

Tiverton High School Year 7 Computing Spring/Summer Terms Knowledge Organiser

Creating Web-Pages (Unit 7-5)

Web-pages can be **displayed** or **viewed** in a program called a **web-browser**.



Web-pages can be **created** using a special language called **HTML** (Hyper Text Markup Language).

When creating a web-page, a person adds special codes called "HTML **tags**" into their document. The tags tell the web-browser exactly **how** to display parts of the document.

A web-page is made up of 2 separate parts: the **HEAD** and the **BODY**.

The **HEAD** section contains **important settings** about the web-page that you cannot actually see in the main browser window.

The **BODY** part contains **all the information that you will be able to see when the web-browser loads the web-page**.

Some tags can also contain extra useful information for the browser to use. These details are placed **inside** the tag. They are called **attributes**. A good example is the **IMG** tag. When typing an **IMG** tag into your web-page, you can add extra information to set the **width** and the **height** of the picture so the web-browser displays it correctly.

HTML Tags for Creating Web-Pages (Unit 7-5)

`<HTML> </HTML>` makes a web-page that can be displayed in a web-browser program.

`<HEAD> </HEAD>` makes the **head** section of the web-page. This holds important **settings** for the web-page.

`<BODY> </BODY>` makes the **body** of the web-page. Anything inside the body section will be **displayed** in the web-browser.

`<TITLE> </TITLE>` must be used **inside** the head section. This sets the **title** (or **name**) of the browser window.

`<H1>Exclusive!</H1>` makes a large **heading/headline**.

`` inserts a **picture** into the page, setting out how large it should be.

`Click here` creates a **hyperlink** so a person can **jump** to another web-page.

Here is an example of a very simple web-page made using HTML:

```
<HTML>
  <HEAD>
    <TITLE>Web-Page!</TITLE>
  </HEAD>
  <BODY>
    <H1>Look!</H1>
    This is a simple web-page that can be displayed in a browser.
  </BODY>
</HTML>
```

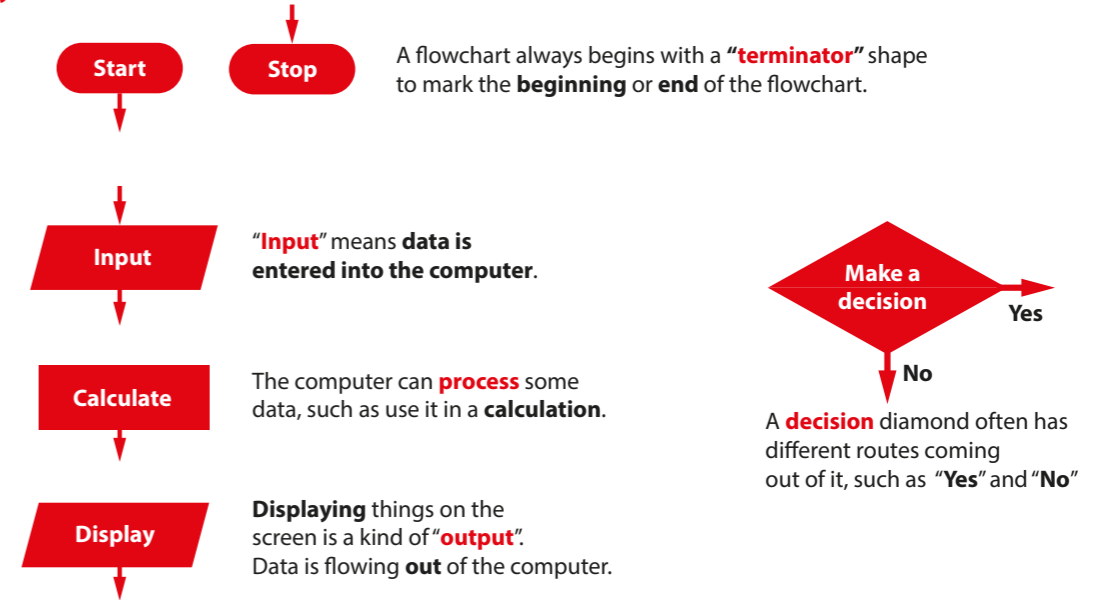
Problem Solving and Planning (Unit 7-6)

A **program** is a sequence of **instructions** that the computer will carry out (**execute**).

An **algorithm** is a precise set of written steps that describe exactly **how to solve a problem**.

A **flowchart** is a **diagram** that shows how an algorithm works.

Flowchart Symbols



Creating and Testing Programs (Unit 7-8)

You can create software by writing new programs. You write the program instructions using a **programming language**.

Input means gathering some data from the keyboard or other input device and storing it in a **variable**:

```
INPUT width
```

Output means displaying something on the screen:

```
PRINT "Your final score is"
PRINT score
```

A **sequence** is a group of program statements that are executed in the correct order, one after the other.

A **variable** is a **named value** that can **change** while your program is running e.g. **score**

Assignment means giving a value to a variable

```
x = 3          password = "Cu5tArd"
```

Iteration means repeatedly executing parts of the program again and again (looping):

```
FOR time = 1 TO 10      WHILE time < 60
```

Selection means making a decision to select which part of the program code should be executed:

```
IF lives > 0 THEN
  PRINT "Lost a life"
ELSE
  PRINT "Game Over"
ENDIF
```

Arithmetic operators

```
+   Addition
-   Subtraction
*   Multiplication
/   Division
```

Relational Operator Symbols when making comparisons

<	less than	>	greater than
<=	less than or equal to	>=	greater than or equal to
==	is the same as	!=	not the same as