Introduction to HTML/CSS

Q1 How are inline and block elements different from each other?

An Inline element doesn’t start on a new line. The space used by the inline element is opening and closing tag. Examples of inline elements are anchor tag, image tag, emphasis tag, span tag, etc. Inline elements cannot take top or bottom margins, width or height properties.

Block-level elements always start on a new line. From left to right, it takes the full width of a parent container. It has a line break before and after the element. Examples of block-level elements are heading tags, ordered list, unordered list, list and description items, blockquote tag, pre-formatted text tag, div tag, etc. Block-level elements can contain inline elements as well as other block-level elements. With the display property of CSS, block-level elements can be converted into inline and inline elements can be converted into block-level.

**Block level elements**

A block-level element always starts on a new line.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

A block level element has a top and a bottom margin, whereas an inline element does not.

example:-<address> <article><aside><div><dl><dt><fieldset><figure><footer><form><h1>-<h6><header><hr><li><main><ol><p><pre>[<se](https://www.w3schools.com/tags/tag_section.asp)ctio[n>](https://www.w3schools.com/tags/tag_section.asp)<table><tfoot><ul>

**Inline level elements**

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

This is a <span> element inside a paragraph.

examples:-<a><big><br><button><cite><code><img><input><label><map><object><output><q><samp><script><select><small><span><strong><sub><sup><textarea><time><tt><var>

Q2 Explain the difference between visibility:hidden and display:none

Both of the property is quite useful in CSS. The visibility: “hidden”; property is used to specify whether an element is visible or not in a web document but the hidden elements take up space in the web document. The visibility is a property in CSS that specifies the visibility behavior of an element and display: “none” property is used to specify whether an element is exist or not on the website.

Syntax:-

Visibility property:

visibility: visible| hidden | collapse | initial | inherit;

Display property:

display: none | inline | block | inline-block;

So, the difference between display: “none”; and visibility: “hidden”;, right from the name itself we can tell the difference as display: “none”;, completely gets rids of the tag, as it had never exists in the HTML page whereas visibility: “hidden”;, just makes the tag invisible it will still be on the HTML page occupying space it’s just invisible.

Q3 Explain the clear and float properties.

Clear properties:

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

The most common way to use the clear property is after you have used a float property on an element.

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

The following example clears the float to the left. Means that no floating elements are allowed on the left side (of the div):

Examples:-

div {

clear: left;

}

Float property

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.

Q4 explain difference between absolute, relative,fixed and static.

The CSS [position](https://www.w3schools.com/cssref/pr_class_position.asp) property defines, as the name says, how the element is positioned on the web page.

If you are interested in reading about the font properties, articles about the [relative font size](https://dzone.com/articles/css-relative-font-size) and [CSS columns](https://dzone.com/articles/css-columns) might be of interest.

So, there are several types of positioning: static, relative, absolute, fixed, sticky, initial, and inherit. First of all, let's explain what all of these types mean.

* Static - this is the default value, all elements are in order as they appear in the document.
* Relative - the element is positioned relative to its normal position.
* Absolute - the element is positioned absolutely to its first positioned parent.
* Fixed - the element is positioned related to the browser window.
* Sticky - the element is positioned based on the user's scroll position.

Now that we have explained the basics, we will talk more about the two most commonly used position values - relative and absolute.

## What Is Relative Positioning?

When you set the position *relative to an element*, without adding any other positioning attributes (top, bottom, right, left) nothing will happen. When you add an additional position, such as left: 20px the element will move 20px to the right from its normal position. Here, you can see that this element is relative to itself. When the element moves, no other element on the layout will be affected.

*There is a thing you should keep in mind while setting position - relative to an element limits the scope of absolutely positioned child elements. This means that any element that is the child of this element can be absolutely positioned within this block.*

After this brief explanation, we need to back it up, by showing an example.

In this example, you will see how the relatively positioned element moves when its attributes are changed. The first element moves to the left and top from its normal position, while the second element stays in the same place because none of the additional positioning attributes were changed

## What Is Absolute Positioning?

This type of positioning allows you to place your element precisely where you want it.

The positioning is done relative to the first relatively (or absolutely) positioned parent element. In the case when there is no positioned parent element, it will be positioned related directly to the HTML element (the page itself).

An important thing to keep in mind while using absolute positioning is to make sure it is not overused, otherwise, it can lead to a maintenance nightmare.

The next thing, yet again, is to show an example.

In the example, the parent element has the position set to relative. Now, when you set the position of the child element to absolute, any additional positioning will be done relative to the parent element. The child element moves relative to the top of the parent element by 100px and right of the parent element by 40px.

Q5 Write the HTML code to create a table in which there are 4 columns( ID , Employee Name, Designation, Department) and at least 6 rows. Also do some styling to it.

<body>

<style>

table,th,td{

border: 1px solid black;

text-align: center;

}

</style>

<table style="width: 100%" >

<thead>

<tr>

<th>id</th>

<th>Employee Name</th>

<th>Designation</th>

<th>Department</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Akash</td>

<td>trainee</td>

<td>jvm</td>

</tr>

<tr>

<td>2</td>

<td>mayank</td>

<td>Software Enginear</td>

<td>jvm</td>

</tr>

<tr>

<td>3</td>

<td>sharma</td>

<td>trainee</td>

<td>jvm</td>

</tr>

<tr>

<td>5</td>

<td>rohit</td>

<td>trainee</td>

<td>jvm</td>

</tr>

<tr>

<td>6</td>

<td>manoj</td>

<td>trainee</td>

<td>jvm</td>

</tr>

<tr>

<td>7</td>

<td>Akash</td>

<td>trainee</td>

<td>jvm</td>

</tr>

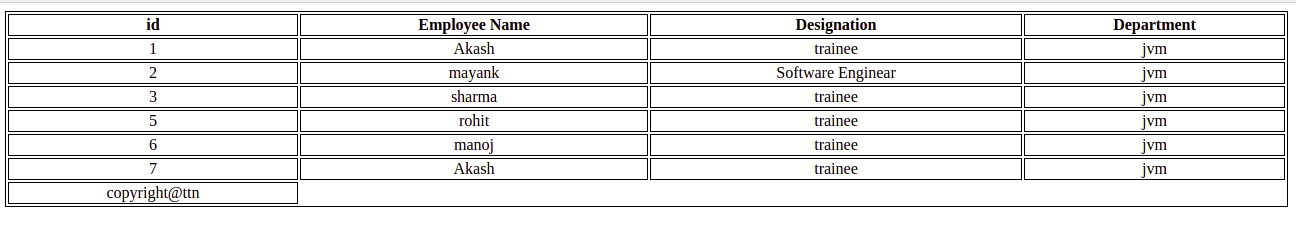
</tbody>

<tfoot>

<tr><td>copyright@ttn</td></tr>

</tfoot>

</table>



Q6 Why do we use meta tags?

The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.

<meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Metadata will not be displayed on the page, but is machine parsable.

Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

Q7 Explain box model.

CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.

Box-Model has multiple properties in CSS. Some of them are given below:

* borders
* margins
* padding
* Content

Border Area: It is the area between the box’s padding and margin. Its dimensions are given by the width and height of border.

Margin Area: This area consists of space between border and margin. The dimensions of Margin area are the margin-box width and the margin-box height. It is useful to separate the element from its neighbors.

Padding Area: It includes the element’s padding. This area is actually the space around the content area and within the border box. Its dimensions are given by the width of the padding-box and the height of the padding-box.

Content Area: This area consists of content like text, image, or other media content. It is bounded by the content edge and its dimensions are given by content box width and height.

Q8 What are the different types of CSS Selectors?

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

* Simple selectors (select elements based on name, id, class)
* Combinator selectors (select elements based on a specific relationship between them)
* Pseudo-class selectors (select elements based on a certain state)
* Pseudo-elements selectors (select and style a part of an element)
* Attribute selectors (select elements based on an attribute or attribute value)

Q9 Define Doctype.

Doctype: A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. The Doctype is not an element or tag, it lets the browser know about the version of or standard of HTML or any other markup language that is being used in the document.

Declaration of a Doctype: A DOCTYPE declaration appears at the top of a web page before all other elements. According to the HTML specification or standards, every HTML document requires a document type declaration to ensure that the pages are displayed in the way they are intended to be displayed.

The DOCTYPE for HTML5 is case-insensitive and can be written as shown below:

< !DOCTYPE html >

Doctype Usage: In the version, HTML 4.01, the usage of DOCTYPE declaration was to create a reference to a document type definition (DTD), since the version HTML 4.01 was completely based on a Standard Generalized Markup Language(SGML).

The document type definition (DTD) is responsible for specifying the rules for the Standard Generalized Markup Language(SGML) so that the browser processes the content correctly. But in the HTML version, HTML 5 there isn’t any need for a reference to a document type definition (DTD) because HTML 5 is not based on a Standard Generalized Markup Language(SGML).

In HTML 5, the DOCTYPE declaration is only required for enabling the standard mode for writing documents.

Below is a sample HTML program with doctype declaration:

Q10 Explain 5 HTML5 semantic tags.

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.

Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.

## **HTML <section> Element**

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

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

A web page could normally be split into sections for introduction, content, and contact information.

## **HTML <article> Element**

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

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

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

* Forum post
* Blog post
* Newspaper article

**The <header> element**

represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

* one or more heading elements (<h1> - <h6>)
* logo or icon
* authorship information

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

## **HTML <nav> Element**

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

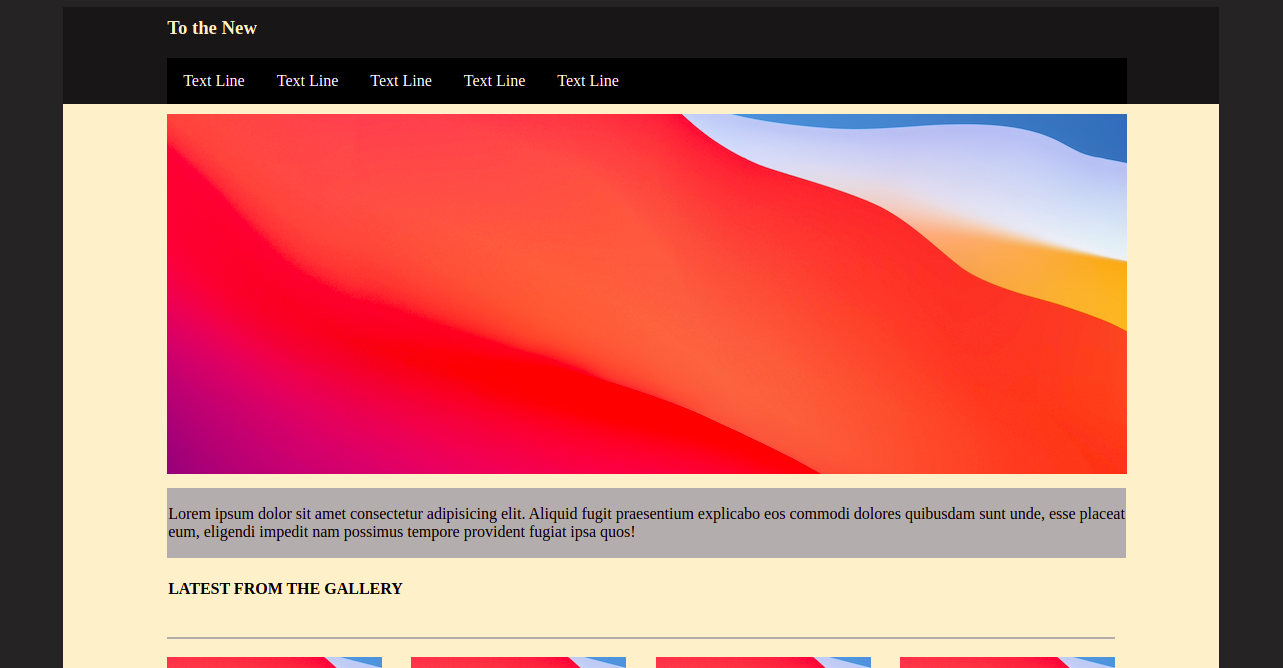
Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

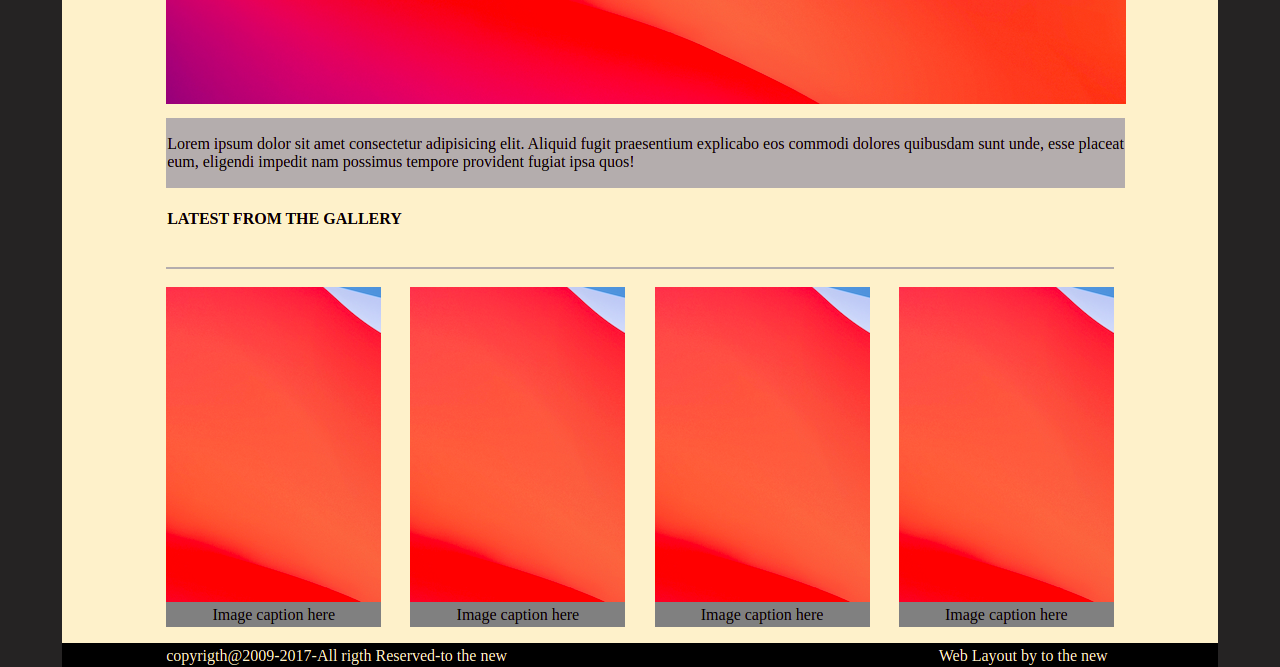
Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

## HTML <aside> Element

The <aside> element defines some content aside from the content it is placed in (like a sidebar).

The <aside> content should be indirectly related to the surrounding content.

Q11 Create HTML for web-page.jpg (check resources, highest weightage for answers)



Q12 Create HTML for form.png (check resources, highest weightage for answers)

