

CSS - Exercises

Exercise 1

In this exercise, you will write some CSS to alter the appearance of the *Balloons* web page you made.

The CSS file

- Open your code editor and create a new, blank file.
- Save this file in the *balloons* folder as **balloons.css**.
 - Note, the file name should be all lower-case
 - If you like, you can save the CSS file in a sub-folder called *styles* or similar, but make sure you include the folder name in the *href* of the *link* tag
- Open *index.html* from the *balloons* folder in your code editor
- Add a **link** tag to the *head* element (as shown in the presentation)
 - The **rel** value should be *stylesheet*
 - The **href** value should be *balloons.css* (i.e. the file you just created)
 - Don't forget that the *link* is a *void* element, so has no closing tag
- Save both of your files.

The *h1* element

First, we will change the appearance of the *h1* element

- Add an empty *rule* to *balloons.css*. The *selector* should be **h1**:

```
h1 {  
  
}
```

- Add a *font-size* declaration to the empty *rule*. The *value* should be **3em**:

```
h1 { font-size:  
    3em;  
}
```

- Save all of your files and open *index.html* in your browser. The *h1* text should be larger than it was previously.
- In *balloons.css*, add another declaration to the *h1* rule. This time, the *property* is *color* and the *value* will be **#660099**:

```
h1 { font-size:  
    3em; color:  
    #660099;  
}
```

- Save your CSS file and preview *index.html* in your browser. The *h1* text should now be purple.

- In *balloons.css*, add another declaration to the *h1* rule. This time, the *property* is *font-weight* and the *value* will be **400** (or **normal** if you prefer):

```
h1 {
  font-size: 3em;
  color: #660099;
  font-weight: 400;
}
```

- Save your CSS file and preview *index.html* in your browser. The *h1* text should now be nonbold.
- Finally, we will add a *font-family* declaration for the *h1*. We will use a *monospace* font stack:

```
h1 {
  font-size: 3em;
  color: #660099;
  font-weight: 400;
  font-family: "Courier New", Courier, monospace;
}
```

Remember, when a font name contains spaces, we must wrap it in double-quotes

- Save your CSS file and preview *index.html* in your browser. The *h1* font should have changed to look like this:

Balloons Ltd.

- If it does not look like the above, double-check your CSS and ensure that the syntax/spelling/case is correct

The *h2* element

- Add another empty *rule* to *balloons.css*. This time, the *selector* should be **h2**:

```
h2 {
}
```

- Using the *h1* rule and the presentation examples as a guide, add declarations to set the following properties for the *h2* elements:

- The *color* will be **#99007f**
- The *font-style* will be **italic**
- The *font-size* will be **1.5em**
- The *font-family* will be **Arial, Helvetica, sans-serif**

- Save your CSS file and preview *index.html* in your browser. The *h2* font should have changed to look like this:

What We Do

- If it does not look like the above, double-check your CSS and ensure that the syntax/spelling/case is correct

The *p* element

- Add another empty *rule* to *balloons.css*. This time, the *selector* should be **p**:

```
p {  
  
}
```

- Using your previous work as a guide, add declarations to set the following properties for the *p* elements:
 - The *color* will be **#555555**
 - The *font-family* will be **Arial, Helvetica, sans-serif**
- Save your CSS file and preview *index.html* in your browser. The *p* font should be displayed as Arial, in a dark-grey colour.
 - If it does not, double-check your CSS and ensure that the syntax/spelling/case is correct

Extra - Grouped selectors

We have set the same *font-family* for the *h2* and *p* elements. As we saw in the presentation, we can combine these 2 declarations into a single *rule*.

- Add a new, empty rule to *balloons.css*. This time, the selector will be **h2, p**
- Inside the rule, add a declaration to set the *font-family* to **Arial, Helvetica, sans-serif**

```
h2, p { font-family: Arial, Helvetica, sans-  
serif; }
```

- When this is done, remove the *font-family* declarations from the *p* and *h2* rules
- Save your files and preview the changes – There should be no difference!

Extra - Color shortcuts

- When using *hexadecimal notation* to specify colours, the six characters used in the code are actually 3 pairs of characters

– #660099 = # 66 00 99

- When we have 3 identical pairs of values, we can abbreviate the code to 3 characters

– #660099 = #609

- But if any of the 3 pairs are not identical, we can't abbreviate it

– #99007f = # 99 00 7f

– Can't abbreviate as 7f is not an identical pair.

- In *balloons.css*, locate the declaration that sets paragraph text to #555555

– Change the *value* to **#555**

- In *balloons.css*, locate the declaration that sets *h1* text to #660099

– Change the *value* to **#609**

- Save your files and preview the changes
 - Again, there should be no difference!

Extra - Overriding defaults

To demonstrate the concept that HTML does not control the appearance of elements, we will modify the styles for the *em* and *strong* elements.

By default, the *em* element is displayed as *italic* text.

- Add a *rule* to *balloons.css* that sets it's font-style to **normal**

Save and preview your page. The *em* elements should now look just like the other text. By default, the *strong* elements are displayed as **bold** text

- Add a *rule* to *balloons.css* that sets it's font-weight to **normal**

Save and preview your page. The *strong* elements should now look just like the other text.

As you can see, although these elements have default display properties, we have full control over those properties with CSS.

Going further...

If you like, you can reverse the style of the *strong* and *em* elements!

- Add a declaration to the *strong* rule that sets the *font-style* to **italic**
- Add a declaration to the *em* rule that sets the font-weight to **700** (or **bold**)

Save and preview your page. The *strong* elements should now look like *em* elements and vice versa.

When you are finished this experiment, remove the *strong* and *em* rules from *balloons.css*

Exercise 2

In this exercise, you will use *CSS Inheritance* to simplify the CSS code you have written so far.

- Add an empty *rule* to the top of *balloons.css*. This time, the *selector* should be **body**:

```
body {  
  
}
```

- Apart from the *h1* element, we want all of our page to use the *Arial* font-stack. So we will set the *font-family* property for the *body* element, instead of setting it for *p* and *h2*.

```
body { font-family: Arial, Helvetica, sans-  
serif; }
```

- When you have done this, you can remove all other *Arial* font-family declarations from the style sheet (i.e. from the *p* and *h2* rules)
 - But don't modify the *h1* rule's *font-family* declaration!
- Save your files and preview the changes
 - Again, there should be no difference!
- Next, we will set a default *color* for all elements on the page. The colour we will use is **#555**. We are currently only using this colour for the *p* elements, but when we use other tags (lists, etc), we will want them to be the same colour.

```
body { font-family: Arial,  
        Helvetica, sans-serif;  
        color: #555;  
}
```

- When you have done this, you can remove the *color* declaration from the *p* rule in *balloons.css*
- Save your files and preview the changes
 - Again, there should be no difference!
- Note, the *p* rule is now empty, as it is inheriting everything from *body*. You can leave this empty rule in your style sheet if you like (it will not break anything). Or you can remove it to keep things tidy. The choice is yours.