

# HTML/CSS ASSIGNMENT

1.How are inline and block elements different from each other?

Solution : -

## Inline Elements

Basically, an inline element does not cause a line break (start on a new line) and does not take up the full width of a page, only the space bounded by its opening and closing tag. It is usually used within other HTML elements.

Other examples of inline elements are:

- anchor <a> tag
- span tag <span>
- image <img> tag

```
<!DOCTYPE html>
<html>
<body>

<p>This is an inline span <span style="border: 1px solid black">SIDDHANT SHARMA</span> and his ID is 4143.</p>

<p>The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.</p>

</body>
</html>
```

This is an inline span SIDDHANT SHARMA and his ID is 4143.

The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.

## Block Elements

A block-level element always starts on a new line and takes up the full width of a page, from left to right. A block-level element can take up one line or multiple lines and has a line break before and after the element.

Other examples of the block-level tag are:

- Heading tags <h1> to <h6>
- List (Ordered, Unordered, Description and List Item) tags <ol> , <ul> ,<dl> , <li>
- Division <div>

```
<!DOCTYPE html>
<html>
<body>

<div style="border: 1px solid black">SIDDHANT SHARMA</div>

<p>The DIV element is a block element, and will always start on a new line and take up the full width available.</p>

</body>
</html>
```

SIDDHANT SHARMA

The DIV element is a block element, and will always start on a new line and take up the full width available.

## 2. Explain the difference between visibility:hidden and display:none

Solution : -

### **display : none;**

display : none means that the tag in question will not appear on the page at all (although you can still interact with it through the dom). There will be no space allocated for it between the other tags.

### **visibility : hidden;**

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.

EXMAPLE:-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>
```

```
    Display:"none" and visibility: "hidden";
```

```
</title>
```

```
</head>
```

```
<body><h1 style="color:yellow;">
```

```
    DISPLAY AND VISIBILITY
```

```
</h1>
```

```
<h3>
```

```
In this section we are using display:none in a class;
```

```
</h3>
```

```
<div class="display">
```

```
<b>
```

```
MY NAME IS<span style="display:none">SIDDHANT SHARMA</span> AND ID is 4143.
```

```
</b> </div>
```

```
<br>
```

```
<div class="visibility">
```

```
<b>
```

```
In this section we have used visibility:hidden
```

```
<span style="visibility:hidden"> SIDDHANT SHARMA</span> as we cannot see my name in the blank area.
```

```
</b>
```

```
</div>
```

```
<p>You can see that there is no name after is don't have any blank space and visibility: "hidden": has the blank space. </p>
```

```
</body>
```

```
</html>
```

# DISPLAY AND VISIBILITY

**In this section we are using display:none in a class;**

**MY NAME IS AND ID is 4143.**

**In this section we have used visibility:hidden  
as we cannot see my name in the blank area.**

You can see that there is no name after is and don't have any blank space and after visibility: "hidden": has the blank space.

3. Explain the clear and float properties.

Solution : -

## **Float Properties**

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.

## **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.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  overflow:auto;
  border: 3px solid #4CAF50;
  padding: 5px;
}
.para{
  clear: left;
}
.img2 {
  float:right;
}
</style>
</head>
<body>
<div>
  <p class="para">
    This is a image of a pineapple in a farm</p>
</div>
</body>
</html>
```

This is a image of a pineapple in a farm



4.Explain the difference between absolute, relative, fixed and static.

Solution : -

**Absolute:-**

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.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.absolute {
  position: absolute;
  top: 80px;
  right: 50px;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>Absolute</h2>
<div class="absolute">This div element has position: absolute</div>
</div>
</body>
</html>
```

## Absolute

This div element has  
position: absolute

## Relative;

An element with `position: relative;` is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>Relative</h2>

<p>An element with position: relative; is positioned relative to its
normal position:</p>

<div class="relative">
This div element has position: relative
</div>

</body>
</html>
```

### Relative

An element with `position: relative;` is positioned relative to its normal position:

This div element has position: relative

## Fixed;

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>Fixed</h2>

<p>An element with position: fixed; is positioned relative to the
viewport, which means it always stays in the same place even if the
page is scrolled:</p>
<div class="fixed">
This div element has position: fixed
</div>

</body>
</html>
```

### Fixed

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

This div element has position: fixed

## Static;

HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties. An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>Static</h2>

<p>An element with position: static; is not positioned in any
special way; it is
always positioned according to the normal flow of the page:</p>

<div class="static">
  This div element has position: static
</div>

</body>
</html>
```

### Static

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This div element has position: static

5. 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.

Solution :

LINK:- [https://github.com/siddhant1098/TTN\\_Assignments/tree/session\\_6\\_HMTL/CSS/html](https://github.com/siddhant1098/TTN_Assignments/tree/session_6_HMTL/CSS/html)

ID	Employee Name	Designation	Department
4143	Siddhant	Big Data	Group 4
4030	Devansh	MEAN	Group 3
4020	Bharat	Android	Group 1
4122	Jay	Dev Op's	Group 5
4121	Saket	IOS	Group 4
4021	Bhupesh	FEEN	Group 3
4023	Vipul	FEEN	Group 3

## 6. Why do we use meta tags?

### Solution : - **Meta tags**

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. Also, it helps to improve the SEO (Search Engine Optimization) of a web page by using certain keywords related to the web page. <meta> tags always go inside the <head> element.

Following are few examples of <meta> tag with different attributes :

1. <meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">  
**(For Search engines)**
2. <meta name="description" content="Free Web tutorials on HTML and CSS">  
**( Description of web page)**
3. <meta name="author" content="John Doe"> **(Author of web page)**
4. <meta name="viewport" content="width=device-width, initial-scale=1.0">  
**(Responsiveness)**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
</head>
<body>

<p>All meta information goes in the head section...</p>

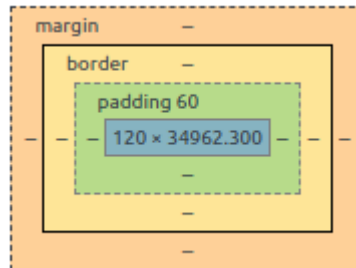
</body>
</html>
```

All meta information goes in the head section...

## 7. Explain box model.

### Solution : - **Box Model**

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.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 400px;
  border: 15px solid green;
  padding: 50px;
  margin: 10px;
}
</style>
</head>
<body>

<h2> Box Model</h2>

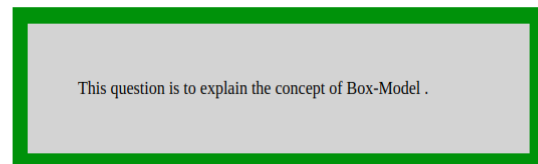
<p>The CSS box model is essentially a box that wraps around every
HTML element. It consists of: borders, padding, margins, and the
actual content.</p>

<div>This question is to explain the concept of Box-Model .</div>

</body>
</html>
```

### Box Model

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





## 8. What are the different types of CSS Selectors?

Solution : -

### **CSS Selectors**

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

- **Element Selector** :The element selector selects HTML elements based on the element name.
- **Id Selector** :The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is 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.
- **Class Selector** :The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.
- **Universal Selector** :The universal selector (\*) selects all HTML elements on the page.

## 9. Define Doctype.

Solution : - A doctype or document type declaration is an instruction which 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.

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

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document.....
</body>

</html>
```

The content of the document.....

## 10. Explain 5 HTML5 semantic tags.

Solution : - HTML5 is the latest version of HTML and it introduces many new semantic <tags>.

Some of the commonly used new semantic tags of HTML5 are :

- <article> - Defines an article in the document. 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.

---

```
<article>
  <h1>What Does WWF Do?</h1>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

- <aside> - Defines content aside from the page content like sidebar. The <aside> element defines some content aside from the content it is placed in (like a sidebar). The <aside> content should be related to the surrounding content.

```
<p>My family and I visited The Epcot center this summer.</p>

<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
```

- <header> - Defines a header for the document or a section. The <header> element should be used as a container for introductory content. You can have several <header> elements in one document.

---

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

---

- <nav> - Defines navigation links in the document. The <nav> element defines a set of navigation links.

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

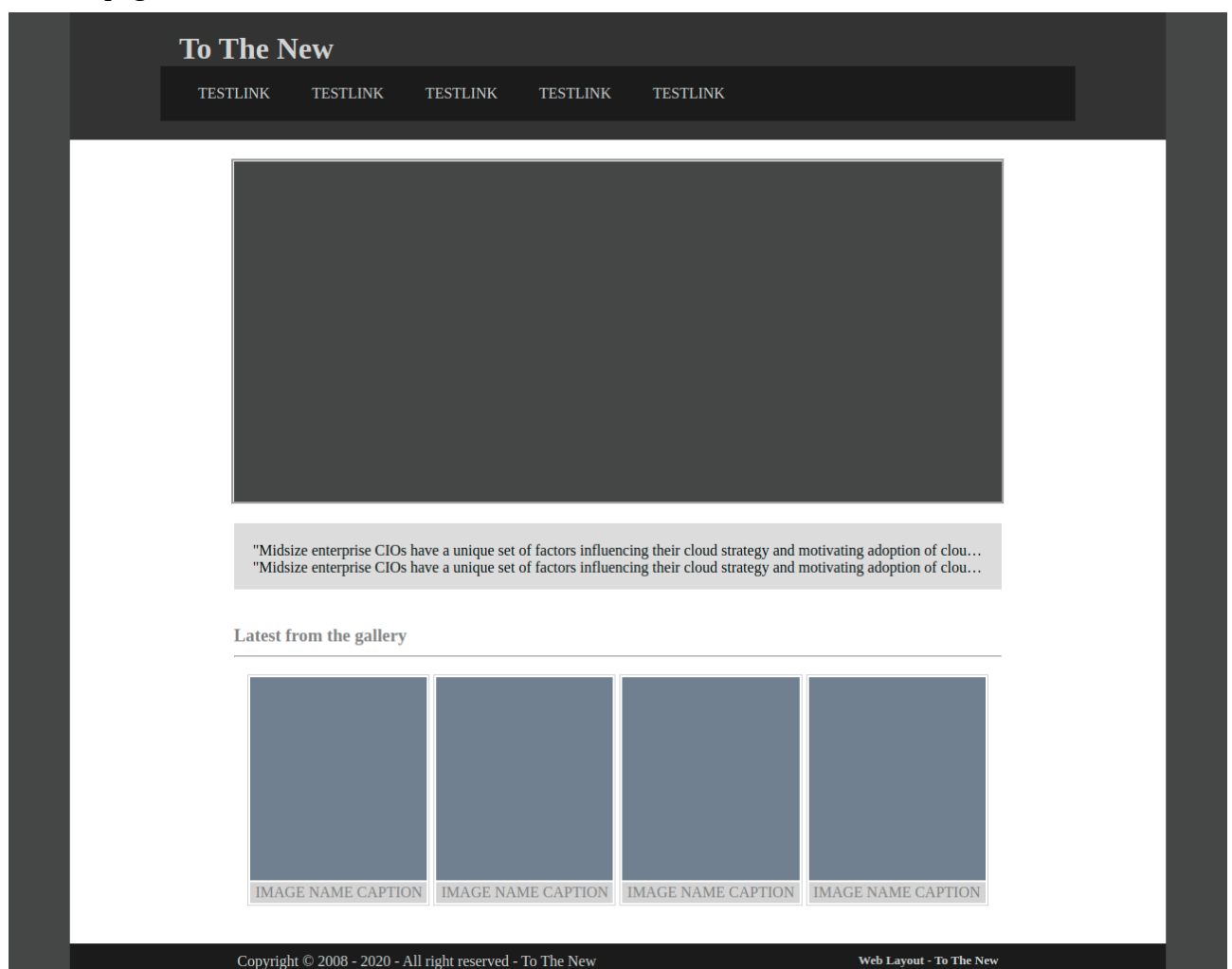
- <section> - Defines a section in the document

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

## 11. Create HTML for web-page.jpg

Solution : -

LINK:-[https://github.com/siddhant1098/TTN\\_Assignments/blob/session\\_6\\_HMTL/CSS/html/Q11\\_Web-page%20.html](https://github.com/siddhant1098/TTN_Assignments/blob/session_6_HMTL/CSS/html/Q11_Web-page%20.html)



## 12. Create HTML for form.png

Solution : -

LINK:-

[https://github.com/siddhant1098/TTN\\_Assignments/blob/session\\_6\\_HMTL/CSS/html/Q12\\_Form.html](https://github.com/siddhant1098/TTN_Assignments/blob/session_6_HMTL/CSS/html/Q12_Form.html)

[To The New](#)

[Home](#) [Quick Help](#)

### Bug Report

**Title\***

**Description\***

Write something..

**Security:**

Critical

**Product\***

Formoid

**Verification\***

**License\***

☒ Free ☐ Business

**Security:**

Critical

**Attachments\***

No file chosen

Choose file

Send