# Lab 02

## HTML & CSS Exercises, and Assignment 1: Career

## Overview

**Lab 02 exercises:**

- Anchor element styles;
- External and Internal CSS;
- Two Column Layout;
- Background and Floating Images;
- Adjusting associated Position element styles
- Continue Assignment 1 with discussion of your IT Career focus for your Personal Website

Today's lab covers more **Cascading Style Sheet** exercises. Try and understand what the code is doing, using the explanations given at the beginning of the lab. Most of the code is self-explanatory, once you are used to the syntax!

Before asking your tutor for the solution to a problem you may have with the exercises, try and find a solution on the Internet. This will give you a good idea what help and tutorials are "out there" – when you have problems with your website, you will know where to go to find answers. Don't forget about http://www.w3schools.com

Changing the **look** or **colour scheme** of a website can be as easy as **altering one file**. Imagine hundreds of pages in a website (not unusual), now imagine having to change each page because the font colour was chosen incorrectly (I know! Should never happen!!). Defining everything that is related to "style" in a **stylesheet** will make alterations a breeze.

The tables below contain some commonly used properties in Cascading Stylesheets, but actual standards lists over 500 CSS properties! The exercises that follow the tables concentrate mainly on using the code mentioned in these tables.

## Font/Text Definitions

| Definition | *relates to* | *example* |
|---|---|---|
| font-family | typeface. | `h2 {font-family: Arial;}` |
| font-style | the style of the text<br>normal, italic, small caps etc | `h3 {font-style: small caps;}` |
| font-size | the size of the text<br>in points (pt), inches(in) pixels(px), %, em, rem | `h4 {font-size: 20px;}` |
| font-weight | text density – extra light, light, demi-light, medium, bold, demi-bold, extra-bold | `a:link {font-weight: demi-light;}` |
| font-variant | variation of the normal font – specify normal or small caps | `h2 {font-variant: small-caps;}` |
| text-align | the alignment of text – left, centre or right | `h1 {text-align: center;}` |

| | | |
|---|---|---|
| **text-decoration** | other text formatting – italic, blink, underline, line-through, overline, or none | `a:visited {text-decoration: blink;}` |
| **text-indent** | margins – most often used with <p><br>if used with <p> use closing tag </p>!<br>values are “in”, “cm” or “px” | `p {text-indent: 1in;}` |
| **word-spacing** | the amount of spaces between words -<br>values are pt, in, cm, px or % | `p {word-spacing: 10px;}` |
| **letter-spacing** | space between letters -<br>values are pt, in, cm, px or % | `p {letter-spacing: 2pt;}` |
| **text-transform** | transformation of the text –<br>capitalise, uppercase or lowercase | `p {text-transform: uppercase;}` |
| **color** | colour of text. | `h3 {color: #FFFFFF;}`<br>`h3 {color:rgb (255, 255, 255);}`<br>`h3 {color: navy;}` |

## Margin/Background Definitions

| Definition | *relates to* | *example* |
|---|---|---|
| **margin-left**<br>**margin-right**<br>**margin-top** | space around the "page". Values are in pt, in, cm, or px<br>used with ***body*** entire page is affected<br>used with ***p*** only paragraph is affected | `body {margin-left: 2in;}`<br>`p {margin-right: 12cm;}`<br>`body {margin-top: 45px;}` |
| **margin** | all three margin command combined. Sequence is top, right, left | `p {margin: 3in 4cm 12px;}` |
| **line-height** | space between lines of text. Values are pt, in, cm, px or % | `p {line-height: 10px;}` |
| **background-color** | page's background colour in hexadecimal or word code – “transparent” is also a value | `body {background-color: #ffffff;}` |
| **background-image** | the background image for pages<br>specify URL where image is kept | `body {background-image: url(image.jpg);}` |
| **background-repeat** | how the image will tile – repeat-x, repeat-y, or no-repeat | `body {background-repeat: repeat-y;}` |
| **background-attachment** | how the image will react to a scroll. | `body {background-attachment: fixed;}` |

## Position/Size Definitions:

| Definition | *relates to* | *example* |
|---|---|---|
| **position** | the placement of an element on the page. Values are absolute (placement) or relative (to other elements) | `img {position: absolute;}` |
| **left** | amount of space allowed from the left of the browser screen when positioning an item – values are pt, in, cm, px or % | `img {position: absolute; left: 20px;}` |
| **top** | amount of space allowed from the top of the browser screen when positioning an item – values are pt, in, cm, px or % | `img {position: absolute; left: 20px; top: 50px;}` |
| **width** | width of image or page division – values are pt, in, cm, px or % | `img {position: absolute; left: 20px; top: 50px; width: 300px;}` |
| **height** | height of image or page division – values are pt, in, cm, px or % | `img {position: absolute; left: 20px; top: 50px; width: 300px; height: 200px;}` |
| **overflow** | what to do when an image or text is too large – values are visible, hidden, scroll | `img {position: absolute; left: 20px; top: 50px; width: 300px; height: 200px; overflow: hidden;}` |
| **z-index** | an item's position in the layering structure – the lower the number, the lower the layer | `img {position: absolute; left: 20px; top: 50px; width: 300px; height: 200px; overflow: hidden; z-index: 10;}` |

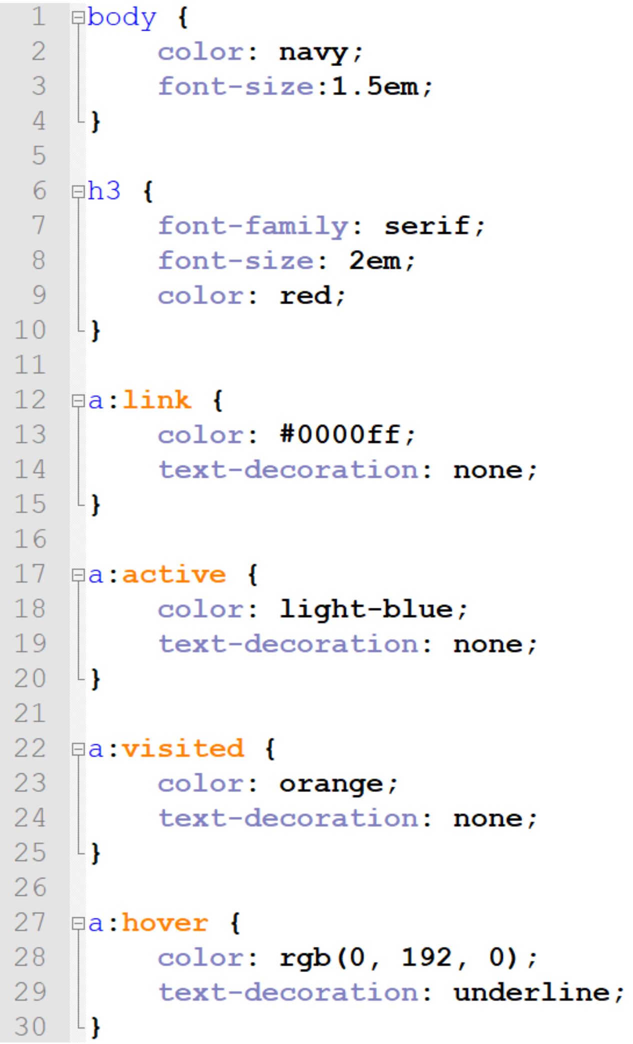
# Exercises

*The fastest way to becoming familiar with any syntax is to actually USE it. In this lab it is EXPECTED of you that you HAND-CODE most of the examples, except where specific other stylesheets are mentioned; in that case you use the one(s) provided on Moodle. When asked to test your code, use Google Chrome as per the previous lab, but also give them a try on Mozilla Firefox, which is also another good browser; browser differences should be taken into account as well when creating your website!*

## Exercise 1: Anchor tag styles

This stylesheet will manipulate the appearance of **heading level 3**, the **different link states** and also specifies what the **colour** of the **body text** is going to be.

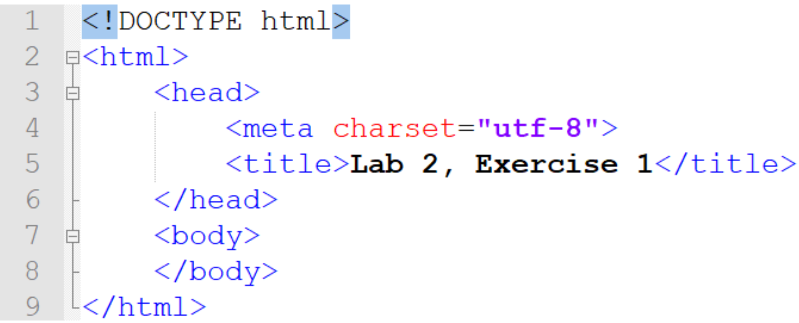
- **Open Notepad++** and create a file called **wk2ex1.css**. Note the extension has to be “.css”.
- Type in the following code:



```
body {
    color: navy;
    font-size:1.5em;
}

h3 {
    font-family: serif;
    font-size: 2em;
    color: red;
}

a:link {
    color: #0000ff;
    text-decoration: none;
}

a:active {
    color: light-blue;
    text-decoration: none;
}

a:visited {
    color: orange;
    text-decoration: none;
}

a:hover {
    color: rgb(0, 192, 0);
    text-decoration: underline;
}
```

- **Question:** What is an **"em"**?
  - **Answer:** It is a **sizing unit** that we will discuss in a future lecture. Basically, this is indicating that that body font-size will be 1.5 times larger than its parent element's font-size (in this case the default font-size).
  - The h3 is set to a font-size of 2em, meaning it will be 2 times larger than its parent element.

- **Create** a **new webpage**. Recall Lab 1 and the starting structural HTML code required:



```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Lab 2, Exercise 1</title>
    </head>
    <body>
    </body>
</html>
```

- Inside the **head** element, **add** the link to the CSS:

```
<link href="wk2ex1.css" rel="stylesheet">
```

- Inside the **body** element, **add** these lines of HTML code:

```
<main>
    <h3>Lab 2 - Exercise 1</h3>
```

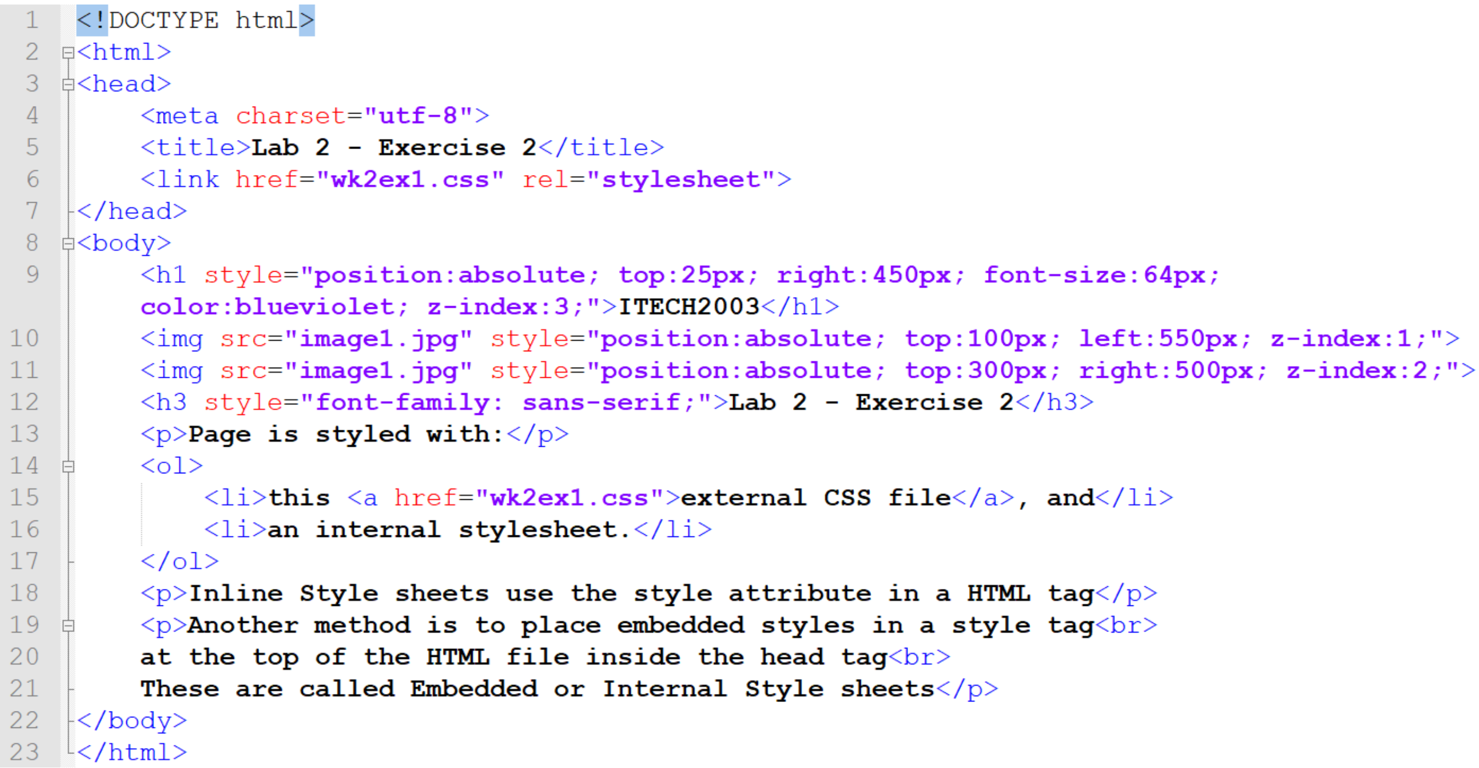
```
        <p>Page is styled with this <a href="wk2ex1.css">external CSS file</a></p>
        <p>Notice how the colours can be defined by:</p>
        <ol>
            <li>keyword names (eg: navy, red, etc),</li>
            <li>hexidecimal values (eg: #000080, #ff0000, etc), or</li>
            <li>rgb values (eg: rgb(0, 0, 128), rgb(255, 0, 0), etc)?</li>
        </ol>
    </main>
```

- **Save** as **wk2ex1.html** and **test** the page in a browser to see what the code in the style sheet did to the page
- **Question:** What are each of the **a:link**, **a:active**, **a:visited**, and **a:hover** element selectors doing?
    - **Answer:** They define the styles for the anchor element under different states.
        - **a:link** defines the styles under its default state.
        - **a:active** defines the styles of the link currently being clicked on.
        - **a:visited** defines the styles when that webpage has been previously visited by the user.
        - **a:hover** defines the styles when the user hovers their mouse over the link. No effect on touch devices, but still intuitive for desktop users.

## Exercise 2: External and Inline styles

This exercise makes use of both a **linked** external style sheet and **inline** styles – the linked style sheet is the same as the one used in exercise 1.

- **Download** the **Lab 02 Files** zip file from **Moodle**.
- This exercise will use the **image1.jpg**. Place it in the same location you will save your HTML file.
- **Open Notepad++** and create a file called **wk2ex2.html**
- Type in the following code:



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Lab 2 - Exercise 2</title>
    <link href="wk2ex1.css" rel="stylesheet">
</head>
<body>
    <h1 style="position:absolute; top:25px; right:450px; font-size:64px; color:blueviolet; z-index:3;">ITECH2003</h1>
    <img src="image1.jpg" style="position:absolute; top:100px; left:550px; z-index:1;">
    <img src="image1.jpg" style="position:absolute; top:300px; right:500px; z-index:2;">
    <h3 style="font-family: sans-serif;">Lab 2 - Exercise 2</h3>
    <p>Page is styled with:</p>
    <ol>
        <li>this <a href="wk2ex1.css">external CSS file</a>, and</li>
        <li>an internal stylesheet.</li>
    </ol>
    <p>Inline Style sheets use the style attribute in a HTML tag</p>
    <p>Another method is to place embedded styles in a style tag<br>
    at the top of the HTML file inside the head tag<br>
    These are called Embedded or Internal Style sheets</p>
</body>
</html>
```

- **Save** your file and **test** it.
- **Question:** Identify and explain the parts of the above code that look after the **`position`** of the **h1** and **img** elements? ***(hint: try resizing your browser window horizontally.)***
  - **Answer:**
    - The **h1 element** is positioned absolute 25 pixels from the top and 450 pixels from the right side of the browser window.
    - The **first image** is positioned absolute 100 pixels from the top and 550 pixels from the left side of the browser window.
    - The **second image** element is positioned absolute 300 pixels from the top and 500 pixels from the right side of the browser window.
- **Question:** How does **`z-index`** work?
  - **Answer:** It works like a stack of layers. The higher in the stack, the more in the foreground the element appears. That is why the first image is always in the background compared to the higher z-index h1 and second image.

## Exercise 3: Two Column Layout

**When you have multiple of the same element in your HTML document that you require to be formatted with different styles, how can you solve this?**

For example, imagine your webpage has three different section elements, and you want each section to have a different background colour to clearly separate them. If you apply a background-colour to the section element selector, it will affect ALL section elements in the HTML. This is where an id or class becomes beneficial!

Developers can apply an id or class with any name they choose to any HTML element, then address the styles specifically for that id or class in the CSS.

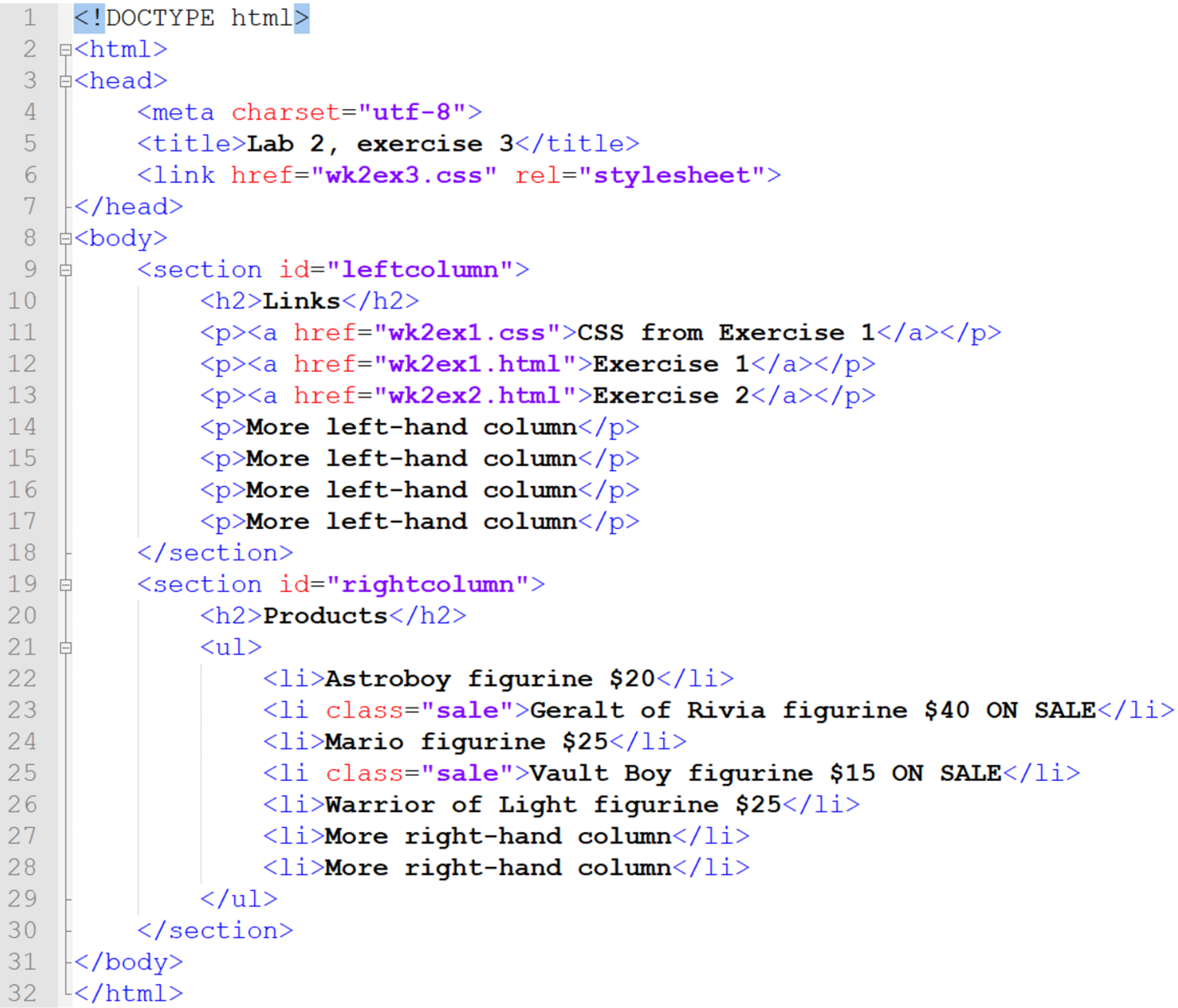
- **Question:** But when is it appropriate to use an **id** and when a **class**? What is the difference?
  - **Answer:**
  - An **`id`** with a unique name can only be used **once** on an element on each webpage
  - A **`class`** can be used multiple times on as many element as you want.

| Type | *Syntax* | *Usage Description* |
|---|---|---|
| **id** | **HTML:**<br>`<h1 id="top">About Us</h1>`<br>**CSS:**<br>`#top { font-size:2em; }` | When you only want one HTML element to use a unique set of styles you should use an **`id`**. This will prevent you from using that CSS id selector on another HTML tag. Also, because an id is unique, you can provide href links to specific parts of your webpage containing these unique id's.<br>`<a href="#top">Return to Top</a>` |
| **class** | **HTML:**<br><h2 class="centered">sub-heading</h2><br>**CSS:**<br>`#centered { text-align:center; }` | When you want the styles created usable on multiple elements on the same page. For example, we could apply a class called **`"centered"`**, to multiple HTML tags. Then all elements with this class would use the styles from the **`.centered`** CSS class selector. |

In this exercise you will create two **HTML ids** with an associated **CSS id selector**, that will display a **2-column layout**. An **id** or a **class** can be any name you choose, give them meaningful names. In CSS, to identify each:

- **id** = prefix the id name with a hash **#**
- **class** = prefix the class name with a period **.**

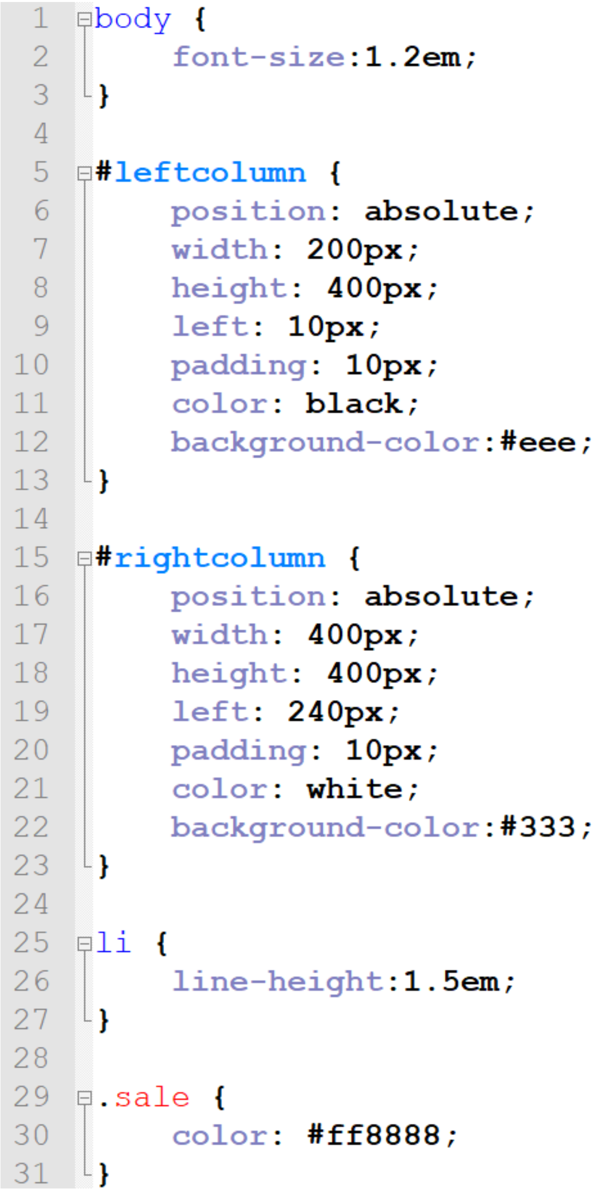
- Enter the following code into a new html file called **wk2ex3.html**



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Lab 2, exercise 3</title>
    <link href="wk2ex3.css" rel="stylesheet">
</head>
<body>
    <section id="leftcolumn">
        <h2>Links</h2>
        <p><a href="wk2ex1.css">CSS from Exercise 1</a></p>
        <p><a href="wk2ex1.html">Exercise 1</a></p>
        <p><a href="wk2ex2.html">Exercise 2</a></p>
        <p>More left-hand column</p>
        <p>More left-hand column</p>
        <p>More left-hand column</p>
        <p>More left-hand column</p>
    </section>
    <section id="rightcolumn">
        <h2>Products</h2>
        <ul>
            <li>Astroboy figurine $20</li>
            <li class="sale">Geralt of Rivia figurine $40 ON SALE</li>
            <li>Mario figurine $25</li>
            <li class="sale">Vault Boy figurine $15 ON SALE</li>
            <li>Warrior of Light figurine $25</li>
            <li>More right-hand column</li>
            <li>More right-hand column</li>
        </ul>
    </section>
</body>
</html>
```

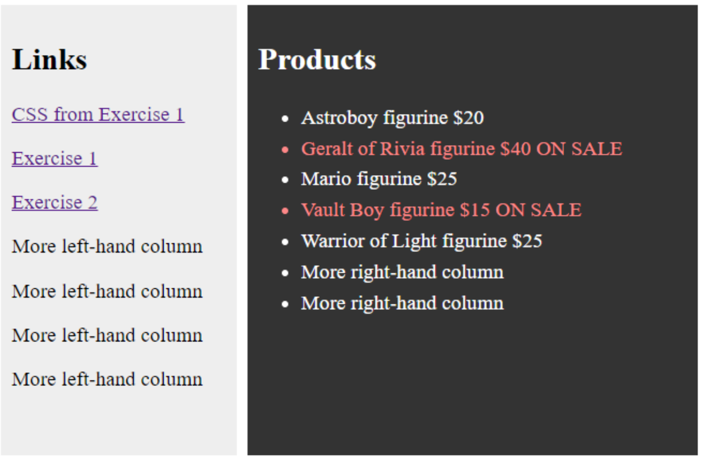
- As you can see, we have assigned an **id** called **leftcolumn** to one **section** element, and another called **rightcolumn** to a different **section** element. The **class** called **sale** has been assigned to two **list item** elements, remembering that it will not affect any of the other list item elements that it is not assigned to.
- Recall that a **class** **is a CSS selector** that can be used over and over on a page. An **id** on the other hand should be used only once (according to the standards).

- Create the following style sheet and call it **wk2ex3.css.**



```css
body {
	font-size:1.2em;
}

#leftcolumn {
	position: absolute;
	width: 200px;
	height: 400px;
	left: 10px;
	padding: 10px;
	color: black;
	background-color:#eee;
}

#rightcolumn {
	position: absolute;
	width: 400px;
	height: 400px;
	left: 240px;
	padding: 10px;
	color: white;
	background-color:#333;
}

li {
	line-height:1.5em;
}

.sale {
	color: #ff8888;
}
```

*The layout we've just created looks like a clear example of how positioning with CSS works, but in reality it is, as yet, far from perfect. There are numerous areas that we will not be addressing but you might like to look into them:*



Links

CSS from Exercise 1

Exercise 1

Exercise 2

More left-hand column

More left-hand column

More left-hand column

More left-hand column

Products

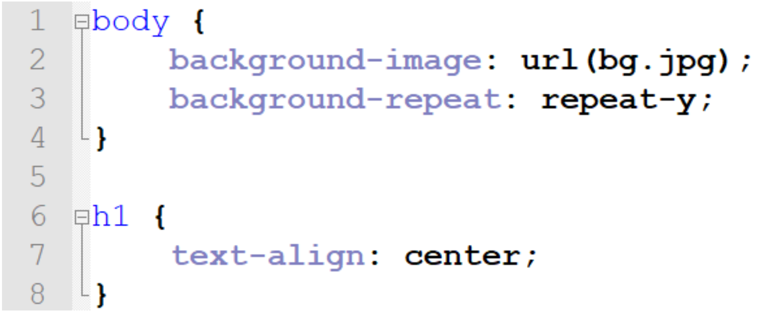
- Astroboy figurine $20
- Geralt of Rivia figurine $40 ON SALE
- Mario figurine $25
- Vault Boy figurine $15 ON SALE
- Warrior of Light figurine $25
- More right-hand column
- More right-hand column

- More formatting properties, such as use of margins etc.
- “Stretchability” – the ability to grow with the page size.
- Inheritance – especially important when you start dealing with relative positioning.

## Exercise 4: Images

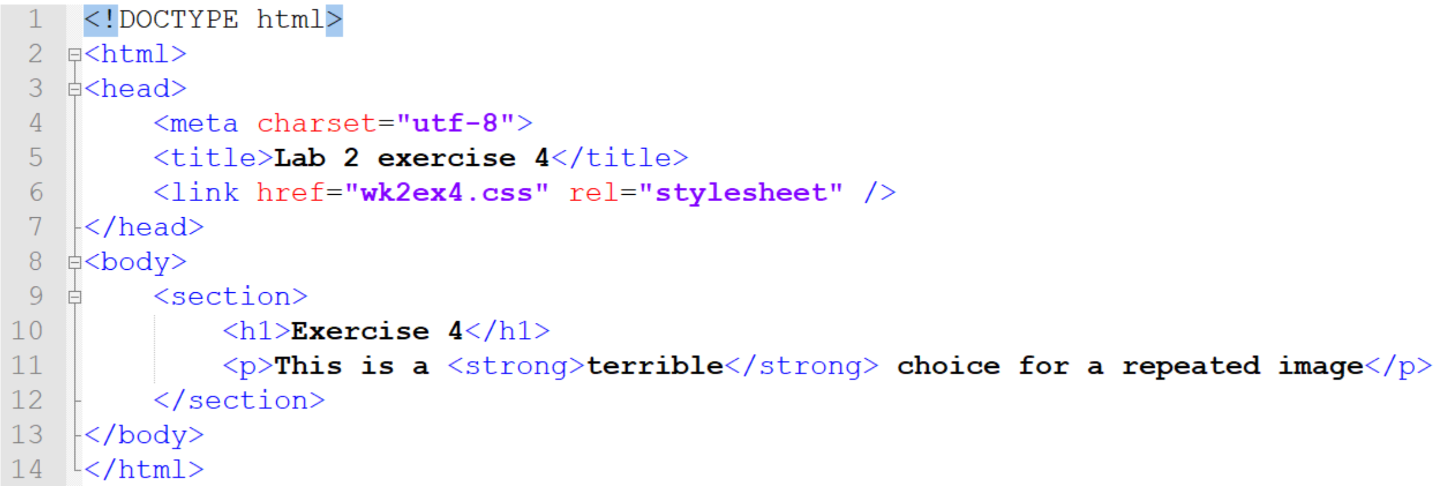
This exercise will use an image on a webpage, this time as a repeated background image.

- This exercise will use the **bg.jpg**. Place it in the same location you will save your HTML and CSS files.
- Create a style sheet called **wk2ex4.css** and enter the following code:



```
body {
    background-image: url(bg.jpg);
    background-repeat: repeat-y;
}

h1 {
    text-align: center;
}
```

- Create the associated **wk2ex4.html** page as follows:



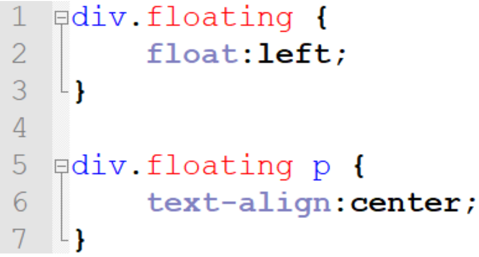
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Lab 2 exercise 4</title>
    <link href="wk2ex4.css" rel="stylesheet" />
</head>
<body>
    <section>
        <h1>Exercise 4</h1>
        <p>This is a <strong>terrible</strong> choice for a repeated image</p>
    </section>
</body>
</html>
```

- As always, **save** and **test**…
- **Make the bg.jpg image fill the entire page**!
  - (*Hint: the* **`repeat-y`** *refers to the vertical repeating*)
- **Question:** What does the **<strong>** element do?
  - **Answer:** It makes the content between the opening and closing tags bolded to the viewer. Specifically, according to the HTML standards, it means that its contents have strong importance, seriousness, or urgency.

## Exercise 5: Floating Images

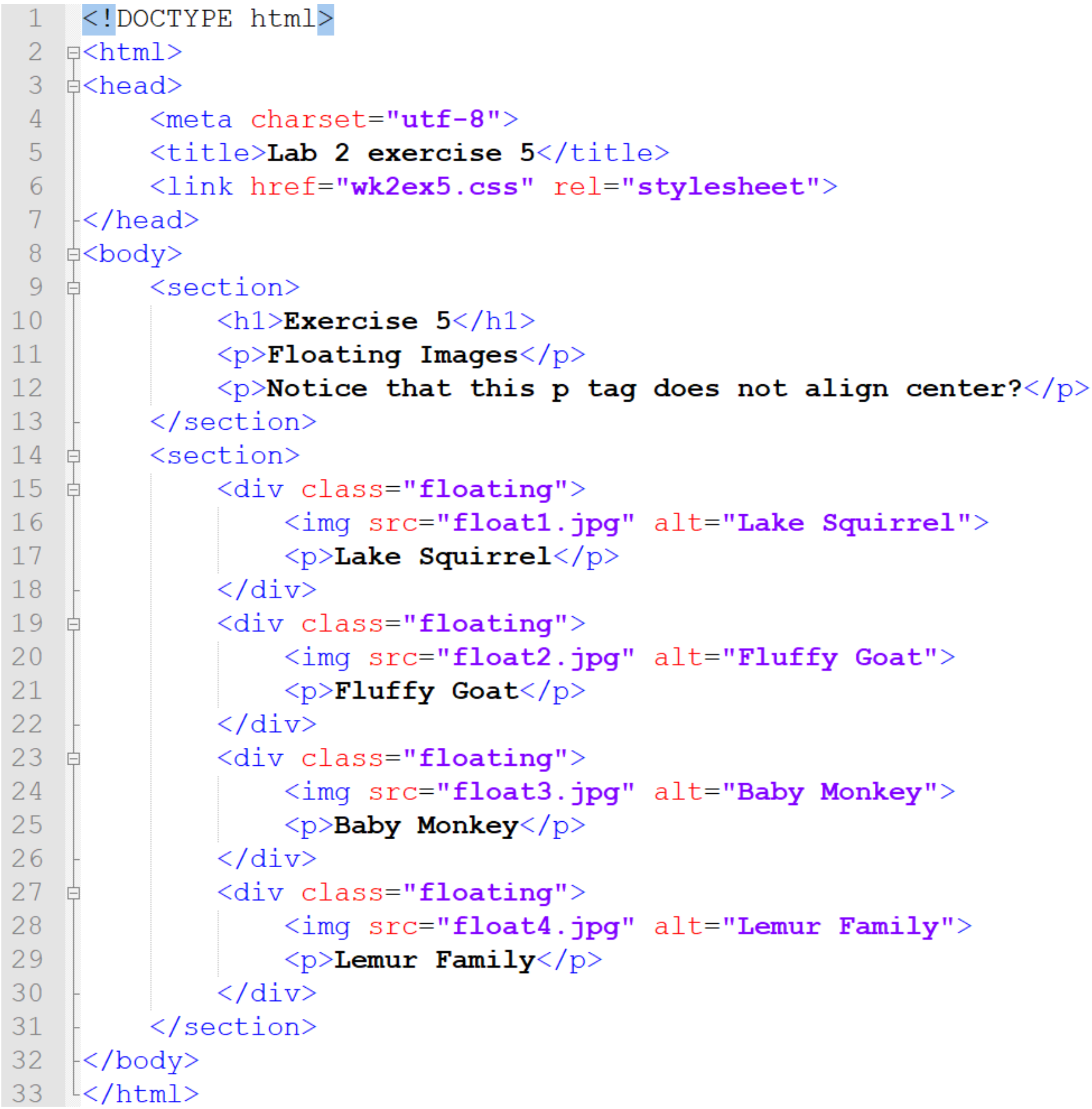
Using the **“float”** property for elements will automatically allow the contents of the element overflow to the next “line”. In this example you will use it for images that may not fit across the page (especially on lower resolutions).

- This exercise needs images **float1.jpg, float2.jpg, float3.jpg, and float4.jpg**. Place them in the same location you will save your HTML and CSS files.
- First of all, **create a class** called **.floating** in a style sheet (shown right) (in the same class we also define what needs to happen with a paragraph)
- **Save** this style sheet as **wk2ex5.css**
- In the HTML page we need to create a **<div>** for each image we want to place on the page. Inside the **<div>** we place the image (which will be “**FLOAT**”ed to the left), followed by a paragraph of the caption.



```css
div.floating {
    float:left;
}

div.floating p {
    text-align:center;
}
```

- **Question:** what is a **<div>** element?
    - **Answer:** It is a block element, like header, section, and footer, but it has no semantic meaning. They should only be used when a semantic element is not appropriate for the content.
- Create the following HTML page named **wk3ex5.html**



```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Lab 2 exercise 5</title>
    <link href="wk2ex5.css" rel="stylesheet">
</head>
<body>
    <section>
        <h1>Exercise 5</h1>
        <p>Floating Images</p>
        <p>Notice that this p tag does not align center?</p>
    </section>
    <section>
        <div class="floating">
            <img src="float1.jpg" alt="Lake Squirrel">
            <p>Lake Squirrel</p>
        </div>
        <div class="floating">
            <img src="float2.jpg" alt="Fluffy Goat">
            <p>Fluffy Goat</p>
        </div>
        <div class="floating">
            <img src="float3.jpg" alt="Baby Monkey">
            <p>Baby Monkey</p>
        </div>
        <div class="floating">
            <img src="float4.jpg" alt="Lemur Family">
            <p>Lemur Family</p>
        </div>
    </section>
</body>
</html>
```

- Now **save** and **test** the page.To see the effect of a smaller sized browser window, **resize your browser window** until 3 images fit across it. Observe what has happened to the 4th image.
- **Resize even further** and observe what happens.

## Exercise 6: Fun with Position

- This exercise requires the **positioning_css.html** from the Moodle files.
- **Open** this file in **Notepad++**. **Save** as **wk1ex6.html**
- **Open** in **Chrome** and as you might have noticed, the text at the top of the page is partially hidden.
- **Make changes to the CSS** code until this text is visible and that each rectangular shaped element filled with text is displayed in such a way that not one element blocks any other element.

## Exercise 7: Internal to External CSS

- Make a **new save** of wk1ex6.html called **wk1ex7.html**. Change the HTML in such a way that the internal CSS (using the style attribute) is placed into an external CSS file and the HTML code is altered to reflect that change.

- To get you started, the **nav** element that is written like this:

```
<nav style="background-color: #292929; color: white; position: absolute;
left: 0px; top: 5%; width: 150px; height: 80%; padding: 5px; padding-
right: 20px;">
    <p>This will be the navigation 'column', which will run down the
left of the page.</p>
</nav>
```

- Should be altered to:

HTML file

```
<nav>
    <p>This will be the navigation
'column', which will run down the left
of the page.</p>
</nav>
```

- **Create a new CSS file**, and place the code on the right into it, to separate HTML and CSS. **NOTE:** your CSS may be different depending on your alterations in Exercise 6.
- **Do not forget to create** a `<link>` between the HTML and CSS files!
- Lastly, **rearrange the HTML** so the document flows in a logical order, starting with the <header> and ending with the <footer>!

CSS file

```
nav {
    background-color:#292929;
    color:white;
    position:absolute;
    left:0px;
    top:20%;
    width:150px;
    height:75%;
    padding:5px;
    padding-right:20px;
}
```

## Exercise 8: Career research for Assignment 1

This exercise is designed to help you continue with your Design Document assignment that you should have started in the first lab. The due date is still some time away, but it is quite a large assignment. Today you will focus on identifying and summarising your IT career focus for your personal website.

Remember that as part of the hypothetical scenario, the website must be about yourself and your future Information Technology (IT) career – so choose an IT career to showcase on the website that you are passionate about. It must be an IT related career for the purposes of the scenario. If you already know what you IT career you want to do in the future, great! It will make it easier, but you still need to do the research. If you are unsure, these guidelines will help you to explore different IT careers and required skills.

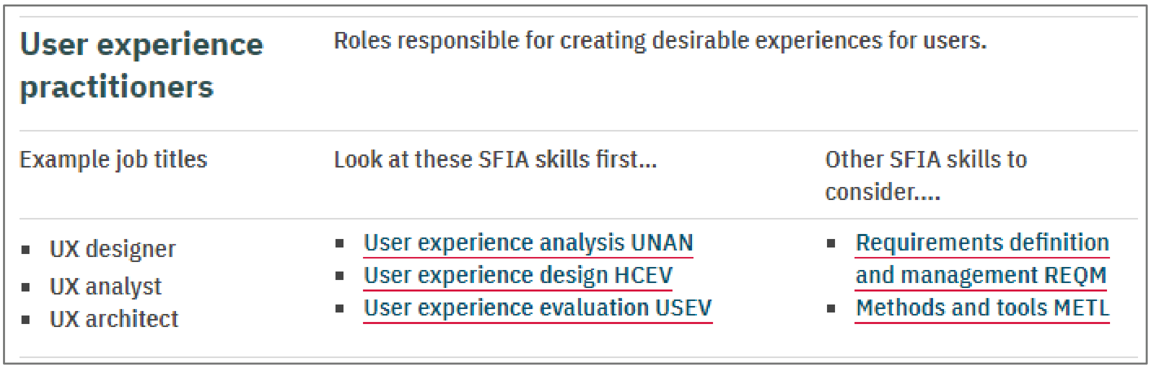
Once again, review the requirements from the **“Assignment Specifications for Assignment 1 and 2”** PDF, and continue with the **Design Document** that you began in Lab 1. Some helpful information on how to research and what to include in the Career section is provided below.

## 1.2. Career

In this section you must outline your future IT career that will be a primary focus for your personal website.

- **Visit the APS website** https://aps-career-pathfinder.digitalprofession.gov.au/. There are a few ways you can use this website:
  1. **Use the search box** to search for roles. **Type in keywords** that match your preferences, for example: “analyst”, “cloud”, “data”, “design”, “network”, “security”, “software”, or “test”.
  2. **Scroll down and explore** the different roles. If you **hover your mouse over each role** that you may be interested in, it will give you a summary of the pathway.
  3. **Click the Add your skills button** to provide information to the system to find suitable roles for you. **Add some skills** (in the menu or through search) – you can be realistic, and/or add skills you desire but are yet to learn (the scenario allows you to be inventive and embellish the truth). It will match careers to your skill set with some 100% matches, and others might require additional skills to satisfy the career path.
- **Visit the SFIA website** https://sfia-online.org/en/tools-and-resources/standard-industry-skills-profiles/sfia-8-skills-for-role-families-job-titles. There are a few ways you can use this website:
  1. **Press CTRL-F** (Win) / **CMD-F** (Mac). **Type in keywords** that match your preferences, for example: “analyst”, “cloud”, “data”, “design”, “network”, “security”, “software”, or “test”. **Cycle** through the matches.
  2. **Scroll down and explore** the different example job titles for different career families. Read the overall summaries and required SFIA skills.
- Once you have found career paths/job title(s) that sound interesting to you, **compare them via the APS and SFIA websites to view the career and skills required**. Note that they may not have the same titles, as IT jobs can have slightly different titles depending on the entity.
  - For example, on SFIA, the “UX designer” job sounded interesting to Joe, so he visits the APS website and searches for “UX Designer”, and it gives a few hits to “Interaction Designer”, “User Interface Designer”, and so on. Joe uses the information from both websites. SFIA lists 3 main / 2 consider skills and APS also lists 3 essential / 2 desirable skills for a similar but not identical career (*see next page for example*). They are all important and will help you write up your Career section.
- Now focus on the **one career/job that appeals to you the most**, and using APS and SFIA as the primary reference source, **describe the role and summarise the required and desirable skills** for section 1.2 Career in your Design Document. You can also **click on the skill links** for more details about each SFIA skill.
- To enhance your discussion, utilise additional resources and cite them as well. Reference all sources **with in-text APA citations** (and APA references at the end of the document).

Example search for **UX designer** on **SFIA**:

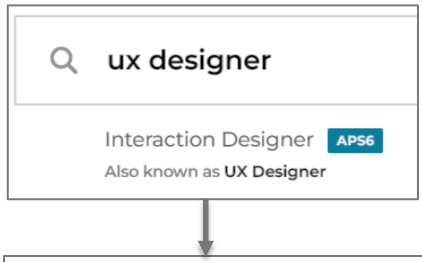


## User experience practitioners

Roles responsible for creating desirable experiences for users.

| Example job titles | Look at these SFIA skills first... | Other SFIA skills to consider.... |
|---|---|---|
| ▪ UX designer<br>▪ UX analyst<br>▪ UX architect | ▪ User experience analysis UNAN<br>▪ User experience design HCEV<br>▪ User experience evaluation USEV | ▪ Requirements definition and management REQM<br>▪ Methods and tools METL |

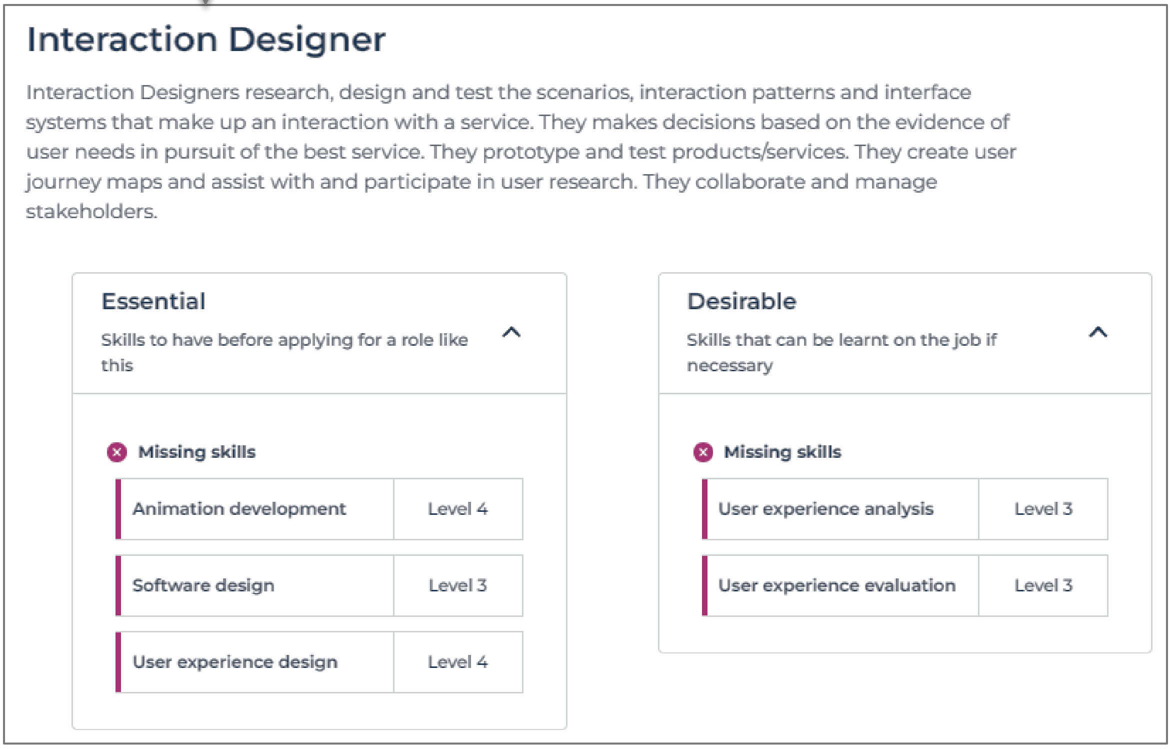
Equivalent search on **APS**:



ux designer

Interaction Designer APS6
Also known as **UX Designer**

You can identify that three of the skills are the same, but what SFIA lists as essential (“Look at these SFIA skills first”), APS only identifies one of these as essential, but places two of these skills in the desirable list.



## Interaction Designer

Interaction Designers research, design and test the scenarios, interaction patterns and interface systems that make up an interaction with a service. They makes decisions based on the evidence of user needs in pursuit of the best service. They prototype and test products/services. They create user journey maps and assist with and participate in user research. They collaborate and manage stakeholders.

**Essential**
Skills to have before applying for a role like this

Missing skills

| | |
|---|---|
| Animation development | Level 4 |
| Software design | Level 3 |
| User experience design | Level 4 |

**Desirable**
Skills that can be learnt on the job if necessary

Missing skills

| | |
|---|---|
| User experience analysis | Level 3 |
| User experience evaluation | Level 3 |

- For all assignment work you should make the most of your tutors while they are present in the lab class with you, and show them your progress as well as see if they have any suggestions. Remember they cannot do the work for you, but they can give you advice or refer you to online material that could help you.