OCR Computer Science A Level

1.3.4 Web Technologies

Advanced Notes

Specification

1.3.4 a)

- HTML
- CSS
- JavaScript

1.3.4 b)

- Search engine indexing

1.3.4 c)

- PageRank algorithm

1.3.4 d)

- Server and Client side processing

Web Development

HTML

HTML is the language/script that web pages are written in. HTML allows a browser to interpret and render a webpage for the viewer by describing the structure and order of the webpage. The language uses tags written in angle brackets (<tag>, </tag>). There are two sections of a webpage: the body and the head. The head contains the title of the webpage and the body contains the content of the webpage.

HTML Tags

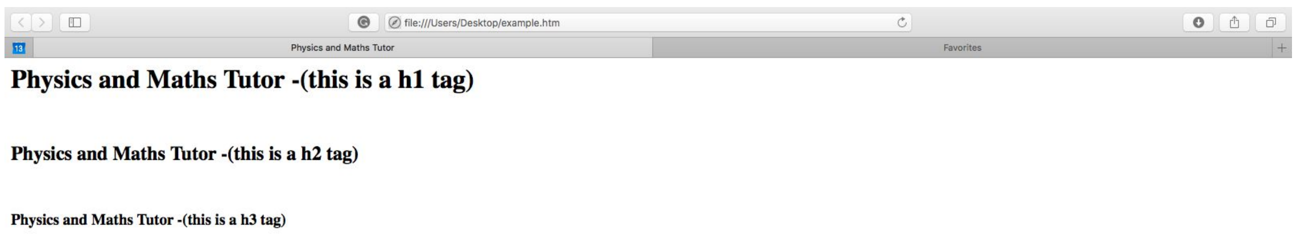
<html>	All code written within these tags is interpreted as HTML
<body>	Defines the content in the main content area of the webpage
<link>	Used to link to additional files, including CSS stylesheets
<head>	Defines the browser tab or window heading area
<title>	Defines the text that appears with the tab or window heading area
<h1>, <h2>, <h3>	Heading styles in decreasing sizes

For example, the HTML below:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
</head>

<body>
     <h1> Physics and Maths Tutor - (this is a h1 tag) </h1>
     <h2> Physics and Maths Tutor - (this is a h2 tag) </h2>
     <h3> Physics and Maths Tutor - (this is a h3 tag) </h3>
</body>
</html>
```

… produces a webpage like this:



Physics and Maths Tutor -(this is a h1 tag)

Physics and Maths Tutor -(this is a h2 tag)

Physics and Maths Tutor -(this is a h3 tag)

Paragraphs

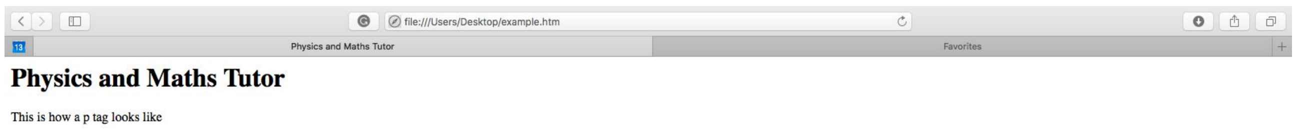
<p> is used to define a paragraph separated with a line space above and below.

The code below:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
</head>

<body>
     <h1> Physics and Maths Tutor </h1>
     <p> This is how a p tag looks like </p>
</body>
</html>
```

Produces the following web page:



Physics and Maths Tutor

This is how a p tag looks like

Images

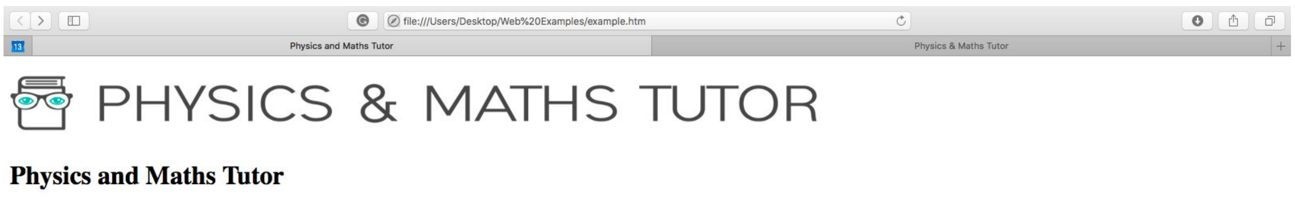
<img> is used for images. This is a self-closing tag, which means that there is no need to include a closing tag (`</img>`) when using it. The following tag parameters must also be included when using this tag: `src` (source), `height=x`, `width=y`)

For example:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
</head>

<body>
     <img src="Header-Physics-Maths-Tutor.png" width="1000"
     height="100">
     <h1>Physics and Maths Tutor</h1>
</body>
</html>
```

Gives:



Physics and Maths Tutor

Link

<a> is an anchor tag that defines a hyperlink with the location parameter, and is laid out in the following form:

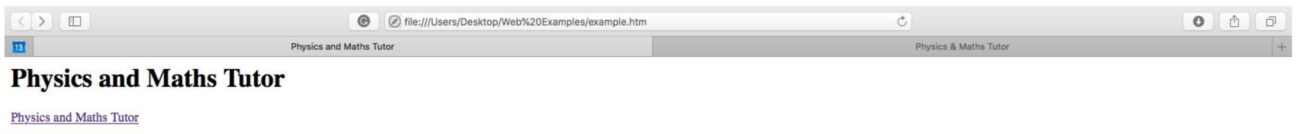
`<a href= location>` **`link text`** `</a>`

For example:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
</head>

<body>
     <h1> Physics and Maths Tutor </h1>
     <a href="https://www.physicsandmathstutor.com"> Physics and
     Maths Tutor </a>
</body>
</html>
```

Produces:



Physics and Maths Tutor

Physics and Maths Tutor

Ordered list

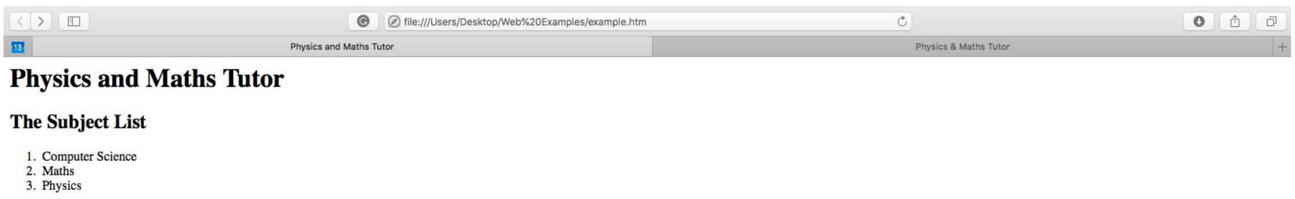
<ol> is used to define an ordered list. Each element within the list is defined using the <li> tag.

For example:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
</head>

<body>
     <h1> Physics and Maths Tutor </h1>
     <ol>
          <li> Computer Science </li>
          <li> Maths </li>
          <li> Physics </li>
     </ol>
</body>
</html>
```

Gives:



file:///Users/Desktop/Web%20Examples/example.htm

Physics and Maths Tutor | Physics & Maths Tutor

Physics and Maths Tutor

The Subject List

1. Computer Science
2. Maths
3. Physics

Unordered list

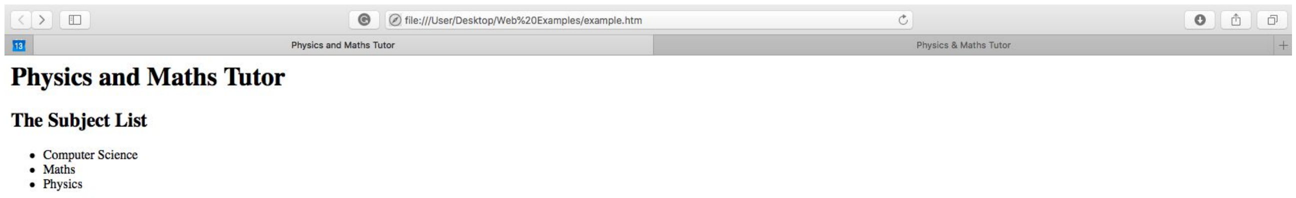
<ul> defines an unordered list. Again, each element within the list is defined using the <li> tag.

For example:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
</head>

<body>
     <h1> Physics and Maths Tutor </h1>
     <ul>
          <li> Computer Science </li>
          <li> Maths </li>
          <li> Physics </li>
     </ul>
</body>
</html>
```

Gives:



Physics and Maths Tutor

The Subject List

- Computer Science
- Maths
- Physics

Division

<div> divides a page into separate areas, each which can be referred to uniquely by name. The following syntax is used when creating a division with the name page:

```
<div id= “page”>
```

Classes and Identifiers

Classes and identifiers are attributes given to elements on a webpage which you wish to style in a particular way.

Multiple elements across web pages can be assigned to a certain class. This means elements can follow a consistent style, and the styling/ formatting only has to be defined once. This can be defined within the head of a web page, or within a linked CSS style sheet. Classes are defined using a full stop, as shown below:

```
.example {                                  // defining
  background-color: green;                     a class
  border: 1px solid black;
}
                                            // using a class
<div class = ‘example’>
     <p> This paragraph will have a green background and a black
     border </p>
</div>
```

Identifiers are a unique name given to an element on a web page. Whereas a class name can be used by multiple elements, only one element can be associated with a particular identifier. They are defined using a hashtag, as shown below:

```
#example {                                  // defining
  background-color: green;                     a class
  border: 1px solid black;
}
                                            // using a class
<div class = ‘example’>
     <p> This paragraph will have a green background and a black
     border </p>
</div>
```

CSS

Cascading style sheets (CSS) is a language which is used to describe the style of a webpage. CSS can be used to specify the way HTML elements look and can be applied to tags such as <h1>, <p> and <div>.

CSS can be used in two different forms: internal/embedded CSS or external CSS. Internal/embedded CSS is placed inside the style tags and is entered directly into the HTML document. Meanwhile external CSS is written in a separate document and a link to this style sheet is added to the HTML document.

Whenever an external style sheet is used, the following link is added to the header:

```
<link href= “nameofstylesheet.css” rel= “stylesheet” type=
“text/css”>
```

CSS has a particular syntax that must be followed, as shown below:

```
body
{
     margin: 0px;
     padding: 0px;
     background-color: white;
     font-family: Arial, Helvetica, sans-serif;
     font-size: 18px;
     Text-align: center;
}
```

Each section of CSS begins with the name of the class/identifier or element to be styled, followed by a set of curly brackets within which the attributes of the element are defined.

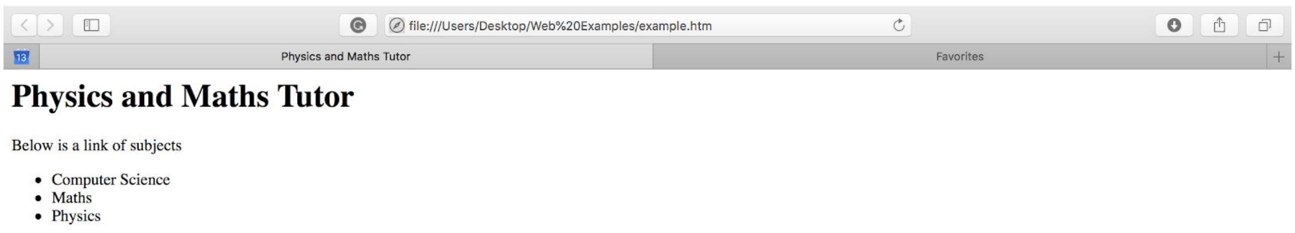
Below is an example of a piece of HTML which uses the above CSS:

```
<html>
<head>
     <title> Physics and Maths Tutor </title>
     <link href= "styles.css" rel="stylesheet" type="text/css">
</head>

<body>
     <h1> Physics and Maths Tutor </h1>
     <p> Below is a link of subjects </p>
```

```
    <ul>
        <li> Computer Science </li>
        <li> Maths </li>
        <li> Physics </li>
    </ul>
</body>
</html>
```

Below is what the web page looks like:
1. Without the CSS linked to it

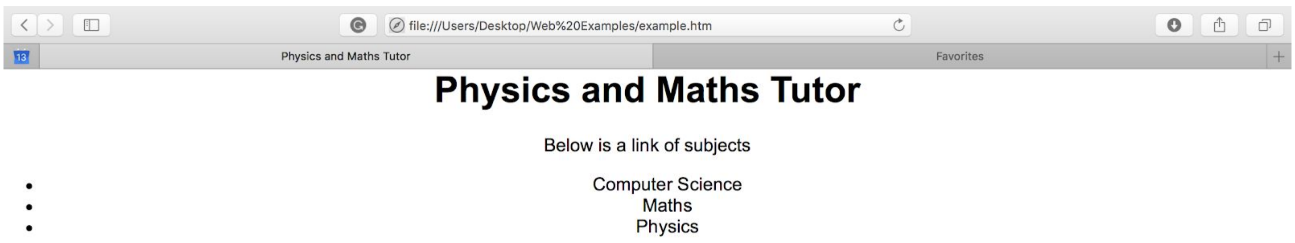


Physics and Maths Tutor

Below is a link of subjects

- Computer Science
- Maths
- Physics

2. With the CSS linked to it



Physics and Maths Tutor

Below is a link of subjects

- Computer Science
- Maths
- Physics

<u>JavaScript</u>

JavaScript is a scripting language which primarily exists to add interactivity to websites. JavaScript is interpreted rather than compiled. This means web pages are interpreted by the browser each time they are loaded, independently of the architecture of the processor which it is being loaded by. Javascript is often used to validate input data on the client computer.

Synoptic Link

JavaScript is a multi-paradigm language.

Paradigms are covered in **1.2.4** under **Programming Paradigms**.

Advantages of using JavaScript:

- Local computer can deal with invalid data before it is sent off to the servers,
- Eases the load on busy servers
- Reduces web traffic

Inputs from HTML forms can be retrieved from a webpage and handled using JavaScript. Below are some uses of JavaScript you should be aware of:

Outputs

Changing the attributes of a HTML element:

```
chosenElement = document.getElementById(“example”);
chosenElement.innerHTML = “Hello World”;
```

Writing directly to the document:

```
document.write(“Hello World”);
```

Displaying an alert box:

```
alert(“Hello World”);
```

Search Engine Indexing

Search Engines

A search engine is a program that searches through a database of internet addresses looking for resources based on criteria set by the client.

The order in which web resources are displayed is very important as it determines which websites users choose to visit and use. Therefore it is important that search engines display the most high-quality websites with relevant content at the top of the page.

Search engines rely on an index of web pages. Web crawlers, occasionally termed spiders, are used to collect information about websites to build this index. Web crawlers work by traversing the Internet, web page by web page using links on websites. The web crawlers collect keywords and phrases from the linked web pages and add this information to the index. They also collect and add meta data from websites, which is the information specified by the website owner.

PageRank Algorithm

The page rank algorithm ranks web pages, determining the order in which web pages are displayed when a search is conducted. Higher ranked pages will show up first.
There are two factors which determine the page rank of a page:

- How many incoming links it has from other web pages
- The page rank of the web pages that link to it

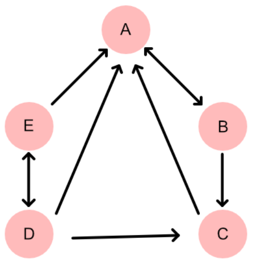
Synoptic Link

Graphs can be used as **visual representations** of **complex relationships**.

Graphs are covered in **1.4.2** under **Data Structures and Graphs**

The data structure used to display this information is a directed graph. This shows which pages link to other websites, and webpages are represented as nodes while links between two pages are represented as arcs between the nodes.
The pagerank algorithm is as follows:
PageRank(x) = (1-d) + d[(PageRank(T1) ÷ Count(T1)) + … + (PageRank(Tn) ÷ Count(Tn)]

where *PageRank(x)* is the page rank of page *x*, *Count(Tn)* is the total count of outbound links from a webpage *n* and *d* is the damping factor. This is usually set to 0.85.
Below is an example in which we calculate the page rank of C.
The first iteration assumes a page rank of 1 for the values that we do not know:



PR(A) = (1-d) + d(PR(B)÷2 + PR(C)÷1 + PR(D)÷3 + PR(E)÷2)	PR(A) = 1.878
PR(B) = (1-d) + d(PR(A)÷1)	PR(B) = 1.746
PR(C) = (1-d) + d(PR(B)÷2 + PR(D)÷3)	PR(C) = 1.175
PR(D) = (1-d) + d(PR(E)÷2)	PR(D) = 0.575
PR(E) = (1-d) + d(PR(D)÷3)	PR(E) = 0.313

The second iteration takes in the values from the first iteration:

PR(A) = 0.15 + 0.85(1.746÷2 + 1.175÷1 + 0.575÷3 + 0.313÷2) = 2.187
PR(B) = 0.15 + 0.85(2.187) = 2.001
PR(C) = 0.15 + 0.85(2.001÷2 + 0.575÷3) = 1.163
PR(D) = 0.15 + 0.85(0.313÷2) = 0.283
PR(E) = 0.15 + 0.85(0.283÷3) = 0.230

The third iteration takes in the values from the second iteration:

PR(A) = 0.15 + 0.85(2.001÷2 +1.163÷1 + 0.283÷3 + 0.230÷2) = 2.321
PR(B) = 0.15 + 0.85(2.321) = 2.122
PR(C) = 0.15 + 0.85(2.001÷2 + 0.283÷3) = 1.081
PR(D) = 0.15 + 0.85(0.230÷2) = 0.248
PR(E) = 0.15 + 0.85(0.248÷3) = 0.220

It is important to remember that in reality, modern search engines use a much more complex algorithm to determine the order in which websites are displayed. These take into account various other factors, such as the age of a web page and how quickly content on the web page loads.

Server and Client Side Processing

Server side Processing

Server side processing is when a client sends data to a server for it to be processed. This means no information is processed on the client computer. Common server side scripting languages are SQL or PHP. Server side processing is useful for several reasons:

- Does not require plugins
- Can perform large calculations much faster than clients
- Not browser dependent,
- More secure

Client side Processing

Client side processing is when a client processes the data on a local device. This means that all of the information is processed on the client computer. This is also called client side scripting, and uses languages such as JavaScript. This is useful for the following reasons:

- Webpage can immediately respond to user actions
- Executes quickly
- Gives developers more control over the behaviour and look of the website