A picture containing logo

Description automatically generated

Distance Learning Activity 1

ICTWEB519 Develop complex web page layouts

ICTWEB520 Develop complex cascading style sheets

### 

**Copyright 2022**

© Australian College of Business Intelligence

All rights reserved

Version: 22.1

Date Modified: December 2022

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Australian College of Business Intelligence.

**Disclaimer:**

The Australian College of Business Intelligence does not invite reliance upon, nor accept responsibility for, the information it provides. The Australian College of Business Intelligence makes every effort to provide a high-quality service. However, neither the Australian College of Business Intelligence, nor the providers of data, gives any guarantees, undertakings or warranties concerning the accuracy, completeness or up-to-date nature of the information provided. Users should confirm information from another source if it is of sufficient importance for them to do so.

# 1. Instructions to Learners

## Purpose of this distance learning activity

This activity requires the learners to code in HTML and CSS based on the following topics:

* HTML file creation
* Defining CSS rules using external, embedded and inline stylesheet
* Colours
* Fonts
* Selectors
* Div element
* float

## What you are required to do

For this distance learning activity, you are required to answer questions and create small web pages.

## Resources

* Please refer to the learner guides.

## How to submit your Distance Learning Activity

**File naming convention**

When submitting your Distance Learning Activity, use the following naming convention

* Unit of Competency code (i.e., ICTWEB519\_520)
* ‘DLA’ followed by the week number (ie., 5)
* Your first name and initial of last name. Capitalise only the first letter of names. Then insert a dash, followed by submission attempt.

Hence as an example of incorporating the three points above your file name or subject line should appear this way: *ICTWEB519\_520* *DLA 1 – JessieA #1*

Create a website project folder called using the naming convention specified above. This folder will be used to store the following

* This distance learning activity
* Other website project folder as required by the questions below

Once completed, compress the website project folder in zip format and then submit. Always submit your distance learning activity on time by the due date. Your completed distance learning activity must be uploaded onto Moodle by 11.55pm on Sunday of the current week.

**2. Questions**  
  
Answer the questions below using the space provided:

* Answer all parts of each question
* Use your own words and give examples wherever possible
* The quality of your answer is more important than how long it is
* Enter your answers in this document

You may use various sources of information to inform your answers, including your resources provided by ACBI, books, and online sources. You must acknowledge and cite your sources.

| Question 1. Create a folder called question1. Inside that folder, create an HTML document called “test.html” that has the following mark-up: |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <title>A test web page</title>  <link rel="stylesheet" href="main.css">  </head>  <body>  <p>A simple large font sentence.</p>  </body>  </html> |
| Then create a file called main.css in the same directory as the HTML file and give it the following code: |
| p {  color: red;  font-style: italic;  } |
| Open the HTML document in the browser, and the paragraph text should be displayed in a large red font. |

| Question 2. In the space provided below, rewrite the test.html document created in the previous activity so that it removes the external style sheet and implements the embedded style sheet instead: |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <title>A test web page</title>  <!-- <link rel="stylesheet" href="main.css"> -->  <style>  p {  color: red;  font-style: italic;  }  </style>  </head>  <body>  <p>A simple large font sentence.</p>  </body>  </html> |

| Question 3. In the space provided below, rewrite the test.html document created in the previous activity so that it removes the embedded style sheet and implements an inline style instead: |
| --- |
| *<!DOCTYPE html>*  *<html>*  *<head>*  *<meta charset="utf-8" />*  *<title>A test web page</title>*  *<!-- <link rel="stylesheet" href="main.css"> -->*  *<!-- <style>*  *p {*  *color: red;*  *font-style: italic;*  *}*  *</style> -->*  *</head>*  *<body>*  *<p style="color: red; font-style: italic">A simple large font sentence.</p>*  *</body>*  *</html>* |

| Question 4. Create a folder called question4. Inside that folder, create an HTML document that contains three different headers: h1, h2, and h3. Each should contain header text specifying what element it is. For example, h1 should specify “This is a header 1 element”.  Link an external style sheet that sets the colour of each of these elements as follows:   * h1: indigo * h2: maroon * h3: orangered, opacity .5 The output should appear as follows: |
| --- |
|  |

| Question 5. Comment out the CSS developed in the previous activity and replace the colour keywords with hexadecimal values as follows:   * h1: #4B0082 * h2: #800000 * h3: #FF4500, opacity .5   The output should appear the same as per the previous activity when the HTML markup is tested as follows: |
| --- |
| Text  Description automatically generated with medium confidence |

| Question 6. Comment out the CSS developed in the previous activity and replace the hexadecimal values with rgb functions. Search online for the rgb functions equivalent of the required colors.  The output should appear the same as per the previous activity when the HTML markup is tested as follows: |
| --- |
| Text  Description automatically generated with medium confidence |

| Question 7. Create a folder called question7. Inside that folder, create an HTML document and link it to an external stylesheet. Create three paragraph elements within the HTML document with the following text:   * p: This is paragraph 1 * p: This is paragraph 2 * p: This is paragraph 3   Give each paragraph element a different class name: “class1”, “class2”, and “class3”. Then in the external CSS, assign the following fonts using the font-family property to each of the classes you have create:   * class1: Arial * class2: Palatino Linotype, Times New Roman * class3: Impact, serif   The output should appear similar to the following (depending on what fonts are on your device): |
| --- |
|  |

| Question 8. Create a folder called question8. Inside that folder, Create three paragraph elements within the HTML document with the following text:   * p: This is a paragraph * p: This is another paragraph * p: This is the last paragraph   Give each paragraph element a different class name: “class1”, “class2”, and “class3”. Then in the external CSS, complete the following:   * class1: ensure the font is Time New Roman, that the colour is “darkslateblue”, and that the font size is 75% of the default font size * class2: ensure the font is Tahoma, that the colour is “goldenrod”, and that the font size is slightly smaller than the default font size * class3: ensure the font is Garamond, that the colour is “seagreen”, and that the font size is xx-large   The output should appear similar to the following: |
| --- |
|  |

| Question 9. Rewrite the following css rule in the space provided below so that only the font property is used:  p {  font-family: Calibri; font-size: 150%; font-style: oblique; font-weight: 300; font-variant: normal;  font-stretch: semi-condensed;  } |
| --- |
| *p {*  *font: oblique, normal, 300, semi-condensed, 150%, Times;*  *}* |

| Question 10. Create a folder called question10. Inside that folder, Create (or modify) an HTML document and link it to an external stylesheet. Create three paragraph elements within the HTML document, each with the following text: “This is a sample paragraph”. Each should have a different class name: “class1”, “class2” and “class3”.  Then code the CSS classes for each so that the output is similar to the following: |
| --- |
|  |

| Question 11. *Using the CSS box model*  Create a folder called question11. Inside that folder, create an HTML document and link it to an external stylesheet. Create three div elements within the HTML document, each with the following text: “This is a div”. Each should have a different class name: “class1”, “class2” and “class3”.  Then code the CSS classes for each so that the output is similar to the following: |
| --- |
|  |

| Question 12. Create a folder called question12. Inside that folder, create an HTML document and link it to an external stylesheet. Add 2 div elements that each have the class names of “class1”, and “class2” respectively.  In the CSS code, create two classes that each link to one of the div elements. Then complete the following:   * class1: give it a background colour of blue and set its height to be a 100px, and its width to be 50% of the viewports width * class2: give it a background colour of red and set its width to be a 100px, and its height to be 25% of the viewports height   The output should appear similar to the following: |
| --- |
|  |

| Question 13. Create a folder called question13. Inside that folder, create an HTML document and link it to an external stylesheet. Create three div elements within the HTML document, each with the following text: “This is a div”. Each should have a different class name: “class1”, “class2” and “class3”.  Then code the CSS classes for each so that:   * class1: has an “aliceblue” background colour and a width of 500px * class2: has a “lightgray” background colour, a width of 50% of the screens width, and a height of 200px   class2: has a “lightgreen” background colour, and a maximum width of 400px.  The output should be similar to the following: |
| --- |
|  |

| Question 14. Create a folder called question13. Inside that folder, create an HTML document and link it to an external stylesheet. Add a p element to the document, and then nest an img element within the p element. Set the img element to have an image of your choice, and then add three or four dummy paragraphs to the p element directly after the nested img element such as lorem ipsum text.  Create a CSS rule for the img and p elements, and practice using the float and clear properties until you are happy with their usage.  An example is shown below where the paragraph text has been floated to the left and an image floated to the right: |
| --- |
|  |

| Question 15. *Working with text and fonts*  Research and practice how to use the following CSS properties   * text-indent * word-wrap * word-spacing * tab-size * font-weight * font-style   Write your answers in the space provided below: |
| --- |
| * text-indent: The text-indent property specifies the indentation of the first line in a text-block. Negative values are allowed. The first line will be indented to the left if the value is negative. CSS syntax text-indent: length|initial|inherit; * word-wrap: The word-wrap property allows long words to be able to be broken and wrap onto the next line. CSS syntax word-wrap: normal|break-word|initial|inherit; * word-spacing: The word-spacing property increases or decreases the white space between words. CSS syntax word-spacing: normal|length|initial|inherit; * tab-size: The tab-size property specifies the width of a tab character.In HTML, the tab character is usually displayed as a single space-character, except for some elements, like <textarea> and <pre>, and the result of the tab-size property will only be visible for those elements. CSS syntax tab-size: number|length|initial|inherit; * font-weight: The font-weight property sets how thick or thin characters in text should be displayed. CSS syntax font-weight: normal|bold|bolder|lighter|number|initial|inherit; * font-style: The font-style property specifies the font style for a text. CSS syntax font-style: normal|italic|oblique|initial|inherit; |