doing? Comments about the nav and aside tags.
- We need to make use of special characters if we want to include certain characters that do not belong to our usual alphabet, or that are being "ignored" by the browsers – those are called special characters.
- We can include special characters using their "special" names or their numeric code but we ALWAYS start a special character with & (ampersand) sign – see the examples below:
 - © or © - this represents the copyright symbol ©
 - ® or ® - this represents the registered symbol ®
 - ← or ← - this represents the leftwards arrow ←
 - < or < - this represents the less than symbol <
- A good reference for special characters can be found at http://www.w3schools.com/html/html_symbols.asp and at http://www.w3schools.com/html/html_entities.asp
- Let's include the non-blank space and the copyright symbols in our code. Between the words Home and Education, and then between the words Education and Experience, you will write 5 non-blank space symbols as or you can type
- Save the file and reopen in the browser – now your words are more separated, right?

- Let's include the copyright symbol. Right after the 2016 in the footer, type the special character code for the copyright symbol – do you remember how it's done?
- Save the file and open in the browser. How is the copyright symbol showing?
- We will work with images now. There are two ways that you can insert an image. Let's first insert an image using the img tag. First, make sure you have an image of yourself or that represents you. If you do not have an image, go to Google and search for a small image of a face or some image to work with. It needs to be jpg, or jpe, or png, or gif – those are the most used images on the web. Save the image in the same folder where your index.html is located and make sure you use all lowercase letters when choosing the name of the image you are saving.
- What is the difference between jpg, png, gif? Is jpe the same of jpg? What is the importance of having lowercase?
- So, right after the <p> and before the text "My name is", you will insert the following code:

```

```

Where **xxx** is the name of the image file you saved

- Save the file and show again in the browser. Based on what we know up to now, is the img tag block or inline? Is it a one-sided tag?
- The image is too big? You can use the width and height attributes such as `` - an important detail here is that those numbers do not have unit values and your code becomes invalid if you put the unit value – the browser will consider it to be pixels by default. So, include the width and the height attributes in the img tag of your HTML file and save it. Show in the browser to see the difference.
- Let's now change the img tag but just adding the <figure> and </figure> tags around the like:

```
<figure>
  
</figure>
```

- Save the file and show on the browser. What is the difference then between having the img tag and the img tag inside the figure tag? What else can you include between the <figure> and </figure> tags? Let's then include the figcaption tag and your figure will now be:

```
<figure>
  
  <figcaption>My cartoonish face</figcaption>
</figure>
```

- Save the file and show in the browser. Everything seems to be ok, right?
- We are growing with our code, so it's time to work with the validation of the code. Why it's important to write a VALID code?
- Open the <http://validator.w3.org> and choose the tab that says "Validate by Direct Input" – see the image below showing the front page of the W3C validator website.

Validate by URI Validate by File Upload **Validate by Direct Input**

Validate by URI
 Validate a document online:
 Address:
 ▶ More Options

- Before we proceed: Who is W3C?
- Select all the code you have up to now in your web editor and paste it on the space provided in the W3C validator web page – see my result below (it's not showing all the code to save some paper here....)

Validate by URI Validate by File Upload **Validate by Direct Input**

Validate by direct input
 Enter the Markup to validate:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="description" content="HTML is fun">
<title>My web page</title>
</head>
<body>
<header>
  <h1>My web page</h1>
</header>
<nav>
  
```

- After you do that, click on the button that says Check and you should get a result page that, if you scroll down, you would see something similar as:

```

    Copyright 2010 &copy; </p>
  </footer>
</body>
</html>

```

Use the Message Filtering button below to hide/show particular messages,

1. **Error** No `p` element in scope but a `p` end tag seen.
[From line 21, column 280: to line 21, column 283](#)
 eb editor.</p>=></sec>

Document checking completed.

Source

```

1. <!DOCTYPE html>=>

```

- It shows that we have an error – it shows the line and column. It seems that it's complaining about that </p> closing paragraph tag?!?! But I need to close the paragraph, right? Yes! But the problem started

Experience – of our website). Now, you should have four files in your folder: the image file, the index.html, the edu.html and the exp.html.

- Open in the web editor the edu.html and let's modify the meta description and the title accordingly and save the file.
- Do the same thing with the exp.html. Remember that we are doing this thinking about SEO (Search Engine Optimization) – the “tip of the iceberg” of SEO!
- Open again the index.html in your web editor and let's modify the code between the <nav> and </nav> tags to be the following:

```
Home&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a  
href="edu.html">Education</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a  
href="exp.html">Experience</a>
```

- Note the <a> tag that was included and its closing . Observe also that the anchor tag (the hyperlink tag) only requires the href attribute and the value of the href attribute will be the page you want to open for the user when the user clicks on that link...which link? The Education word will become a link and the Experience word will become a link too. Save the file now and reopen in the browser and you should see something as (showing just the top of the page):

My web page

Home [Education](#) [Experience](#)

Who I am?



- Only Education and Experience are hyperlinks because we coded the anchor tags only around those words. Comments about the importance of the text that you use to create a hyperlink.
- Let's now create the hyperlink code for the two websites you listed between the <aside> and </aside> tags. **ATTENTION: The difference here is that we will be directing the user to an external website** (a website, or web page, that does not belong to the current website context). In this case, **you NEED to include the COMPLETE URL to the website** – it will not work if you just write, for example, google.com or yahoo.com, or hired.com, etc. you need to write, for the href attribute value, the **COMPLETE URL** which for google, for example, would be http://www.google.com . If you would open a certain web page inside the Google website, you would then write something as http://www.google.com/**pathhere** where **pathhere** will be the path to or the name of the file (web page, image, audio, video, etc... any file).
- With that information, let's then create the correct link to the sites we mentioned between <aside> and </aside>
- Save the file and test your links on the browser after refreshing the page.
- Let's create now a folder called pages and move the edu.html and exp.html to those folders.
- Refresh the web page again and now try to click on the Education and also on the Experience link – any luck? Probably not because you modified the path of your file in the system!!! This means you need to modify the code accordingly. So, if you open index.html and modify the href value to be href="pages/edu.html" for the Education hyperlink and modify the href value to be

href="pages/exp.html" for the Experience hyperlink everything will then work fine (**REMEMBER TO ALWAYS SAVE THE FILE BEFORE REOPENING IT IN THE BROWSER!**)

- How should I then save my files? Comments about organization of files in the real world.
- What if I want to open the website in another window/tab of the browser? You can use one more attribute called target and the code would be something as `Google Website` - when the user clicks on the hyperlink text "Google Website", it will take the user to the Google front page but in another window/tab of the browser – note the target="_blank" attribute – this is what "makes the magic". Let's then include that target="_blank" attribute in the hyperlinks we coded between the `<aside>` and `</aside>` tags.
- Save the file and refresh on the browser and use the links to observe the opening of the websites in another window/tab of your browser. When should this attribute be used?
- Lists – used to organize items that should show up as lists. In HTML we have ordered, unordered (bullet) list, and description lists.
- We will start with the unordered list – for example, the sites we listed and linked between `<aside>` and `</aside>` tags could be in an unordered (bullet) list. Let's open index.html in the web editor and then use the following code to substitute the two paragraphs for the two hyperlinks you created in the aside area:

```
<ul>
  <li><a href="xyz" target="_blank">Site 1</a></li>
  <li><a href="xyz" target="_blank">Site 2</a></li>
</ul>
```

Where **xyz** is the COMPLETE URL to the website you want to take the user to and, of course, **Site 1** and **Site 2** should have a clear name of that website

- Save the file and reopen it in the browser. Note that the `<p>` and `</p>` tags have disappeared from the code and again because the ul and li tags are block tags so they cannot be inside a paragraph which is also a block element. Notice also the bullets that are shown in front of each item you coded using the `...` tags. The bullet is the default way to show items of an unordered list (that's why it's also called bullet list, got it?)
- The `` tag starts the unordered list while the `` closes it and between those two tags, you will include as many `...` pairs as the number of items you want to create.
- If you want to change an unordered list to be an ordered list, all you need to do is change the `` and `` tags to be `` and `` tags respectively. Just do it in your index.html, save the file and reopen in the browser. Now you should see numbers instead of bullets in front of each item.
- Let's stop now and think about this: when should I use unordered or ordered list? Does it matter?!?
- Based on our discussion, let's then change back the `` and `` tags to be `` and `` again and save our index.html file.
- What about the description list?
- Right after the ``, still in the aside area of our page, let's write the following code:

```
<d1>
  <dt>HTML</dt>
  <dd>Hyper Text Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheet</dd>
</d1>
```

- Save the file and open it in the browser again (or refresh it). Notice how the description list was coded and how the browser shows it! What would then be a great use for this type of list? Remember that it starts with the <dl> tag and ends with the </dl> tag and between those two tags, you will code as many <dt></dt> and <dd></dd> pairs as the items you have to “describe”. This list is also known as definition list.
- Tables – should ALWAYS be used to present TABULAR data. Can you give examples of what tabular data would be?
- Tables have been used by some web developers to layout the whole entire page but this has been considered completely obsolete when CSS came up. Do you know the reasons it’s not a good idea to layout a web page based on tables?
- Let’s include a table in our index.html and you will see that it’s not hard to code a table in HTML
- Open index.html and inside the section area (between <section> and </section>), right after the closing paragraph tag (</p>) where you have introduced yourself, you will start your table. So, create another line and type the following code:

```
<p>You will find the following pages in this website to know more about myself:</p>
<table>
  <tr>
    <th>Title</th>
    <th>Description</th>
    <th>Link</th>
  </tr>
  <tr>
    <td>Education</td>
    <td>About my educational background</td>
    <td>Go to <a href="edu.html">Education page</a></td>
  </tr>
  <tr>
    <td>Experience</td>
    <td>About my professional experience</td>
    <td>Go to <a href="exp.html">Experience page</a></td>
  </tr>
</table>
```

- **NOTE:** if you want to copy and paste the code, you need to make sure that you change the double quotes otherwise they will be **CURVED** (rich text format) and not **STRAIGHT** (plain text format) which means that the browser will NOT recognize the quotes as part of the code and the browser will not parse the code correctly. So, if you copy the code from this document and paste in your index.html file, remember to delete and retype the double quotes!!! **HTML, CSS, JavaScript need all to be in PLAIN TEXT FORMAT!!!**
- Save your file and reopen in the browser. Observe how the table shows. What is the difference between the <th> and <td> tags? What is the <tr> tag used for? Is there any separation between the cells of the table?
- Final observations:
 - Each tag in HTML creates an element that can be inline or block – this is called the display status of the element that can be modified via CSS
 - By default, the width of a block element will be 100% of the width of the browser window and the width of the element can also be modified via CSS

- The browsers have a default method to present each tag/element that you code because the browser works with what is called “user-agent style” (kind of the internal CSS of the browser) and, of course, you can override these default characteristics when you use CSS
- Introduction to CSS – what is the meaning of CSS? Why the word Cascading? What is the meaning of parent or child elements in HTML and CSS?
- How can CSS be used?
 - Inline CSS – using the style attribute for the tag – this is the one we will learn in this course
 - Embedded or document-level CSS – coded between the <style> and </style> tags that are positioned in the head section of your HTML document (in the head section, not the header tag – remember when we mentioned that an HTML document has two different sections: the head and the body)
 - External CSS – coded inside a plain text format file and this file will be named with the extension .css and the way you link that file to your HTML document is using the <link> tag that is coded in the head section of the HTML document
- Which one is most used? When would I be using inline CSS?
- CSS is made of selectors, properties and values for the properties. In the case of inline CSS we will only be using the properties and the values for the properties because the selector will be the tag/element that we are modifying using the inline CSS.
- How many CSS properties are there? Should I memorize those properties and values?
- References to CSS:
 - <http://www.w3schools.com/cssref/>
 - <http://htmldog.com/references/css/properties/>
 - <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
 - <http://meiert.com/en/indices/css-properties/>
 - etc.
- In inline CSS, we use the style attribute in the tag and the code should follow this format:

`style="cssproperty:cssvalue(s); cssproperty:cssvalue(s);..."`

- The : (colon) separates the CSS property from the CSS value(s) and after the value(s) you include a ; (semi colon). Some developers, when coding only one CSS property with its CSS value(s) do not include the ; (semi colon) at the end – it’s not wrong, but I recommend that you ALWAYS include that ; (semi colon) in case later on you decide to add more CSS properties.
- We will be using the inline CSS which is coded by using the style attribute inside the opening tag of an element. You should NEVER use the style attribute or any other attribute in the closing tag of any element – Example:

`<body style="background-color:cyan; color:blue;"> - this is ok`

`</body style="background-color:cyan; color:blue;"> - this is NOT ok because you are writing the style attribute in the closing tag and THIS DOES NOT EXIST!`

- What is the code above doing? What element(s) will be affected with this code?
- A good website to refer to, before using new HTML elements or CSS properties is the <http://www.caniuse.com> – you can type the name of an element or the name of a property and then you will have the graphic showing the browsers support – for example, the image below shows the caniuse.com screen after I typed background-color:

Can I use `background-color` ? [Settings](#)

2 results found

CSS `currentColor` value REC Global 94.68%

A CSS value that will apply the existing `color` value to other properties like `background-color`, etc.

Current aligned Usage relative Show all

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			29						
			45						
			48					4.3	
		45	49			8.4		4.4	
8		46	50			9.2		4.4.4	
11	13	47	51	9.1	38	9.3	8	50	50
	14	48	52	10	39				
		49	53	TP	40				
		50	54						