

Task 1

Explain the web architecture and the components needed for a client request to view a page using the image below. Use Task 1 PPT and links provided to help you

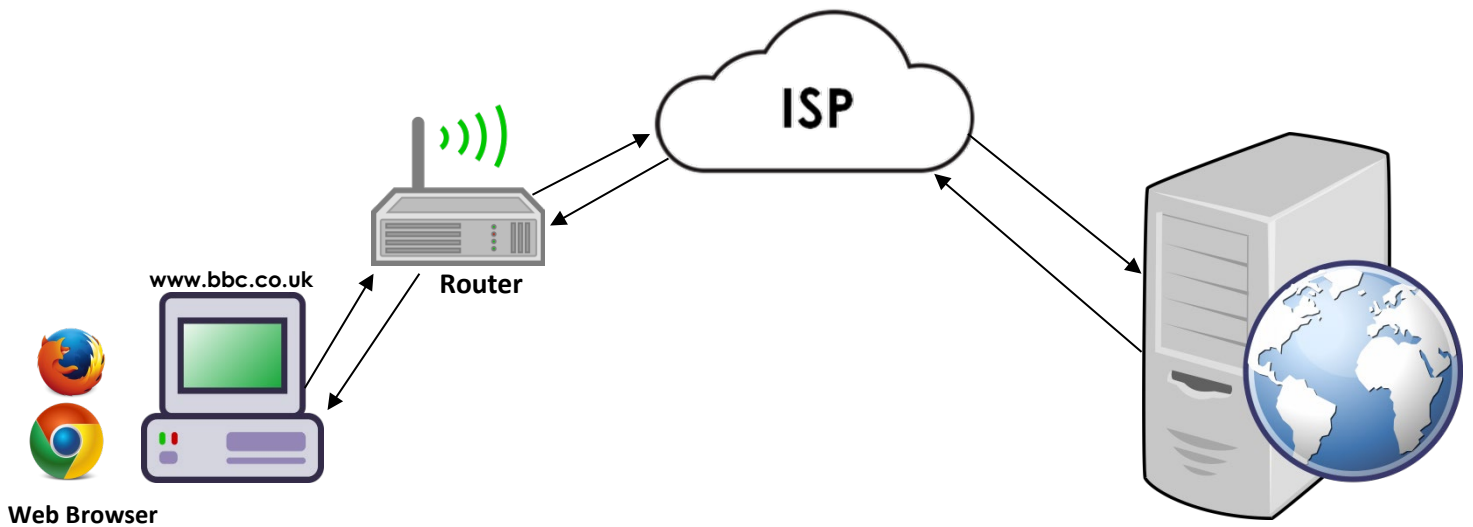


Fig 1.1 Web page request

Web Architecture
Internet service provider
Web hosting services
Domain structure
Domain name registrars

Components
Routers
Internet browser
Web server/Proxy Server/Email server

Note: You must explain the diagram using the keywords in the boxes above

See how websites work: <https://www.youtube.com/watch?v=D8c4JZW73cM>

See how DNS Server works:

<https://www.youtube.com/watch?v=mpQZVYPuDGU>

Task 2 Research Task- Complete the table below

Term	Definition
Protocol	
HTTP Hypertext Transfer Protocol	
<i>HTTPS</i> <i>Secure Hypertext Transfer Protocol</i>	
FTP File Transfer Protocol	
IP Internet Protocol	
TCP Transmission Control Protocol	
SMTP Simple Mail Transfer Protocol	
POP Post Office Protocol	

Task 3

Explain the TCP\IP protocol

Explain each layer within TCP\IP (use illustrative examples)

Task 4- Website Development

NOTE: Need to read through Website Guide (Pages 5-12)

You have been hired by to create a website on a topic of your choosing.

He has specified that your website must be created using notepad. It must also include:

- At least two pages
- A consistent theme applied using CSS
- Headings
- Main text
 - Some italicised
 - Some bolded
- Dividers in the HTML code
- A form
- A text box with user input
- Both a bullet pointed list and a numbered list
- Hyperlinks
- Images
- An interactive element

Will need to provide of the website through Screens of

1. HTML files
2. CSS File
3. Website working in browser

Task 5 Answer the following Questions

1. What does HTML stand for?
2. What does CSS stand for?
3. Why do we create CSS as a separate notepad file?
4. What are the HTML tags for the following functions?
 - a. Links to CSS files
 - b. Headings
 - c. Paragraphs
 - d. Dividers
 - e. Images
 - f. Hyperlinks
 - g. Forms
 - h. User input
 - i. Bullet pointed list
 - j. Numbered list
5. What language do we use to add interactivity to websites?

The basics of HTML

Web pages are made from commands called tags that describe how the page should be displayed in the browser. Each tag is enclosed in chevrons: <>. There is often an opening tag, the start of the command and a closing tag, the end of the command. The end of the command is signified by: /

Convention states that tags should be in lowercase. String identifiers can use single or double quotes.

Web page structure

All web pages have the basic structure of:

```
<html>      Indicates the language used: the page is hypertext mark-up language.
<head>      Data about the page is placed in the head section. This is known as meta-data.
</head>
<body>      The content of the page is placed in the body section.
</body>     Note how each section is ended with a corresponding closing tag.
</html>
```

Title bar

To give the web page a title in the bar at the top of the browser, use a TITLE tag in the head section:

```
<title>Put the title you want to appear in the title bar here</title>
```

Headings

To put a heading on the page, use h tags in the body section:

```
<h1>A top level heading</h1>
<h2>A sub heading</h2>
<h3>A sub-sub heading</h3>
```

Fonts, colours and formatting of text

This can be done using in-page styles with the style attribute inside the opening tag.

For example, if you wanted to make a heading blue, you could use the style attribute like this:

```
<h1 style="color:blue;">This is a title in blue</h1>
```

Using paragraphs for text

Text is usually enclosed in paragraph tags: <p> and </p>. Each paragraph automatically has a blank line between each one, which is the standard way of presenting paragraphs on a web page.

Images

Images must be saved in a separate file in GIF, JPEG or PNG format. They are then linked into the page using the tag. There is no closing tag.

```

```

There are additional attributes for the image tag to specify alternative text for the blind, height and width attributes. E.g.

```

```

Hyperlinks

A link to another page requires an anchor tag: <a> and a hyperlink reference attribute.

```
<a href="the filename of the page to link to goes here">The words the user actually clicks here</a>
```

Bullet points (unordered lists)

Bullet lists require two tags. One to signify the start and end of unordered list , and one for each list item . It is good practice to indent list items.

```
<ul>
  <li>Bullet point one</li>
  <li>Bullet point two</li>
  <li>Bullet point three</li>
</ul>
```

Numbered points (ordered lists)

Numbered lists require two tags. One to signify the start and end of the ordered list , and one for each list item . It is good practice to indent list items.

```
<ol>
  <li>Point one</li>
  <li>Point two</li>
  <li>Point three</li>
</ol>
```

Other tags

There are many other tags, but these are the basic ones you need to know for exams. In addition, for other concepts you need to understand you will need to know:

- `<div>` Explained in the Basics of CSS guide
- `<form>` Explained in the Basics of HTML forms guide
- `<input>` Explained in the Basics of HTML forms guide
- `<script>` Explained in the Basics of Javascript guide

No other knowledge of HTML is necessary.

The basics of CSS

Cascading style sheets (CSS) allow you to separate the content of a web page from how it is presented. This is useful if you want to be able to change how a page is printed, compared to how it is viewed on-screen. Or to allow it to be more accessible to people with disabilities: allowing them to choose a high contrast style in their browser for example. By using CSS there are many more creative things you can do to the content that are not possible with HTML alone.

1. Create a new text file called stylesheet.css
2. In the file, enter:

```
h1 {  
    font-family: Verdana;  
    font-size: medium;  
    color: #039;  
    background-color: #99CCFF;  
    border-width: 1px;  
    border-style: solid;  
    border-color: #039;  
    width: 50%;  
    height: 30px;  
}
```

3. Save the file as: stylesheet.css
(Make sure that no additional extensions are added, for example, Notepad may add .txt)
4. Now you need to link the stylesheet to the web page.
Inside the <head> section of a html page, add a link reference:

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

Note the effect that these styles have on the heading on the web page.

Applying styles to other elements

If you want to style other sections of the page, such as a block of text with an image, you need to create a class or identifier in the CSS file. Identifiers are used for items that only exist once on a page, such as a navigation bar. Styles are used where you want to apply the mark-up to more than one aspect of a page. In reality, it doesn't really matter if you use styles for everything.

Classes are defined with a dot like this:

```
.content {  
    font-family: Verdana;  
}
```

Identifiers are defined with the hash symbol like this:

```
#menu {  
    background-color: #99CCFF;  
}
```

Whether you use classes or identifiers, you need to specify in the HTML page which elements this style applies to using the <div> tag and the class attribute or the id attribute. E.g.

```
<div class="content">  
    This is the content you want to style  
</div>
```

Note, the dot only exists in the CSS file, not in the class attribute.

The basics of HTML forms

Form controls are put on a web page using a <form> tag wrapper.

Simple text boxes are defined with the <input> tag.

Form structure

A basic form with one text box and a submit button would be created with this HTML:

```
<form id="form1" action="">  
  Forename:  
  <input type="text" name="forename" value="forename">  
  <input type="submit" name="submit" value="Submit">  
</form>
```

You need to know about text input and a submit button for exams, although other input options are achieved in a similar way with alternative tags.

The basics of JavaScript

It is possible to put client-side programming elements on a web-page using JavaScript.

HTML and JavaScript example

Enter this basic HTML form into the body section of a page:

```
<form id="form1" action="">
  Forename:
  <input type="text" name="forename">
  <input name="Submit" type="submit" id="Submit" value="This submits the
data to another page">
</form>
```

A button outside the form is used to trigger the Javascript code to execute.

Enter this HTML:

```
<button onclick="myFunction()">Try Javascript</button>
```

An output identifier is used to capture the output from the code.

Enter this HTML:

```
<p id="output_here"></p>
```

Enter the JavaScript overleaf.

Study the example to see how:

- Javascript elements have a <script> tag.
- Command lines are terminated with a semicolon.
- Curly brackets are used to define sections of code.
- Functions are defined.
- Variables are defined.
- For loops work.
- An array of inputs can be referred to.
- Variables are assigned values.
- An If condition can be used for selection.
- How the output is returned to the page.

```
<script>
function myFunction() {
    var x = document.getElementById("form1");
    var text = "";
    var i;
    for (i = 0; i < x.length ;i++) {
        data = x.elements[i].value;
        if (data!="This submits the data to another page") {
            text += "Your name was entered as " + data + ".";
        }
    }
    document.getElementById("output_here").innerHTML = text;
}
</script>
```

Additional output options

The above example changes the contents of an existing HTML element called "output here".

You can also write directly to the document using:

```
document.write("Hello World")
```

You can also use alert pop-up boxes:

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