

OCR Computer Science AS Level

1.3.3 Web Technologies

Advanced Notes



Specification

1.3.3 a)

- HTML
- CSS
- JavaScript

1.3.3 b)

- Lossy Compression
- Lossless Compression



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

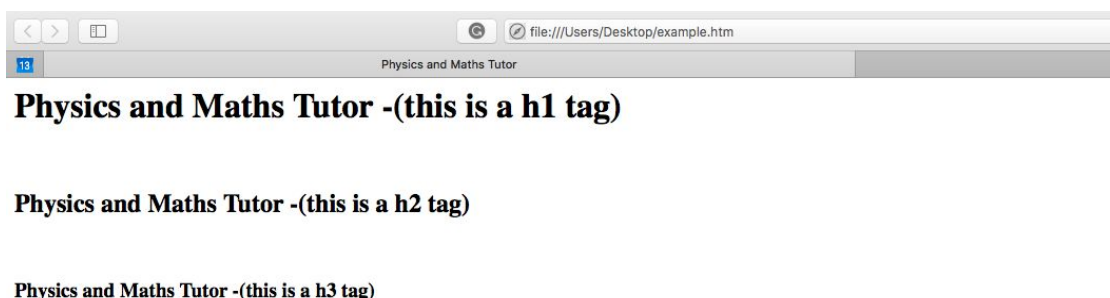
- `<html>` : All code written within these tags is interpreted as HTML
- `<body>` : Defines the content in the main browser content area
- `<link>` : Used to link to additional files, like CSS stylesheets
- `<head>` : Defines the browser tab or window heading area
- `<title>` : Defines the text that 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 an h1 tag) </h1>
  <h2> Physics and Maths Tutor - (this is an h2 tag) </h2>
  <h3> Physics and Maths Tutor - (this is an h3 tag) </h3>
</body>
</html>
```

... results in a webpage like this:



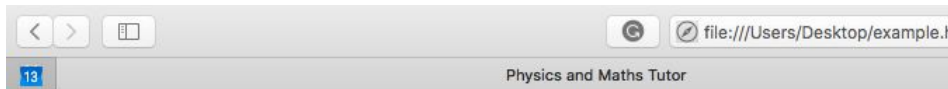
- `<p>` : A paragraph separated with a line space above and below

For example:

```
<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>
```

Gives:



Physics and Maths Tutor

This is how a p tag looks like



- `` : A self-closing tag (there's no need to include ``) used for images. Tag parameters include (src (source), height=x, width = y)

For example:

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

<body>
  
  <h1>Physics and Maths Tutor</h1>
</body>
</html>
```

Gives:



PHYSICS & MATHS TUTOR

Physics and Maths Tutor



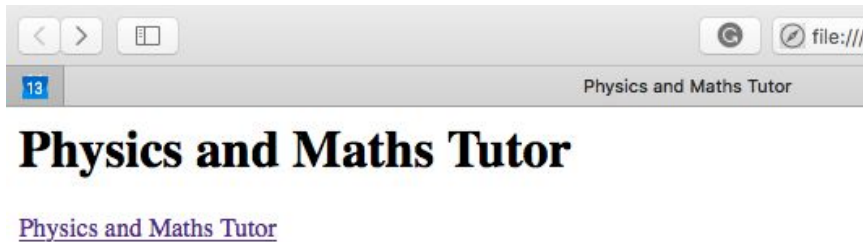
- `<a>` : Anchor tag defining a hyperlink with location parameters (`link text`)

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>
```

Gives:



- `` : Defines an ordered list,

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:



Physics and Maths Tutor

The Subject List

1. Computer Science
2. Maths
3. Physics



- `` : Defines an unordered list,

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

- `` : Defines an individual list item
- `<div>` : Creates a division of a page into separate areas each which can be referred to uniquely by name, (`<div id= "page">`)



Classes and Identifiers

Class and identifier selectors are the names which you style, this means groups of items can be styled, the selectors for html are usually the div tags.

Identifiers are defined with a hashtag #header. Identifiers must be **unique** to each webpage

Classes work in a similar way but use a full stop as a prefix to the class name e.g. .list . classes can be used multiple times on a webpage.

CSS

CSS is a **script / language** like HTML, except that it is used to **describe the style** of a webpage. CSS styles can be used to specify the way HTML elements look, they can be applied to whole tags such as <h1>, <p> or <div>.

CSS can be used using two different forms, internal (embedded) or external. The **internal/embedded CSS** is placed inside the style tags and is **entered directly** within the HTML document. The external CSS is placed inside an external style sheet. A link is created in an external sheet which can be accessed from the HTML.

Whenever you have an external style sheet you add this to the header. Just replace styles.css with the name of the stylesheet.

```
<link href= "styles.css" rel= "stylesheet" type= "text/css">
```

CSS has a format you have to follow, below is an example shown below:

```
@charset "utf-8";
```

```
Body
```

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



Here is an example of HTML code which uses the above css code:

```
<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>
```

From the above two code here is what the webpage looks like:

1 - Without the CSS code

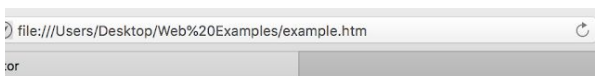


Physics and Maths Tutor

Below is a link of subjects

- Computer Science
- Maths
- Physics

2 - With the CSS code



Physics and Maths Tutor

Below is a link of subjects

Computer Science
Maths
Physics



JavaScript

JavaScript is a language which has a similar layout to languages like [python](#). The main function of JavaScript is to add [interactivity](#) to websites. JavaScript isn't [compiled](#), instead it is [interpreted](#), this is so it can be interpreted in the browser every time the webpage is displayed, compiled code is dependant on the chips used whereas interpreted code isn't.

Synoptic Link

JavaScript is a multi-paradigm language.

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

Javascript be used to [input data](#) on the [client's computer](#), this may change the local page interactively or post data to a server. The advantages:

- The local computer can fix invalid data before sending it off to the servers
- It can ease the traffic off of the busy servers

You can take in inputs from forms on a webpage, you don't need to memorise doing this however, you should understand what to do once you have this input. Below are the examples given by the exam board.

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");
```

You can use an alert box:

```
alert("Hello World");
```

Javascript uses a similar layout of programming to what you're used to. It relies on a lot on [functions](#), a common mistake made is thinking you have to enter parameters, you don't instead you use the identifiers. Functions are initiated using the following structure:

```
function functionname() {
    // enter the code in here
}
```



Lossy vs Lossless Compression

There are two categories of compression: **lossy** and **lossless**. As the name suggests, lossy compression **reduces the size of a file** while also **removing some of its information**. This could result in a **more pixelated** image or **less clear** audio recording. On the other hand, lossless compression **reduces the size** of a file **without losing any information**.

When using **lossless compression**, the original file **can be recovered** from the compressed version. Something which is **not possible** when using lossy compression which reduces the size of the file by **completely disregarding** some information. For example, audio files can be compressed lossily by removing the very high or very low frequencies which are least noticeable to the ear. There's no way to go from the lossy version of the recording back to the full version as there's no record of what the high and low frequencies were.

