HTML

What is HTML?

- Hypertext Markup Language
- Markup language for `building` web pages
- HTML `Elements` allow you to define different areas of a web page
- Created by *Tim Berners-Lee* in late 1991
- Version: HTML, HTML 2.0, HTML 3.2, HTML 4.01, XHTML, HTML5

What we need to start?

A Web Browser

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge
- Internet Explorer

Lightweight editors

- Sublime Text
- Visual Studio Code
- Notepad++
- Atom
- Brackets

Creating an HTML file

File must end with the `.html` extension.

Runs in a web browser (Chrome, IE, Firefox, Safari, etc.,).

`index.html` is the name used for the home page of the web site.

- http://www.abc.com
  - Loads the index.html file

- http://www.abc.com/about.html
  - Loads the about.html file

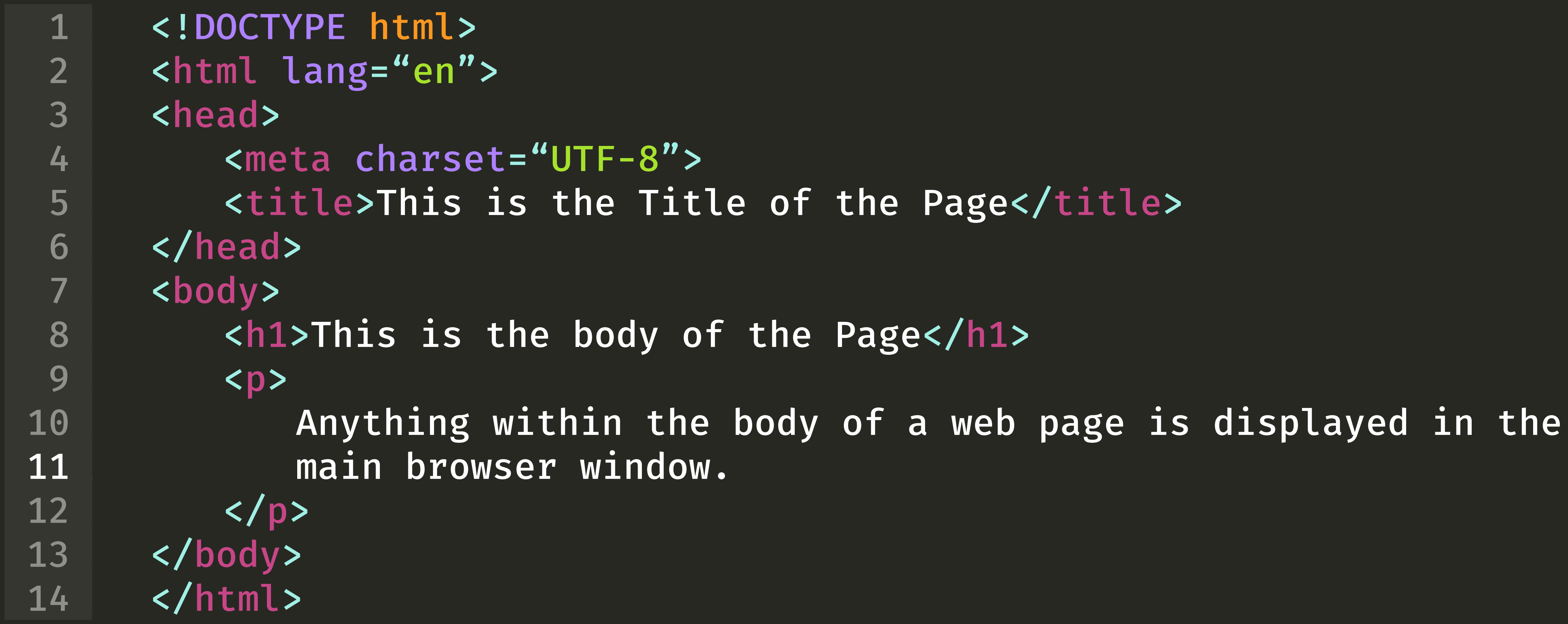
Why is it important that we name the index.html?

- The **`index.html`** page is the `most common name` used for the *default page* shown on a website.
- Websites are build inside of directories on a `Web Server`.
- Just like you have folders on your computer that you save files into, you do the same with a web server by adding your website files, including `HTML pages, images, scripts, CSS, etc.,`
- Other default page names:
  - index.htm
  - default.htm or default.html
  - home.htm or home.html
- Should have an `index.html` page in *All Your Directories.*

File naming

- Keep file names `short`
- They should be descriptive
  - Eg. “contact.html”, “about-us.html”
- `No spaces`, use underscores or hyphens instead
- Use `lower case`
  - Eg. “About-Us.html” vs “about-us.html”

HTML Structure



```html
<!DOCTYPE html>
<html lang=“en”>
<head>
    <meta charset=“UTF-8”>
    <title>This is the Title of the Page</title>
</head>
<body>
    <h1>This is the body of the Page</h1>
    <p>
        Anything within the body of a web page is displayed in the
        main browser window.
    </p>
</body>
</html>
```

BODY, HEAD & TITLE

`<body>`

You met the <body> element in the first example we created. Everything inside this element is *shown inside the main browser window*.

`<head>`

Opens the head section, which *does not appear* in the main browser window but mainly contains information about the HTML document, called metadata. It can also contain imports from external stylesheets and scripts

`<title>`

The title of the page and displayed on the tab of the page or in the title bar of the browser.

OTHERS

`<!DOCTYPE>`

Defines the HTML *version* used in the document. In this case it is HTL5.

`<html>`

Opens the page. No markup should come after the closing tag(</html>). The lang attributes declares the primary language of the page using the ISO language codes (en for English).

`<meta>`

Gives the browser some metadata about the document. The charset attribute declares the character encoding. Modern HTML documents should always use UTF-8. In HTML, the <meta> tag *does not need require a closing tag.*

`<h1>`

A level 1 heading for the page.

`<p>`

Represents a common paragraph of text.

HTML tags

Container tag

```
<opening_tag>content</closing_tag>
```

```
<html>...</html>
<body>...</body>
<div>...</div>
```

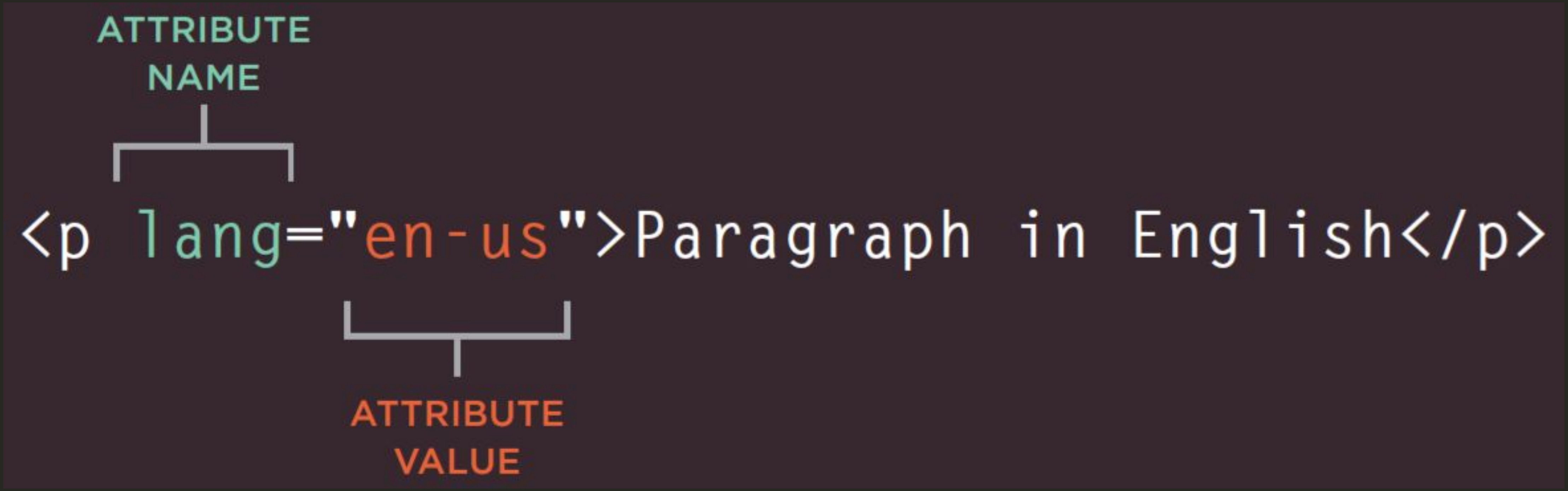
Empty tag

```
<empty>
```

```
<br>
<hr>
<img>
```

Attributes

Attributes provide additional information about the contents of an element. They appear on the opening tag of the element and are made up of two parts: a `name` and a `value`, separated by an equals sign.



```
<p lang="en-us">Paragraph in English</p>
```

TEXT

HEADINGS

HTML has `six` "levels" of headings:

- <h1> is used for main headings
- <h2> is used for subheadings

If there are further sections under the subheadings then the <h3> element is used, and so on...

```html
<h1>Heading Level 1</h1>
<h2>Heading Level 2</h2>
<h3>Heading Level 3</h3>
<h4>Heading Level 4</h4>
<h5>Heading Level 5</h5>
<h6>Heading Level 6</h6>
```

PARAGRAPHS

To create a paragraph, surround the words that make up the paragraph with an opening <p> tag and closing </p> tag.

```
<P>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Minima,
    necessitatibus in voluptates blanditiis porro aspernatur ipsum
    dignis simos as periores, soluta iusto perspiciatis, saepe. At
    eaque aliquid aspernatur placeat odit laborum ab.ernatur ipsum
<P>
```

BOLD & ITALIC

<b>

By enclosing words in the tags <b> and </b> we can make characters appear `bold`.

```html
<p>This is how we make a word appear <b>bold.</b></p>
```

<i>

By enclosing words in the tags <i> and </i> we can make characters appear `italic`.

```html
<p>This is how we make a word appear <i>italic.</i></p>
```

SUPERSCRIPT & SUBSCRIPT

<sup>

The <sup> element is used to contain characters that should be superscript such as the suffixes of dates or mathematical concepts like raising a number to a power such as 2^2.

```html
<p>On the 19<sup>th</sup> of July.</p>
```

<sub>

The <sub> element is used to contain characters that should be subscript. It is commonly used with footnotes or chemical formulas such as H_20.

```html
<p>The amount of CO<sub>2</sub></p>
```

WHITE SPACE

In order to make code easier to read, web page authors often add extra spaces or start some elements on new lines.

When the browser comes across *two or more spaces* next to each other, it only displays `one space`. Similarly if it comes across a line break, it treats that as a single space too. This is known as `white space collapsing`.

```
<P>This is a              first



    paragraph.
<P>
```

LINE BREAKS & HORIZONTAL RULES

<br />

As you have already seen, the browser will automatically show each new paragraph or heading on a new line. But if you wanted to add a line break inside the middle of a paragraph you can use the line break tag <br />.

```
<p>This is a <br> new line.</p>
```

<hr />

To create a break between themes — such as a change of topic in a book or a new scene in a play — you can add a horizontal rule between sections using the <hr /> tag.

```
<p>This is a first line.</p>
<hr>
<p>This is a second line.</p>
```

SEMANTIC MARKUP

There are some text elements that are not intended to affect the structure of your web pages, but they do add extra information to the pages — they are known as `semantic markup`.

The reason for using these elements is that other programs, such as `screen readers` or `search engines`, can use this extra information.

For example, the voice of a screen reader may add emphasis to the words inside the `<em>` element, or a search engine might register that your page features a quote if you use the <blockquote> element.

STRONG & EMPHASIS

<stong>

The use of the <strong> element indicates that its content has strong importance. For example, the words contained in this element might be said with strong emphasis.

By default, browsers will show the contents of a <strong> element in bold.

```html
<p><strong>Beware:</strong> Pickpockets operate in this area.</p>
```

<em>

The <em> element indicates emphasis that subtly changes the meaning of a sentence.

By default browsers will show the contents of an <em> element in italic.

```html
<p>I <em>think</em> John was the first.</p>
```

AUTHOR DETAILS

<address>

The <address> element has quite a specific use: to contain contact details for the *author* of the page.

It can contain a physical address, but it does not have to. For example, it may also contain a *phone number* or *email address*.

```html
<address>
    <p>
        <a href=“mailto:john-doe@email.com”>john-doe@email.com</a>
        <a href=“tel:123456789”>(+95) 123456789</a>
    <p>
</address>
```

CHANGES TO CONTENT

`<ins>` `<del>`

The <ins> element can be used to show content that has been `inserted` into a document, while the <del> element can show text that has been `deleted` from it.

```
<p>It was the <del>worst</del> <ins>best</ins> Idea</p>
```

`<s>`

The <s> element `indicates` something that is no longer accurate or relevant (but that should *`not be deleted`*).

```
<p>MacBook Pro</p>
<p><s>Was $2,000</s></p>
<p>Now only $1,500</p>
```

LISTS

ORDERED LISTS

`<ol>`

The ordered list is created with the `<ol>` element.

`<li>`

Each item in the list is placed between an opening `<li>` tag and a closing </li> tag. (The li stands for *`list item`*.)

```
<ol>
    <li>Ordered list item 1</li>
    <li>Ordered list item 2</li>
    <li>Ordered list item 3</li>
</ol>
```

UNORDERED LISTS

`<ul>`

The unordered list is created with the `<ul>` element.

`<li>`

Each item in the list is placed between an opening `<li>` tag and a closing </li> tag. (The li stands for *`list item`*.)

```
<ul>
    <li>Unordered list item 1</li>
    <li>Unordered list item 2</li>
    <li>Unordered list item 3</li>
</ul>
```

DEFINITION LISTS

<dl>

The definition list is created with the `<dl>` element and usually consists of a series of terms and their definitions.

<dt>

This is used to contain the `term` being defined (the definition term).

<dt>

This is used to contain the `definition`.

```html
<dl>
    <dt>HTML</dt>
    <dd>Hypertext Markup Language</dd>
</dl>
```

NESTED LISTS

You can put a second list *inside an* `<li>` element to create a `sublist` or `nested list`.

```html
<ul>
    <li>List Item 1</li>
    <li>List Item 2
        <ul>
            <li>Sub-list Item 1</li>
            <li>Sub-list Item 2</li>
            <li>Sub-list Item 3</li>
        </ul>
    </li>
    <li>List Item 3</li>
</ul>
```

LINKS

LINKING TO OTHER SITES

<a>

Links are created using the <a> element which has an attribute called href. The value of the href attribute is the page that you want people to go to when they click on the link.

```html
<h3>Social Networking Sites</h3>
<ul>
    <li><a href=“http://www.facebook.com”>Facebook</a></li>
    <li><a href=“http://www.twitter.com”>Twitter</a></li>
    <li><a href=“http://www.instagram.com”>Instragram</a></li>
</ul>
```

LINKING TO OTHER PAGES ON THE SAME SITE

<a>

When you are linking to other pages within the same site, you `do not need to specify` the domain name in the URL. You can use a shorthand known as a `relative URL`.

```
<h1>My Website</h1>
<ul>
   <li><a href=“index.html”>Home</a></li>
   <li><a href=“about.html”>About</a></li>
   <li><a href=“contact.html”>Contact</a></li>
</ul>
```

EMAIL LINKS

`mailto:`

To create a link that starts up the user's email program and addresses an email to a specified email address, you use the `<a>` element. However, this time the value of the href ***attribute starts*** with `mailto:` and is followed by the email address you want the email to be sent to.

```
<a href=“mailto:johndoe@email.com”>Email: John-Doe</a>
```

LINKING TO A SPECIFIC PART OF THE SAME PAGE

At the top of a long page you might want to add a list of contents that links to the corresponding sections `lower down`. Or you might want to add a link from part way down the page back to the top of it to save users from having to `scroll back to the top`.

```
<h1 id=“top”>Top</a>
<a href=“#bottom”>Goto bottom</a>
<p>
   ...
</p>
<h1 id=“bottom”>Bottom</a>
<a href=“#top”>Goto top</a>
```

IMAGES

IMAGE

`<img>`

To add an image into the page you need to use an `<img>` element. This is an empty element (which means there is ***`no closing tag`***). It must carry the following two attributes:

`src`

This tells the browser where it can find the image file. This will usually be a **`relative URL`** `pointing` to an image on your own site.

`alt`

This provides a text description of the image which describes the image if you `cannot see` it.

`title`

Most browsers will display the content of this attribute in a `tooltip` when the user hovers over the image.

WHERE TO PLACE IMAGES IN YOUR CODE

```
<img src=“https://picsum.photos/id/45/100/100” alt=“Car”>
<p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ipsam quas nam eveniet
    unde soluta non suscipit dolor odio nihil debitis, quibusdam incidunt, repudi andae
    omnis placeat asperiores, itaque libero perspiciatis distinctio.
</p>
<hr>
<p>
    <img src=“https://picsum.photos/id/45/100/100” alt=“Car”>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ipsam quas nam eveniet
    unde soluta non suscipit dolor odio nihil debitis, quibusdam incidunt, repudi andae
    omnis placeat asperiores, itaque libero perspiciatis distinctio.
</p>
<hr>
<p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ipsam quas nam eveniet
    unde soluta <img src=“https://picsum.photos/id/45/100/100”
    alt=“Car”>  non suscipit dolor odio nihil debitis, quibusdam incidunt, repudi
    andae omnis placeat asperiores, itaque libero perspiciatis distinctio.
</p>
```

THREE RULES FOR CREATING IMAGES

There are three rules to remember when you are creating images for your website which are summarized below. We go into greater detail on each topic over the next nine pages.

1

SAVE IMAGES IN THE RIGHT FORMAT

2

SAVE IMAGES IN THE RIGHT SIZE

3

USE THE CORRECT RESOLUTION

TABLES

BASIC TABLE

`<table>`

The <table> element is used to `create a table`. The contents of the table are written out row by row.

`<tr>`

You indicate the start of each row using the opening <tr> tag. (The tr stands for `table row`.)

`<td>`

Each cell of a table is represented using a <td> element. (The td stands for `table data`.)

BASIC TABLE STRUCTURE

```
<table>
    <tr>
        <td>Name</td>
        <td>Email</td>
    </tr>
    <tr>
        <td>John Doe</td>
        <td>johndoe@email.com</td>
    </tr>
    <tr>
        <td>Will Smith</td>
        <td>willsmith@email.com</td>
    </tr>
</table>
```

LONG TABLE

There are `three elements` that help distinguish between the main content of the table and the first and last rows (which can contain different content).

`<thead>`

The `headings` of the table should sit inside the <thead> element.

`<tbody>`

The `body` should sit inside the <tbody> element.

`<tfoot>`

The `footer` belongs inside the <tfoot> element.

LONG TABLE STRUCTURE

```html
<table>
    <thead>
        <tr>
            <th>Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>John Doe</td>
            <td>johndoe@email.com</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan=“2”>Copyright &copy; 2019</td>
        </tr>
    </tfoot>
</table>
```

FORMS

FORM STRUCTURE

<form>

Form controls live inside a `<form>` element. This element should always carry the action attribute and will usually have a method and id attribute too.

action

Every <form> element requires an action attribute. Its value is the `URL` for the page on the server that will receive the information in the form when it is submitted.

method

Forms can be sent using one of two methods: `get` or `post`.

```
<form action=“process.php” method=“post”></form>
```

INPUT: TEXT, PASSWORD

<input>

type=“text”

When the type attribute has a value of text, it creates a single line text input.

```
<input type=“text”>
```

type=“password”

When the type attribute has a value of password it creates a text box that acts just like a single-line text input, except the characters are blocked out.

```
<input type=“password”>
```

TEXT AREA

<textarea>

The <textarea> element is used to create a `multi line text input`. Unlike other input elements this is not an empty element. It should therefore have an opening and a closing tag.

```
<textarea name=“message” cols=“30” rows=“10”></textarea>
```

RADIO BUTTON

<input>

type=“radio”

Radio buttons allow users to pick just one of a number of options.

```
<input type=“radio” name=“gender” value=“male” id=“male”> Male
<input type=“radio” name=“gender” value=“female” id=“female”> Female
```

CHECKBOX

`<input>`

`type=“checkbox”`

Checkboxes allow users to select (and unselect) `one or more options` in answer to a question.

```
<input type=“checkbox” checked> HTML
<input type=“checkbox”> CSS
<input type=“checkbox”> JS
```

LABEL FORM CONTROLS

`<label>`

When introducing form controls, the code was kept simple by indicating the purpose of each one in text next to it. However, each form control should have its own <label> element as this makes the form accessible to `vision-impaired users`.

`for`

The for attribute states which form control the label belongs to. Note how the radio buttons use the `id attribute`. The value of the id attribute `uniquely` identifies an element from all other elements on a page.

DROP DOWN LIST BOX

<select>

A drop down list box (also known as a `select box`) allows users to select one option from a drop down list.

The <select> element is used to create a drop down list box. It contains `two or more <option>` elements.

```
<select>
    <option selected>Apple</option>
    <option>Orange</option>
    <option>Lemon</option>
    <option>Banana</option>
<select>
```

BUTTON

<button>

The <button> element was introduced to allow users more control over how their buttons appear, and to allow other elements.

```html
<button>Read more &raquo;</button>
```

SUBMIT BUTTON

`<input>`

`type=“submit”`

The submit button is used to `send` a form to the server.

```
<input type=“submit” value=“Log In”>
```

IMAGE BUTTON

<input>

type=“image”

If you want to use an image for the submit button, you can give the type attribute a value of image. The src, width, height, and alt attributes work just like they do when used with the <img> element.

```
<input type=“image” src=“https://picsum.photos/id/45/100/40”
width=“100” height=“40”>
```

HTML5: FORM VALIDATION

```
<input>

type=“email”
type=“date”
type=“search”
type=“url”

placeholder
required
```

EXTRA MARKUP

COMMENTS IN HTML

```
<!-- -->
```

If you want to add a comment to your code that will not be visible in the user's browser, you can add the text between these characters:

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

ID ATTRIBUTE

Every HTML element can carry the `id attribute`. It is used to `uniquely` identify that element from other elements on the page. Its value should start with `a letter` or `an underscore` (not a number or any other character). It is important that `no two elements` on the the ***same page*** have the same value for their id attributes (otherwise the value is no longer unique).

```
<h1 id="head">
    This is a ID attribute.
</h1>
```

CLASS ATTRIBUTE

Every HTML element can also carry a `class attribute`. Sometimes, rather than uniquely identifying one element within a document, you will want a way to `identify several elements` as being different from the other elements on the page.

```html
<p class=“para”>This is a first paragraph.</p>

<p class=“para”>This is a second paragraph.</p>
```

BLOCK ELEMENTS

Some elements will always appear to start on a `new line` in the browser window. These are known as `block level elements`.

Examples of block elements are `<h1>, <p>, <ul>, and <li>.`

```
<h1>Heading Level 1</h1>

<p>This is a paragraph.</p>

<ul>
    <li>Unordered List 1</li>
    <li>Unordered List 2</li>
    <li>Unordered List 3</li>
<||ul>
```

INLINE ELEMENTS

Some elements will always appear to continue on the `same line` as their neighbouring elements. These are known as `inline elements`.

Examples of inline elements are `<a>, <b>, <em>, and <img>.`

```
<a href=“http://www.facebook.com”>Facebook</a>

<b>Inline Bold</b>

<em>Inline Emphasis</em>

<img src=“https://picsum.photos/id/45/100/100” alt=“Car”>
```

ELEMENTS IN A BLOCK

`<div>`

The <div> element allows you to `group a set of elements` together in one block-level box.

For example, you might create a <div> element to contain all of the elements for the `header` of your site (*`the logo and the navigation`*), or you might create a <div> element to contain comments from visitors.

```
<div>
    <!-- header -->
    <!-- navigation -->
</div>
```

ELEMENTS INLINE

`<span>`

The most common reason why people use `<span>` elements is so that they can control the appearance of the content of these elements *`using CSS`*.

```
<p>
    Hello!, Welcome to my <span>home<span>.
</p>
```

ESCAPE CHARACTERS

There are some characters that are used in and reserved by `HTML code`. (For example, *`copyright or space`*.)

```html
<p>
    Copyright &copy;&nbsp;2019.
</p>
```