# 1. WHAT IS HTML?

- HTML stands for **Hyper Text Markup Language**.
- HTML describes the **structure of web pages** using markup.
- HTML is used to provide the content(words, images, audio, video, and so on) to the web pages.
- HTML is a tag based language. They are defined within the **angle brackets**.
- HTML file can be created using a text editor.

Following is a sample HTML code :-

```
<!DOCTYPE html>
<html>
    <head></head>

    <body>
        This is sample tex....<br/> You are beginning to learn HTML.
        <!-- We use this syntax to write comments -->
        <!-- Page content and rest of the tags here.... -->
        <!-- This is the actual area that gets shown in the browser -->
    </body>

</html>
```

Remember to save the above file with **".html" extension.**

Start the browser and open the file; it will appear as:


# 2. COMMENTS IN HTML

The comment tag **<!--...-->** is used to insert comments in the source code. Comments are **not displayed in the browsers**.

You can use comments to explain your code, which can help you when you edit the source code at a later date. This is especially useful if you have a lot of code.

# 3. TAGS

Tags define all elements of the document, i.e. they give meaning to the plain text of HTML.

- HTML tags are surrounded by the **two characters < and >** (They are called **angle brackets**).
- The tag name can either start from an **alphabet** or an **underscore(_)**.
- The text between the start and end tags is the **element content**.
- Tags with an opening and closing can have **any number of tags within itself.**
- HTML tags are **not case sensitive**, <p> means the same as <P>.
- HTML tags normally **comes in pairs(container tags)**, i.e. both opening and closing(it is same, just the name of the tag with character '/ ' in the beginning) tag.
- Eg: **<html> and </html>** is a tag that comes in pair.
- Eg: **<br>** does not have a closing tag.

## 3.1. Description of tags used till now

- ***<!DOCTYPE html>*** tells the browser that the file being displayed is HTML5 page.
- ***<html>*** *</html>* meant to contain all the html data and is the **start of an HTML document.**
- ***<head>*** *</head>* provides **information about the document.** It is not displayed in the browser window.
- ***<title>*** *</title>* provides a **title for the document.**
- ***<body>*** *</body>* contains all the **things visible** on the web page.

***NOTE:*** *You might come across "self-closing" tags, whereby a br tag, for eg., will look like "**<br/>**" instead of simply "**<br>**".*

***EXTRA:***

***To get the list of all valid tags in HTML5, visit:***
***https://developer.mozilla.org/en-US/docs/Web/HTML/Element***

# 4. DOCTYPE

The **DOCTYPE** declaration describes **what version of HTML the page is written in.** It is the very first thing in your HTML document that you see in every web page. It is written at the top of every page before the <html> tag.
The doctype declaration is not an HTML tag. It is the one recommended by HTML5.

The syntax for doctype is: **<!DOCTYPE html>**.

# 5. HTML ELEMENTS

Elements are the things that actually make up the web page. **Tags just define the beginning and end of the elements**. Everything that a web page includes is an HTML element.

Eg: this is an HTML element: `<b>This text is bold</b>`

- ➢ The HTML element starts with a opening tag: **
- ➢ The content of the HTML element is: *This text is bold*
- ➢ The HTML element ends with an closing tag: **

The basic elements used till now have been briefly described below:

## 5.1. Paragraphs

Paragraphs are **blocks of text** separated from each other by some space. They are defined using the ** and ** tags. When p element ends, the *next element appears in next line.*

Eg: here's a sample of code for  tag

```
<!DOCTYPE html>
<html>
    <head>
        <title>p tag</title>
    </head>
    <body>
        <p>This is line 1.</p>
        <p>This is line 2.</p>
        <!-- trying to format the text without using p-tag -->
        This is line 1.
        This is line 2.
        This is line 3.
    </body>
</html>
```

It appears on a web browser like this:

This is line 1.

This is line 2.

This is line 1. This is line 2. This is line 3.

***NOTE:*** *When formatting without p-tag, new lines are appended on the current line. This happens because the* ***spacing of text doesn't matter to the browser****.*

## 5.2. Headings

These are tags in HTML to mark some content as headings. In fact, there are six different levels of headings **h1, h2, h3, h4, h5** and **h6**. Among them *h1 defines the largest heading* and *h6 defines the smallest level heading*.

Eg: here's a sample of code for H tags

```
<!DOCTYPE html>
<html>
    <head>
        <title>Heading Levels</title>
    </head>
    <body>
        <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>
    </body>
</html>
```

The content appears as:

# Heading level 1

## Heading level 2

### Heading level 3

#### Heading level 4

##### Heading level 5

###### Heading level 6

## 5.3. BR Tag

**<br>** tag is used to introduce a **single line break** between the contents. This means that when this tag is used in between a single line, the content after this tag will move to the next line. Do not use it to provide space between block of elements( eg., paragraph and heading).

Eg.,

```
<h3>We are studying in<br>Coding Ninjas</h3>
```

will show the heading as:

**We are studying in**
**Coding Ninjas**

# 6. LISTS

Lists are used to ***group together*** *related pieces of information* so they are clearly associated with each other and easy to read. Lists are good from a **structural** point of view as they help create a well-structured, more accessible, easy to maintain document.
HTML supports ordered, unordered and definition lists.

## 6.1. Unordered Lists

It is used to group a set of related items in **no particular order**. Unordered lists are used when the numbering of items is not required. By default the items are followed by **bullets**.

They are defined using the **<ul>** tag. Eg:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Unordered Lists</title>
    </head>
    <body>
        <h1>Lists</h1>
        <ul>
            <li>first item</li>
            <li>second item</li>
            <li>third item</li>
        </ul>
    </body>
</html>
```

The output is as follows:

**Lists**

- first item
- second item
- third item

HTML provides an interesting feature to change the style of the list item marker. There are 4 types of style in unordered lists:

- ● ***type*="disc"** - sets the list item marker to a bullet (default)
- ○ ***type*="circle"** - sets the list item marker to a circle
- ■ ***type*="square"** - sets the list item marker to a square
  ***type*="none"** - the lists items will not be marked

***NOTE:*** *The above styles can be produced by using the* ***'type'*** *attribute. However, this attribute is now* ***not supported in HTML5*** *and you now need to change the style using CSS(we will learn later about it).*

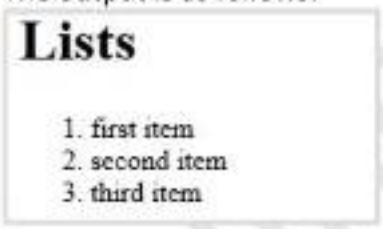
## 6.2. Ordered Lists

It is used to group a set of related items in a **specific order**. Ordered lists are used when the numbering of items is required. By default the items are followed by **numerical numbering**.

They are defined using the **<ul>** tag. Eg:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Unordered Lists</title>
    </head>
    <body>
        <h1>Lists</h1>
        <ol>
            <li>first item</li>
            <li>second item</li>
            <li>third item</li>
        </ol>
    </body>
</html>
```

The output is as follows:




Similarly, like the unordered lists, there are also different types of ways to number the ordered lists using the **'type'** attribute:

1. ***type*="1"** - The list items will be numbered with numbers (default)
A. ***type*="A"** - The list items will be numbered with uppercase letters
a. ***type*="a"** - The list items will be numbered with lowercase letters
I. ***type*="I"** - The list items will be numbered with uppercase roman numbers
i. ***type*="i"** - The list items will be numbered with lowercase roman numbers

*Now, what if you want to change the starting numbering of the lists?*
HTML has got the solution for it: the **'start'** attribute.
So, if we change <ol> to **<ol start="7" >**, you will now see the output as:

## 6.3. Description Lists

A definition list is not a list of items. This is a list of terms and explanation of the terms.
A definition list starts with the **<dl>** tag. Each definition - list term starts with the **<dt>** tag. Each definition - list definition starts with the **<dd>** tag.

Description lists are very specific in use as compared to ordered and unordered lists and hence are very less used. But whenever, a structure like a list of terms and their description is required, the description lists are the the perfect elements.

Eg:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Description Lists</title>
    </head>
    <body>
        <h2>Description List</h2>
        <dl>
            <dt>Coffee</dt>
                <dd>- black hot drink</dd>
            <dt>Milk</dt>
                <dd>- white cold drink</dd>
        </dl>
    </body>
</html>
```

The output is as follows:

**A Description List**

Coffee
- black hot drink

Milk
- white cold drink

## 7. NESTING ELEMENT

HTML elements can be nested i.e. **elements can contain elements**. Actually, all HTML documents consist of nested HTML elements.

Eg:

```
<ul>
    <li>first item</li>
    <li>second item
        <!-- Look, the closing </li> tag is not placed here! -->
        <ul>
            <li>second item first subitem</li>
            <li>second item second subitem
                <!-- Same for the second nested unordered list! -->
                <ul>
                    <li>second item second subitem first sub-subitem</li>
                    <li>second item second subitem second sub-subitem</li>
                    <li>second item second subitem third sub-subitem</li>
                </ul>
            </li>
```

```
            <!-- Closing </li> tag for the list that contains the third unordered
list -->
            <li>second item third subitem</li>
        </ul>
    </li><!-- Here is the closing </li> tag -->
    <li>third item</li>
</ul>
```

This will give the output as:

Lists

- first item
- second item
  - second item first subitem
  - second item second subitem
    - second item second subitem first sub-subitem
    - second item second subitem second sub-subitem
    - second item second subitem third sub-subitem
  - second item third subitem
- third item

***NOTE:*** *There is no limitation to the depth of nested lists. Although it is true for all paired/container tags, but we should be careful in nesting elements inside each other and should only do for something meaningful.*

# 8. IMAGES IN HTML

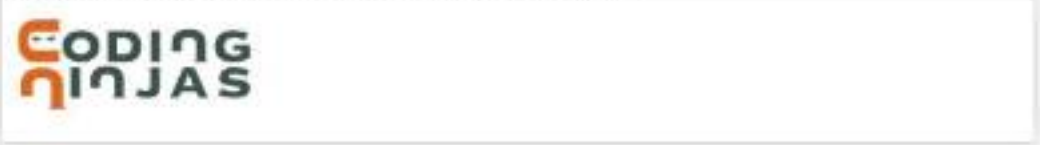
With HTML you can also display images in a document. In HTML, images are defined with the **<img>** tag.

To display an image on a page, you need to use the **src** attribute. Src stands for "source". The value of the src attribute is the **URL of image** you want to display on your page.

The syntax of defining an image:

```
<img src="images/logo.png">
```

The image will now be displayed on the page like:




Some points you need to know:

- <img> tag is a **self closing tag** which means that it doesn't contain the closing tag.
- The **src** tag can contain both relative and absolute paths, as well as internet image links.

## 8.1. The ALT Attribute

The **alt** attribute or **alternate text** tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image.

Now, we can use the **alt** attribute as:

```
<img src="images/logo.png" alt="Coding Ninjas Image">
```

The text would be seen now as:

Coding Ninjas Logo

***NOTE:** It is a good practice to include the "**alt**" attribute for each image on a page.*

### 8.2. Height and Width

The height and width of an image can be set directly by using the **height="value"** and **width="value"** attributes. By default, the value provided is in pixels.

Eg.,

```
<img src="images/logo.png" alt="Coding Ninjas Image" height="500"
width="500">
```

This will fix the height and width of the image to 500px(pixel).
There is an alternate for height and width attribute in CSS. We can come to this later.

***NOTE:** Value provided should be in **numerical** form. Pixel is a unit of measurement, to set the dimensions of the image.*

## 9. ATTRIBUTES

Attributes can provide **additional information** about the HTML elements on your page and **control their behaviour**.
Eg: `<tag_name attribute_name="value_value">Content Enclosed</tag_name>`.

Some points to remember:

i. Attributes always come in name/value pairs like this: **attribute_name="value"**.
ii. Attributes are always added to the **start tag** of an HTML element.
iii. Attribute values should always be enclosed in quotes. **Double style quotes (" ")** are the most common, but **single style quotes (' ')** are also allowed.
iv. In some rare situations, like when the *attribute value itself contains quotes*, it is necessary to use single quotes: **name='John "ShotGun" Nelson'** and vice-versa.

## 10. ANCHOR TAG

The <a> tag defines a **hyperlink**, which is used to link from one page to another.

You have seen that clicking on a link opens a new page may be on the same page or another. These web pages are connected using links. They give us the ability to go to a different web page without each time entering its URL. This kind of links are **external links** i.e. they help in connecting to external web pages.

Links can also be **internal** which means that they will be linking the content within the same page. Eg: link to the top of the page or any link to any specific content on the page.

By default, links will appear as follows in all browsers:

- An **unvisited** link is underlined and blue
- A **visited** link is underlined and purple
- An **active** link is underlined and red

## 10.1. href Attribute

The most important attribute of the <a> element is the **href** attribute, which indicates the **link's destination**. In other words, the href attribute is used to address the document to link to.

Eg:

```
<h2>A Great place to practice codirg</h2>
<p> Take daily challenges at
    <a href="http://www.codingninjas.in/students/assignments">Coding
    Ninjas</a>.
        <!-- clickable content for the link is mentioned here -->
        <!-- any html element can be included here like image, gif, etc. -->
</p>
```

You will see this:

**A Great place to practice coding**

Take daily challenges at Coding Ninjas.

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc. These all are known as **external links**.

***NOTE:*** *You need to remember that here also, we can provide* ***relative url*** *of a file as a value to href attribute. Eg:* ***href="/home/myPC/Documents/test.html".***

## 10.2. Relative and Absolute Linking

**Relative linking** is used to **specify local links**, i.e. link to files inside the root folder.
**Absolute linking** is used to **specify outside links**, i.e. URL of the web pages.

Relative links works relative to the page. So, when a user clicks a relative link, the browser looks for the location of the file relative to the current page.
Four situation arises in this case:-

- ***File is present in the same folder*** - In this case, the name of the file is provided. Eg: `<a href="relativeFile.html">Click Me</a>`, will look for the file inside the same folder.
- ***File is present in the subfolder*** - In this case, the name of the file provided is preceded with the folder names according to hierarchy. Eg: `<a href="subfolder/down/relativeFile.html">Click Me</a>`, will move to the 'subfolder' folder, then to 'down' folder and look for the file inside it.
- ***File is present somewhere in the parent folder*** - In this case, to move one folder above use '**../**'. Eg: `<a href="../relativeFile.html">Click Me</a>`, will move to the parent folder and look for the file inside it.
- ***File is present in another subfolder of parent folder*** - This case covers above two cases. Eg: `<a href="../subfolder/relativeFile.html">Click Me</a>`, will move to the parent folder, then to folder named 'subfolder' and look for the file inside it.

Absolute links provides complete web address of the web page where you want to go. Eg: `<a href="https://www.codingninjas.in/">Click Me</a>`, will make the browser directly go to the specified URL.

## 10.3. target Attribute

With the **target** attribute, you can define where the **linked document will be opened**. The target attribute has the following values:

- **_self**: load the URL into the current tab itself. This is the default.
- **_blank**: load the URL into a new tab or browser window.
- **_parent**: load the URL into the parent browsing context. If there is no parent, this behaves same as _self.
- **_top**: load the URL into the top-level browsing context. If there is no parent, this behaves same as _self.

The line below will open the document in a new browser window:

```
<a href="http://www.codingninjas.in/students/assignments"
target="_blank">Coding Ninjas</a>
```

***NOTE:*** *By default, linked page will be displayed in the* ***current browser window****.*

# 1. <DIV> TAG

The  tag defines a **block-level section** or a division in an HTML document.
The  tag is a block element. It is often used as a **container** for other HTML elements.
The  element has *no required attributes*.

The  element is very often used together with CSS, to layout a web page. By default, browsers always place a line break before and after the  element.

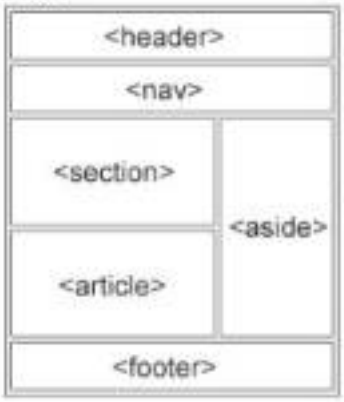
# 2. SEMANTIC ELEMENTS

**Semantic HTML** or semantic markup is HTML that **introduces meaning** to the web page rather than just presentation. For eg., a **<p>** tag indicates that the enclosed text is a paragraph.

Semantic HTML tags *provide information about the contents of those tags* that goes beyond just how they look on a page. For eg., text that is enclosed in the **<code>** tag is immediately recognized by the browser as some type of coding language.

Examples of **non-semantic** elements: * * and ** - tells nothing about its content.
Examples of **semantic** elements: *<h1>* to *<h6>*, *<p>*, and *<ul>* - clearly defines its content.

Now, here is an example of how some of the common semantic tags are used to construct a page:




***EXTRA:***

***To know more about semantic elements and the benefit of using them, visit:***
***https://developer.mozilla.org/en-US/docs/Glossary/Semantics#Semantics_in_HTML***

# 3. INTERNAL LINKS

Instead of having to resort to the task of scrolling down long pages, you can make your readers very happy by offering them page jumps as an alternative mode of transport around

your site. Basically, page jumps are just **links** (they use the same <a> element as all links), but links that point to a certain part of the same document, i.e. internal links.

You simply add a unique id value to an existing element. Now, you will know about id attribute afterwards in CSS, so don't stress too much on it for now.

For eg., here's how you set up a link:

```
<h2 id="heading">This is the top</h2>
    .....
    .....
<a href="#heading">Go to top</a>
```

**Explanation:** In the **href** attribute, *heading* is the id of the heading of this page. *id* of an html element is an attribute and it can have any value. While referring to an id, '#' is used in the beginning of its name. Clicking on the link shown below will scroll you to the heading such that it is the first line of the display.

## 4. BLOCK VS INLINE ELEMENTS

**Block elements** are those that take up the **full width** available on a web page, effectively blocking out any other elements from sitting next to it on the left or right.
**Inline elements** are those who only take up as much **width as much needed** to display the contents of the element, thereby allowing other elements to be in line with the inline element.

**Block elements** always start on a new line.
**Inline elements** does not start from a new line.

Examples of **block elements** are , <p>, <h1> to <h6>, <nav>, etc.
Examples of **inline elements** are <b>, <i>, , <img>, etc.

**NOTE:** You can also check which elements are block and which are inline by inspecting them using chrome dev tools.

## 5. TEXT FORMATTING TAGS

HTML provides us with the ability for formatting text just like we do it in MS Word or any text editing software.
The following HTML tags are used to format the appearance of the text on your web page. This can jazz up the look of the web page, *however,* too much variety in the text formatting can also look displeasing.HTML also defines special **elements** for defining text with a special **meaning**.
HTML uses elements like <b> and <i> for formatting output, like **bold** or *italic* text.
Formatting elements were designed to display special types of text:

- **<b>** - defines bold text

- **** - defines emphasized text
- **<i>** - defines italic text
- **<small>** - defines smaller text
- **** - defines important text
- **<sub>** - defines subscripted text
- **<sup>** - defines superscripted text
- **<u>** - defines underlined text
- **<ins>** - defines inserted text by underlying the text
- **<del>** - defines deleted text by striking through the text
- **<s>** - defines text that is no longer correct, accurate or relevant by striking through it
- **<mark>** - defines marked/highlighted text
- **<pre>** - defines preformatted text which is presented exactly as written in HTML
- **<tt>** - defines text appears as typed by a typewriter
- **<code>** - defines piece of computer code
- **<q>** - defines short quoted text
- **<cite>** - defines reference to a cited work
- **<abbr>** - defines an abbreviation or acronym
- **<var>** - defines a variable name
- **<kbd>** - defines keyboard input
- **<samp>** - defines sample output from a computer program

***EXTRA:***

***To know more about them, visit:***
***https://developer.mozilla.org/en-US/docs/Web/HTML/Element#Inline_text_semantics and https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Advanced_text_formatting***

# 6. SPECIAL CHARACTERS

In HTML, we have **some characters that are reserved**, e.g. less than (<) and greater than (>) signs, known as angle brackets , that are used to define a tag. Using them as symbols for the page, the browser could mistake them for markup.
While there are some characters that are not present on the keyboard.

These characters are called **special characters** or **HTML entity**, that either cannot be used or not available on the keyboard. So, to display these special characters, they must be replaced with the character entities.

An **HTML entity** is a piece of text that **begins with an ampersand (&)** and **ends with a semicolon(;)** and **between is the hex code or entity name**. These entities are used to display the reserved characters.

Eg., these are some html entities with how they will look on browser:
- some useful character entities - * *(single space), *<*(<), *'*('), *©*(©)
- diacritical marks - *à*(à), *Ô*(Ô)
- mathematical symbols - *∀* or *∀* for(∀), *∑* or *∑* for(∑)

- some other entities - *←*(←), *♥*(♥), *™*(™)

***EXTRA:***

***To know more about HTML entities, visit:***
***https://developer.mozilla.org/en-US/docs/Glossary/Entity***
***You can get all the available entities list here:***
***https://dev.w3.org/html5/html-author/charref***

# 7. TABLES

Tables are used to show the tabular data. To achieve this many tags are used. All the table-data is enclosed within the **<table>** tags.

A table is divided into rows (with the **<tr>** tag), and each row is divided into data cells (with the **<td>** tag). **tr** stands for table row, which represents the row of a table and **td** stands for table-data, which is the content of a data cell.

A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

Eg:

```
<table border=1>
<tr>
    <td>Row 1, cell 1</td>
    <td>Row 1, cell 2</td>
    <td>Row 1, cell 3</td>
</tr>
<tr>
    <td>Row 2, cell 1</td>
    <td>Row 2, cell 2</td>
    <td>Row 2, cell 3</td>
</tr>
<tr>
    <td>Row 3, cell 1</td>
    <td>Row 3, cell 2</td>
    <td>Row 3, cell 3</td>
</tr>
<tr>
    <td>Row 4, cell 1</td>
    <td>Row 4, cell 2</td>
    <td>Row 4, cell 3</td>
</tr>
</table>
```

The table will be seen something like this:

| | | |
|---|---|---|
| Row 1, cell 1 | Row 1, cell 2 | Row 1, cell 3 |
| Row 2, cell 1 | Row 2, cell 2 | Row 2, cell 3 |
| Row 3, cell 1 | Row 3, cell 2 | Row 3, cell 3 |
| Row 4, cell 1 | Row 4, cell 2 | Row 4, cell 3 |

## 7.1. border Attribute

The **border** attribute is used for mentioning the thickness of the borders. If you do not specify a border attribute the table will be displayed without any borders. Sometimes this can be useful, but most of the time, you want the borders to show.

## 7.2. Headings in a Table

If you want to add column names, then HTML provides a separate tag for that. Headings in a table are defined with the **<th>** tag.

Eg:

```
<table border=1>
    <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
    </tr>
    <tr>
        <td>Row 1, cell 1</td>
        <td>Row 1, cell 2</td>
        <td>Row 1, cell 3</td>
    </tr>
    <tr>
        <td>Row 2, cell 1</td>
        <td>Row 2, cell 2</td>
        <td>Row 2, cell 3</td>
    </tr>
</table>
```

The table will be seen like this

| Column 1 | Column 2 | Column 3 |
|---|---|---|
| Row 1, cell 1 | Row 1, cell 2 | Row 1, cell 3 |
| Row 2, cell 1 | Row 2, cell 2 | Row 2, cell 3 |

## 7.3. <thead>, <tbody>, <tfoot>

The **<thead>** tag is used to group header content in an HTML table.
The **<tbody>** tag is used to group the body content in an HTML table.
The **<tfoot>** tag is used to group footer content in an HTML table.

These are the semantic tags that not only provide meaning to the elements but also have some other useful functionality as well.
Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans

multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.

Eg:

```
<table border="1">
    <thead>
        <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3 </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Row 1, cell 1</td>
            <td>Row 1, cell 2</td>
            <td>Row 1, cell 3</td>
        </tr>
        <tr>
            <td>Row 2, cell 1</td>
            <td>Row 2, cell 2</td>
            <td>Row 2, cell 3</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3 </th>
        </tr>
    </tfoot>
</table>
```

The table now looks like:

| Column 1 | Column 2 | Column 3 |
|---|---|---|
| Row 1, cell 1 | Row 1, cell 2 | Row 1, cell 3 |
| Row 2, cell 1 | Row 2, cell 2 | Row 2, cell 3 |
| Column 1 | Column 2 | Column 3 |

## 7.4. caption Tag

The <caption> tag defines a table caption.
The <caption> tag must be inserted immediately after the <table> tag.

Eg: If you add `<caption>Table Example</caption>` just after the <table> tag, the table will now look like this:

Table Example

| Column 1 | Column 2 | Column 3 |
|---|---|---|
| Row 1, cell 1 | Row 1, cell 2 | Row 1, cell 3 |
| Row 2, cell 1 | Row 2, cell 2 | Row 2, cell 3 |
| Column 1 | Column 2 | Column 3 |

***NOTE:*** *You can specify only one caption per table.*

## 7.5. colspan and rowspan Attribute

To manage the layout of the the tables, two attributes are used, **rowspan** and **colspan**.

Attribute **rowspan** is used to mention the number of rows that a particular cell will be occupying. Attribute **colspan** is used to mention the number of columns that a particular cell will be occupying.
They both are used with the **td** tag and can also be used with the **th** tag.

Eg: adding attributes colspan and rowspan to the table

```
<table border="1">
    <thead>
        <tr>
            <th>Column 1</th>
            <th colspan="2">Column 2 and 3 heading</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Row 1, cell 1</td>
            <td colspan="2">(Row 1, cell 2) and (Row 1, cell 3)</td>
        </tr>
        <tr>
            <td rowspan="2">(Row 2, cell 1) and (Row 3, cell 2)</td>
            <td>Row 2, cell 2</td>
            <td>Row 2, cell 3</td>
        </tr>
        <tr>
            <td>Row 3, cell 2</td>
            <td>Row 3, cell 3</td>
        </tr>
    </tbody>
</table>
```

The table now looks like:

| Column 1 | Column 2 and 3 | |
|---|---|---|
| Row 1, cell 1 | (Row 1, cell 2) and (Row 1, cell 3) | |
| (Row 2, cell 1) and (Row 3, cell 2) | Row 2, cell 2 | Row 2, cell 3 |
| | Row 3, cell 2 | Row 3, cell 3 |