

# HTML

Hypertext Markup Language



# HTML

HTML is the language we have used to create webpages since the first webpages arrived on the web.

HTML uses tags to format web content, such as `<b>this is bold</b>`, which would cause the text within the starting b tag and ending b tag to render as bolded text.

`<b>Bold</b>` renders: **Bold**

starting tag                      ending tag

Many but not all HTML tags have a matching end tag. HTML tags such as `<br>` and `<img>` did not have ending tags because they just render a specific object (e.g. a line break; an image).

HTML code is interpreted by **browsers**. Browsers are designed to be HTML backward compatible and forward compatible. Hence, browsers just ignore tags that they don't recognise. For example, if a browser came across a `<xyz>` tag that it didn't recognise, it would skip over the tag as though it didn't exist.

# HTML5 introduces semantic mark-up

Semantic is defined as the study of the meaning of linguistic expressions.

HTML5 stresses separating structure, presentation, and behaviour; a good practice to adhere to. Before HTML5, that separation was accomplished by providing structure via HTML, maintaining presentation in your CSS3 style sheet, and providing behaviour via JavaScript. However, HTML5 tags also aim to provide meaning to the content in the HTML document.

Tags in HTML has new definitions, stressing their semantic part. For example `<b>` is no longer *“a tag to make a text become darker and a little bigger”*. The W3C<sup>(1)</sup> now defines the `<b>` tag as *“a span of text offset from its surrounding content without conveying any extra emphasis or importance, and for which the conventional typographic presentation is bold text; for example, keywords in a document abstract, or product names in a review.”*

Hence, HTML tags can now provide structure and meaning too. A semantic element clearly defines the role of the content it contains.

---

<sup>(1)</sup> The *World Wide Web Consortium*, also known as W3C (<http://www.w3c.org>), is responsible for developing open standards for the web.

# HTML Elements

An element is composed of a beginning tag, an ending tag, and the content between the tags.

For example the following is a `<div>` element:

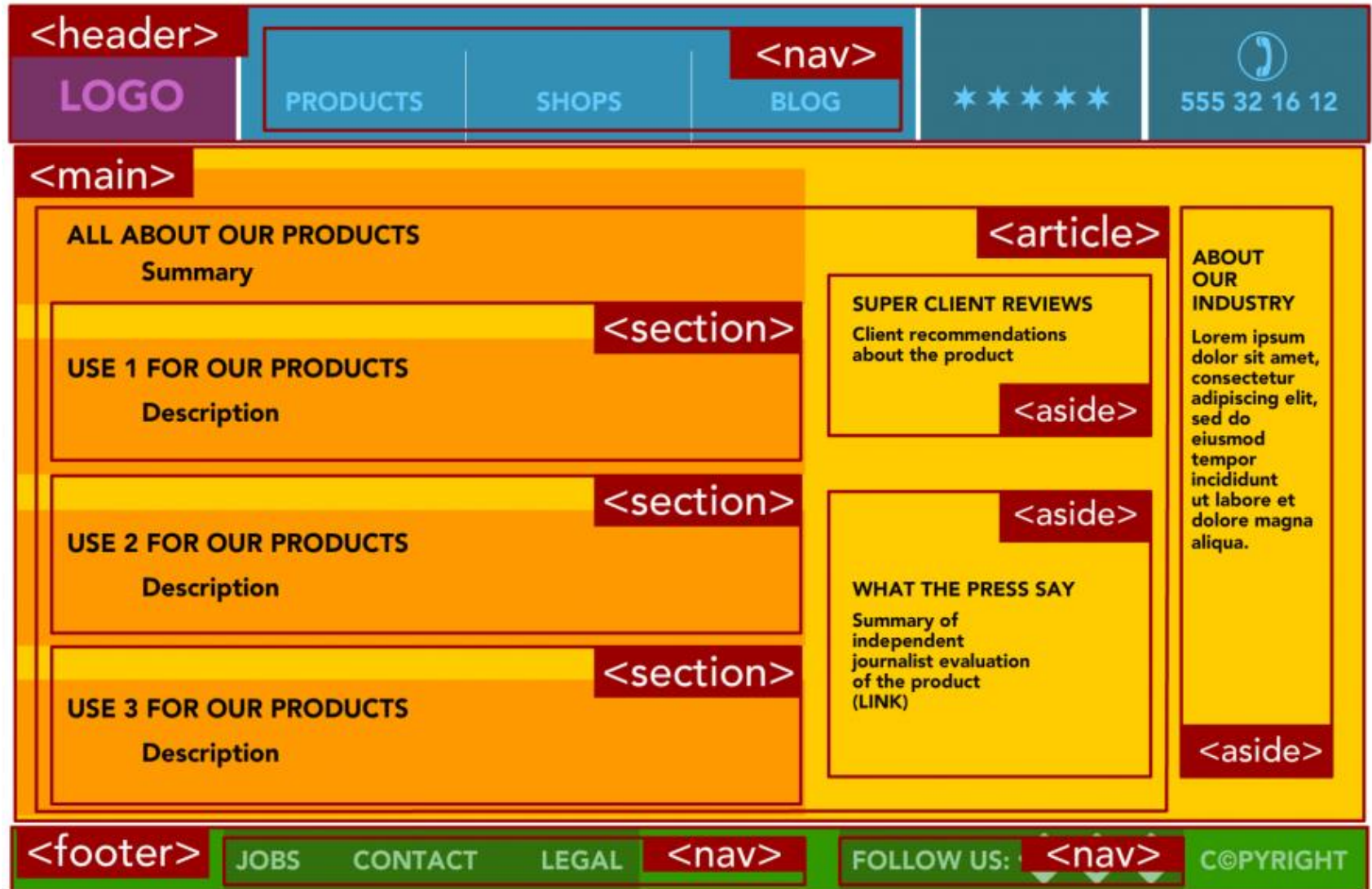
`<div>` The quick brown `<b>`fox`</b>` jumps over the lazy dog `</div>`

In this example, the `<div>` element also contains a `<b>` element. The `<b>` element consists of the beginning `<b>` tag, the content, which is the word “fox,” and the ending `</b>` tag.

HTML tag names are not case sensitive and although HTML5 does not mandate lowercase tag names, lowercase tag names are recommended.

HTML5 has more than 100 defined elements that you can use to create rich webpages and applications. The W3C defines a list of these elements with a brief description at <https://www.w3.org/wiki/HTML/Elements>

## Main “landmark” semantic elements:



You must use semantic elements to define landmarks on the page. The use of landmarks makes it easier for AT users to find their way around the page; hence enriching their UX (Accessibility). (AT: Assistive Technology)

## Adding attributes to elements

The starting tag can contain one or more attributes. An *attribute* is a name="value" pair in which name is unique within the tag and value is always enclosed within either single quotes or double quotes. You can also have Boolean attributes that contain the attribute name but no value.

Example:

```
<div id="main" class='mainContent'></div>
```

In this example, `id` and `class` are attributes. The `id` attribute uniquely identifies an element within an HTML document. The `class` attribute specifies a named CSS style that should be applied to every element having the `class='mainContent'` attribute.

## Working with Boolean attributes

*Boolean* attributes indicate that an option is set. Examples:

**checked** Used with the check box and option button to indicate selection

**selected** Used to indicate which option is selected in a drop-down or select list

**disabled** Used to disable input, text area, button, select, option, or opt group

**readonly** Used to prevent users from typing data into a text box, password, or text area

## Working with Boolean attributes

A Boolean attribute can be indicated using its minimised form (without providing a value). Example:

```
<input type="checkbox" name="fruit" value="Apple" checked />
```

Another way to indicate a Boolean attribute is to use the redundant form, in which you provide either an empty value or the name of the attribute as its value. Example:

```
<input type="checkbox" name="fruit" value="Apple" checked="" />
```

```
<input type="checkbox" name="fruit" value="Apple" checked='checked' />
```

The latter seems redundant but is usually considered to be the preferred way to represent the Boolean attribute.

## HTML global attributes

These attributes can be applied to any HTML element. These elements are listed at <http://www.htmlquick.com/reference/attributes.html>

## Expando attributes (also known as data- attributes)

They are custom attributes that you define to embed your data. Any time you want to attach data to an HTML tag, you can just create an attribute with the name of your choice and assign the data. However, to ensure that the name you create doesn't conflict with either an existing or a future W3C-defined attribute name, you will assign a name that is prefixed with **data-**.

You can use **data-** attributes :

- ✓ To include in your HTML, data that you want to use after your page has loaded but you don't want to display it.
- ✓ To store values of an element which might be required in later JavaScript interactions.
- ✓ To power UX, for example Accessibility, as demonstrated by [Bruce Lawson](#)

Data attributes should not be used if there is a existing attribute or element which is more appropriate for storing your data. In addition, custom data attributes are not intended to be publicly available or usable by external software. The presence/absence of a particular data attribute should not be used as a CSS hook for any styling either.

**Tip: Data-** attributes are mainly used in combination with JavaScript to create a more engaging user experience .



# HTML



data-

```
<ul id="vegetable-seeds">
  <li data-spacing="10cm" data-sowing-time="March to
June">Carrots</li>
  <li data-spacing="30cm" data-sowing
March">Celery</li>
  <li data-spacing="2cm" data-sowing
```

Sep  
</u

## Carrots:

Space between each plant: **10 cm**

When to sow carrots: **March to June**



## Working with self-closing tags

*self-closing tag* is a beginning tag and an ending tag in one. You end the starting tag with a space, slash, and greater-than symbol. You should use self-closing tags for tags that cannot have content. For example, the `<br>` element cannot have any content, so here is the beginning and ending tag in one: `<br />`.

Empty elements that are capable of having content but currently don't have content should have separate end tags. An example is `<div></div>`; there is no content, but the beginning and ending tags still exist.

The `<script>` tag is one of the few exemptions to this rule. When using the `<script>` element to reference an external JavaScript file, the element will not have content when used in this context, but you must always include a separate end tag as follows:

```
<script src="[full-path-to-your-JavaScript-file]"></script>
```

## Adding comments

Comments are added by using the following syntax.

```
<!--comment here -->
```

Comments are not displayed on the rendered browser page but are sent to the browser. Comments can help document your HTML source code. No spaces are allowed between the `<!` characters and the `--` characters at the beginning of the comment.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="description" content="HTML5, CSS3">
```

```
  <title>Simple HTML5 page</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <hgroup>
```

```
      <h1>Well-formed & Valid</h1>
```

```
      <h2>A Simple HTML5 example</h2>
```

```
    </hgroup>
```

```
    <nav>
```

```
      <ul>
```

```
        <a href="index.php">Home</a> <a href="">About</a> <a href="">Help</a>
```

```
      </ul>
```

```
    </nav>
```

```
  </header>
```

```
  <section> Here comes an important content
```

```
</section>
```

```
  <footer> <p>&copy; 2017</p></footer>
```

```
</body>
```

```
</html>
```

# 1 Structure

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="description" content="HTML5, CSS3">
```

```
  <title>Simple HTML5 page</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <hgroup>
```

```
      <h1>Well-formed & Valid</h1>
```

```
      <h2>A Simple HTML5 example</h2>
```

```
    </hgroup>
```

```
    <nav>
```

```
      <ul>
```

```
        <a href="index.php">Home</a> <a href="">About</a> <a href="">Help</a>
```

```
      </ul>
```

```
    </nav>
```

```
  </header>
```

```
  <section> Here comes an important content
```

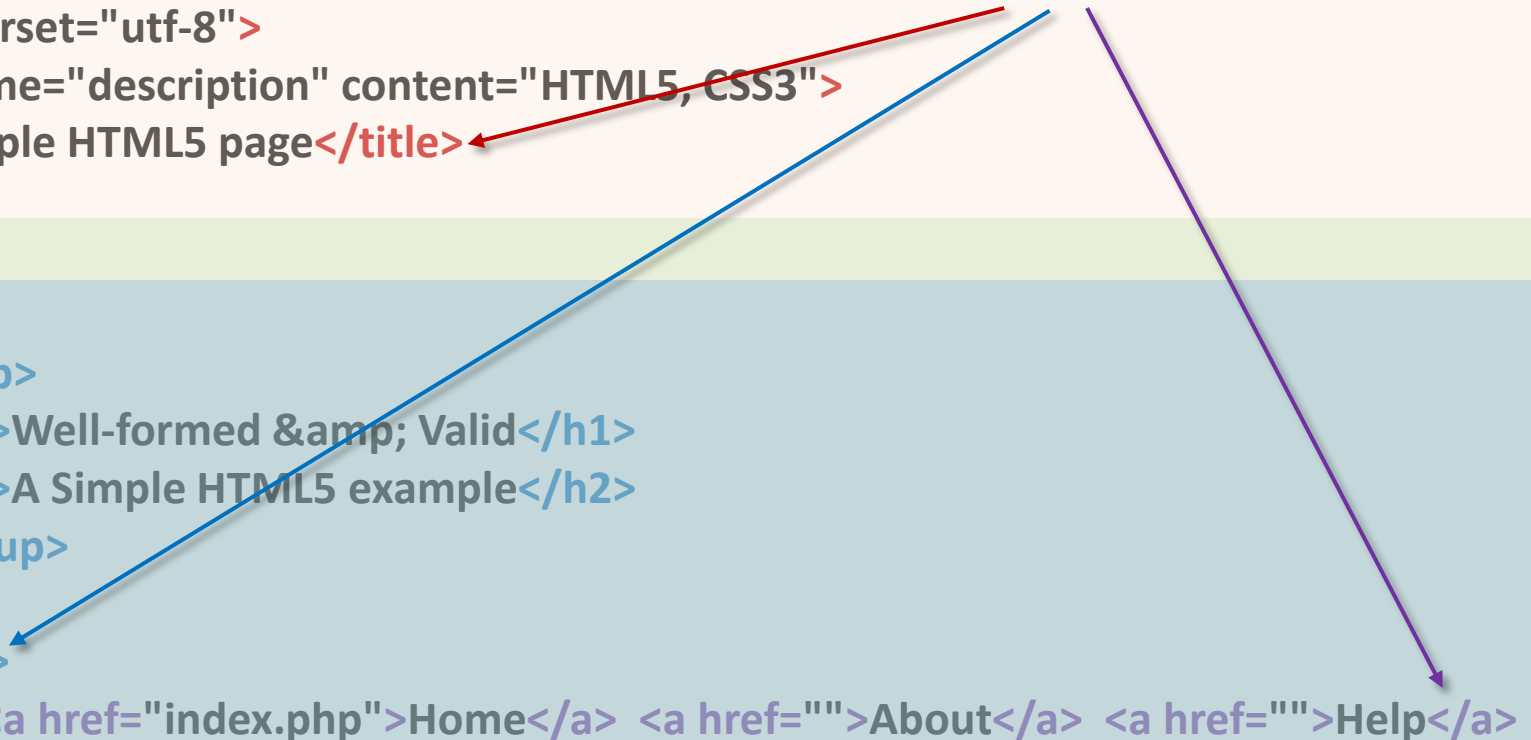
```
  </section>
```

```
  <footer> <p>&copy; 2017</p></footer>
```

```
</body>
```

```
</html>
```

## 2 Tags



```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="description" content="HTML5, CSS3">
```

```
  <title>Simple HTML5 page</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <hgroup>
```

```
      <h1>Well-formed &amp; Valid</h1>
```

```
      <h2>A Simple HTML5 example</h2>
```

```
    </hgroup>
```

```
    <nav>
```

```
      <ul>
```

```
        <a href="index.php">Home</a>
```

```
        <a href="">About</a>
```

```
        <a href="">Help</a>
```

```
      </ul>
```

```
    </nav>
```

```
  </header>
```

```
  <section> Here comes an important content
```

```
  </section>
```

```
  <footer> <p>&copy; 2017</p></footer>
```

```
</body>
```

```
</html>
```

## 3 Hyperlinks

HTML  
entities

```
<a href="index.php">Home</a>
```

```
<a href="">About</a>
```

```
<a href="">Help</a>
```

Your HTML code must be

**Well-formed** (Free from HTML syntax errors)

**and Valid** (Conforming to the W3C HTML standard)

<https://validator.w3.org>

---

## Well-formed and Valid?

<http://www2.macs.hw.ac.uk/~santiago/F27WD/html/test1.html>

Remember to make the most of HTML5 semantic:

```
<form>
```

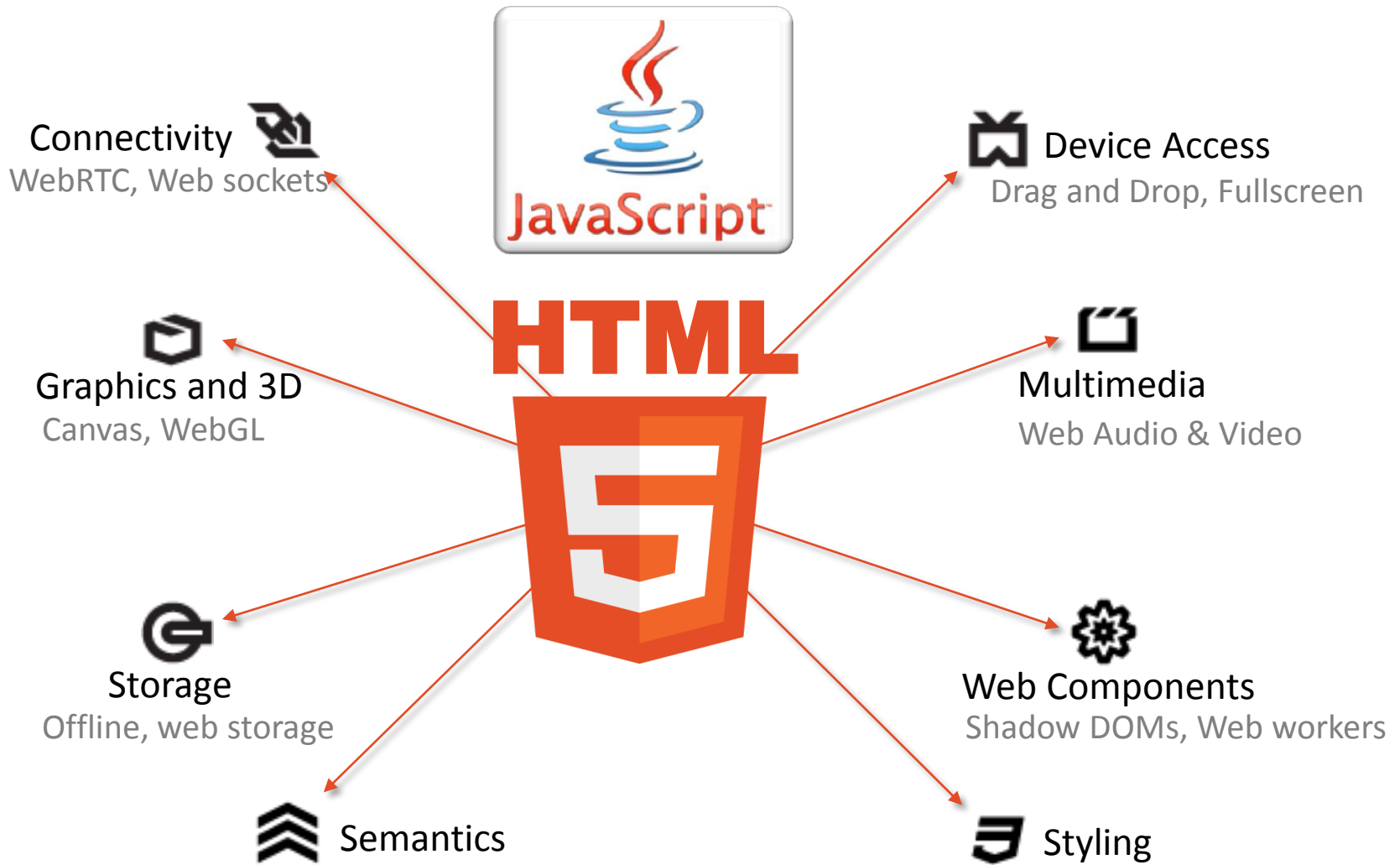
```
<div>
```

```
<label for="bday">Enter your birthday:</label>
```

```
<input type="date" id="bday" name="bday">
```

```
</div>
```

```
</form>
```



<http://www.telegraph.co.uk/foodanddrink/restaurant-reviews/8749302/Bunga-Bunga-London-restaurant-review.html>

<https://schema.org/Thing>





## Lab 2

MONDAY 21<sup>st</sup> January 2019  
at 13:15h Room EM2.50

**Mark: 1% out of 20% of CW1**

Your team is required to attend the lab session and upload to VISION:

- ✓ the mock-up or wireframe of your redesigned website (0.75%)
- ✓ the URL of your starting HTML code (IA content only) (0.25%)

[http://www2.macs.hw.ac.uk/~\[YOUR\\_USERNAME\]](http://www2.macs.hw.ac.uk/~[YOUR_USERNAME])

*PS: At least **three members** of your team must be present in the lab for your team to be awarded marks.*

Team work example: <http://www2.macs.hw.ac.uk/~gd21/IDT>