Block & inline in HTML

Block element 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 that only take up as much width as much needed to display the content of the element, thereby allowing other element to be in line with the inline element.

- Block element always start on a new line.
- Inline element do not start from a new line.

Some block elements are <div>, <p>, <h1> to <h6>, <nav>, etc.

Some inline elements are <b>, <i>, <span>, <img>, etc.

div tag:

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

The <div> 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 <div> element.

Example:

```
<p> Hello
   <div>
      Welcome to India!
   </div>
</p>
```

Browser:

```
Hello

Welcome to India!
```

Adding style to div

Adding background color to an element

```
<p> Hello
   <div style="back-ground color:cyan">
   </div>
</p>.
```

Span tag

- <span> element is an inline container used to mark up a part of a text or a part of a document.
- The <span> element has no required attributes, but style, class and id are common.
- When used together with CSS, the <span> element can be used to style parts of the text.
- It does not shift the content to the next but helps to style a specific part of the content.

Class and Id in HTML

- Classes and id are selectors in HTML
- They are attributes to any tags.
- Majorly used for styling the parts of your content.
- Multiple HTML elements can also share the class id.
- The class name and id name are case sensitive.

Internal styling

Earlier, we covered the inline type of styling the element by using the style attribute, but styling can also be internally in the HTML file by using an <style> element in the <head> section.

Example:

```
    <head>
     <style>
      P{ color:red
       }
     </style>
    </head>
   <body>
    <p>Hello </p>
    </body>
```

Output:

Hello

Class Attribute

The HTML class attribute is used to specify a single or multiple class name for an HTML element. The class name can be used by CSS and Javascript to do some task for HTML element. You can use this class in CSS with a specific class, write a period (.) character, followed by the name of the class for selecting elements.

Multiple cases

HTML elements can belong to more than one class, or you can say multiple classes can be given to an element. To define multiple classes, separate the class names with a space.

Id attribute

- The HTML id attribute is used to specify a unique id for an HTML element.
- You cannot have more than one element with the same id in an HTML document.
- Javascript also uses it to access and manipulate the element with the specific id.

Syntax: Write a Hash character (#), followed by an id name Then define the CSS property using curly brackets.

Example:

```
<head>
<style>
   #one{color: blue;}
    #two{ color: Red;}
    # three{ color: black}
</style>
</head>
<body>
<p id="one"> Hellow!.</p>
<p id="two"> Hey! </p>
<p id="three"> Good Morning:)</p>
</body>
```

Browser

```
Hellow!
Hey!
Good Morning :)
```

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 jumps as an alternative mode of transport around your site. Basically, page jumps 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 specific part of the same document, i.e. internal links.

Adding Script to HTML

To use Javascript in your web page, we need to insert
it into your HTML page.
We can write our Jawascript code by using the
<script> tag. you then need to write Jawascript code in
between them.

Example:-

```
<script type ="text/javascript">
document.getElement By Id("demo").inner.HTML ="My First Javascript";
</script>
```

you can add a type attribute to mention the type of script you
are written in
are using. But since default scripts
Javascripts.

Adding External scripts
<script> tag can be used in another manner as well. It can
add external javascript files to the web page.
writing Javascript code in external files separates it
from HTML code. It makes HTML and Javascript easier to
read and maintain.
The external javascript file should have the extension -'.js'

Example

```
<script type ="text/javascript" src="myscript.js"></script>
```

The Javascript file name with the extension is
mentioned inside the src attribute, i.e. file name is
myscript.js.
External scripts can be referenced with a full absolute
URL or with a path relative to the current web page.

Javascript in <head> and <body> tag
you place scripts inside the <head> tag, just like the <link>
tag. you can use both of the ways mentioned above
to add the script to the web page by writing them
inside the <head> tag.

Example:-

```
<head>
  <!-- other header tag --!>
  <script type ="text/ Javascript" src= "my script.js"></script>
</head>
```

But when you use the above two methods, the Javascript compilation is done first, even before the HTML code is rendered on the web page. This slows down the loading time of the web page, and also, some things might not be as expected as element are not rendered at that time.

To improve the web page's loading time, we can also load and compile the Javascript after the Page is loaded. To do this, we need to add the script at the bottom of the <body> tag after the HTML code like -

Example

```
<body>
   ---
   <!-- HTML code -->
   ---
   <script type="text/Javascript" src="my script.js"></script>
</body>.
```

Now, the html code is rendered first, and then after that, Javascript is loaded.

Internal Javascript

Rather than making another Javascript file and attaching it to the HTML file, you could also write the Javascript code within the HTML file. as we did for the CSS.

But this method is not recommeded much. It could be done if there are some lines of Javascript code.

External Javascript files help us to reuse them in multiple HTML files.

Example.

```
<body>
  <h1 id="heading">Hello!</h1>
  <script type="text/Javascript">
  document.getElementById("heading").innerHTML="Bye
   Javascript!";
  </script>
  </Body>
```

Browser:

Bye JavaScript!

<noscript> tag

The HTML <noscript> tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts:

Example:

```
<script type="text/javascript">
document.getElementById("heading").innerHTML = "Bye
 Javascript!";
 </script>
<noscript>Sorry, your Browser does not support
          Javascript! </noscript>.
```

Browser

```
Hello World!
Sorry, your browser does not support Javascript!
```

HTML Vs XHTML

XHTML

- It stands for Extensible Hypertext Markup Language.
- It can be consider as a part of ~~HTML~~ XML markup language because ~~of~~ XHTML have features of both XML and HTML. XHTML is extended from XML and HTML. It can be considered as best version of HTML.
- It is a stricter, more XML-based versiong HTML.

HTML

HTML is Hyper Text Markup Language. It is used to create web pages and link them from one to another. It is a markup language.

Differences for HTML

- <!DOCTYPE> is mandatory.
- The ~~xhmns~~ xmlns in <html> is mandatory.
- <html>, <head>, <title>, and <body> are mandatory.
- Element must always be properly nested.
- Element must always be closed.
- Element must always be in lowercase.
- Attributes names must always be in lowercase.
- Attribute value must always be quoted.
- Attribute minimization is forbidden.