

Introduction to HTML Assignment by Kanchan Sinha

1. **Block:** A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can), like `div` tag, `p` tag, etc by default is block type.

Inline: An inline element does not start on a new line and only takes up as much width as necessary, like `span` tag by default is inline

`` element is used as an inline element Other inline elements are:

- anchor `<a>` tag
- emphasis `` tag
- image `` tag

Other block-level tags are:

- Heading tags `<h1>` to `<h6>`
- List (Ordered, Unordered, Description and List Item) tags `` , `` , ``

2.

display: none- doesn't preserve the space. There will be no space allocated for it between the other tags.

visibility: hidden- means that unlike `display: none`, the tag is not visible, but space is allocated for it on the page. The tag is rendered, it just isn't seen on the page.

3.

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The float property can have one of the following values:

- left - The element floats to the left of its container
- right - The element floats to the right of its container
- none - The element does not float (will be displayed just where it occurs in the text). This is default
- inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images.

The clear property specifies what elements can float beside the cleared element and on which side.

The clear property can have one of the following values:

- none - Allows floating elements on both sides. This is default
- left - No floating elements allowed on the left side
- right - No floating elements allowed on the right side
- both - No floating elements allowed on either the left or the right side
- inherit - The element inherits the clear value of its parent

4.

Static. This is the default for every single page element. Different elements don't have different default values for positioning, they all start out as static. Static doesn't mean much; it just means that the element will flow into the page as it normally would.

- **Relative.** This type of positioning is probably the most confusing and misused. What it really means is "relative to itself". If you set position: relative; on an element but no other positioning attributes (top, left, bottom or right), it will no effect on it's positioning at all, it will be exactly as it would be if you left it as position: static; But if you do give it some other positioning attribute, say, top: 10px;, it will shift its position 10 pixels down from where it would normally be.

- **Absolute.** We can use the positioning attributes top, left, bottom. and right to set the location.

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

- **Fixed.** A fixed position element is positioned relative to the viewport, or the browser window itself.

5.Screenshot of Output of this question

Table For Experiment			
ID	Employee Name	Designation	
1	TTN	Lohit	JVM
2	TTN	Aditya	Python
3	TTN	Rohit	Android
4	TTN	Nikhil	JVM
5	TTN	Gagan	JVM
6	TTN	Arpit	IOS

6.

Metadata is data (information) about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

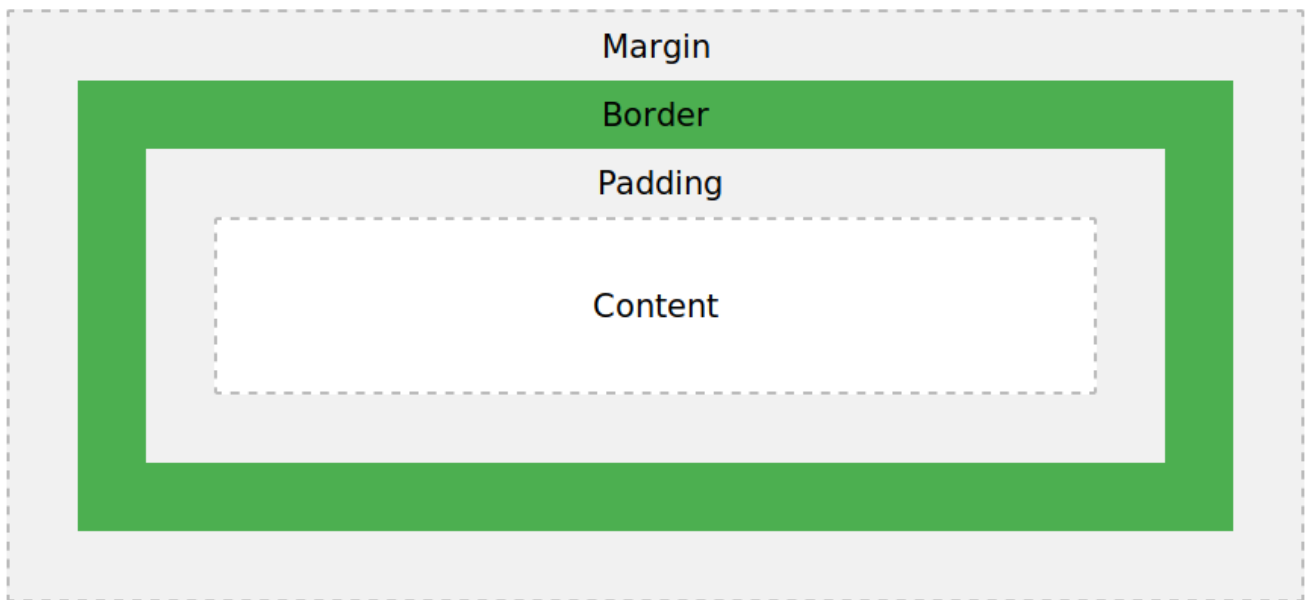
Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

7.

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

8.

The element Selector

The element selector selects elements based on the element name.

```
p {  
text-align: center;  
color: red;  
}
```

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element should be unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
#para1 {  
text-align: center;
```

```
color: red;
}
```

The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

```
.center {
text-align: center;
color: red;
}
```

The Universal Selector

universal selectors select any type of elements in an HTML page. It matches a single element. An asterisk (i.e. "*") is used to denote a **CSS universal selector**. An asterisk can also be followed by a selector. This is useful when you want to set a style for of all the elements of an HTML page or for all of the elements within an element of an HTML page.

```
*{
color : blue;
}
```

9.

Doctype:

The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html>

tag.

The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what

version of HTML the page is written in.

10.

<section>

The <section> element defines a section in a document.

According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with

a heading."

<section>

<h1>TTN Bootcamp</h1>

<p>Bootcamp for trainees</p>

</section>

<article>

The <article> element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to read it independently from the rest

of the web site.

Examples of where an <article> element can be used:

- Forum post

- Blog post

- Newspaper article

<article>

<h1>TTN Bootcamp</h1>

<p>Bootcamp for trainees</p>

</article>

<header>

The <header> element specifies a header for a document or section.

The <header> element should be used as a container for introductory content.

<article>

<header>

<h1>TTN ASSIGNMENT</h1>

</header>

<p>Assignment</p>

</article>

<footer>

The <footer> element specifies a footer for a document or section.

A <footer> element should contain information about its containing element.

A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.

<footer>

<p>Posted By:Prateek</p>

</footer>

<nav>

The <nav> element defines a set of navigation links.

<nav>

HTML

CSS

</nav>