

Exercise

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

BLOCK ELEMENTS	INLINE ELEMENTS
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).	An inline element does not start on a new line and only takes up as much width as necessary.
The <div> element is a block-level element.	This is an inline element inside a paragraph.

The display Property

display: inline

INLINEEEEEEEEEEEEE

display: inline-block

BLOCKKKKKK

display: block

BLOCK STILL

BOX BLOCK

BOX BLOCK

BLOCK DISPLAY

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

visibility:hidden	display: "none"
1. 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	1. display: "none" property is used to specify whether an element is exist or not on the website.
2. Syntax: <ul style="list-style-type: none">• Visibility property: visibility: visible hidden collapse initial inherit;	2. Syntax: <ul style="list-style-type: none">• Display property: display: none inline block inline-block;
3. display: "none"; completely gets rids of the tag, as it had never exists in the HTML page	3. visibility: "hidden";, just makes the tag invisible it will still be on the HTML page occupying space it's just invisible.

AshitaKumar

display:"none"; and visibility: "hidden";

display: "none";

visibility: "hidden";

You can see that the display: "none"; don't have any blank space and visibility: "hidden": has the blank space.

3. Explain the clear and float properties.

CLEAR	FLOAT
1. The CSS clear property specifies what elements can float beside the cleared element and on which side.	1. The CSS float property specifies how an element should float.
2. The clear property specifies what elements can float beside the cleared element and on which side.	2. The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.
<p>3. The clear property can have one of the following values:</p> <ul style="list-style-type: none">• 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 <p>The most common way to use the clear property is after you have used a float property on an element.</p>	<p>3. The float property can have one of the following values:</p> <ul style="list-style-type: none">• 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 <p>In its simplest use, the float property can be used to wrap text around images.</p>

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

The position property specifies the type of positioning method used for an element.

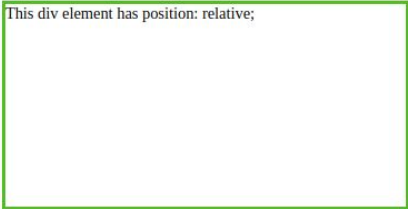
There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

ABSOLUTE	RELATIVE	FIXED	STATIC
1. An element with position : absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).	1. An element with position: relative is positioned relative to its normal position.	1. 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.	1. Static positioned elements are not affected by the top, bottom, left, and right properties. Static positioned elements are not affected by the top, bottom, left, and right properties.
2. If an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.	2. 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. This <div> element has position: relative;	2. A fixed element does not leave a gap in the page where it would normally have been located. The fixed element in the lower-right corner of the page.	2. 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;

position: relative;

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

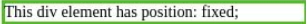
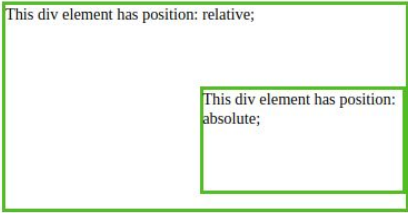


position: fixed;

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:

position: absolute;

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



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.

Employee Table

ID	Employee Name	Designation	Department
101	Ashita Kumar	Trainee	BigData
102	Smriti Vohra	Analyst	Business
103	Neraj Natarajan	Engineer	Automobile
104	Abha Rana	Consultant	Engineer
105	Arushi Tandon	Technical Lead	IT
106	Rashika Sharma	Trainee	Sales

6. Why do we use meta tags?

- 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.

EXAMPLE : You should include the following <meta> viewport element in all your web pages:

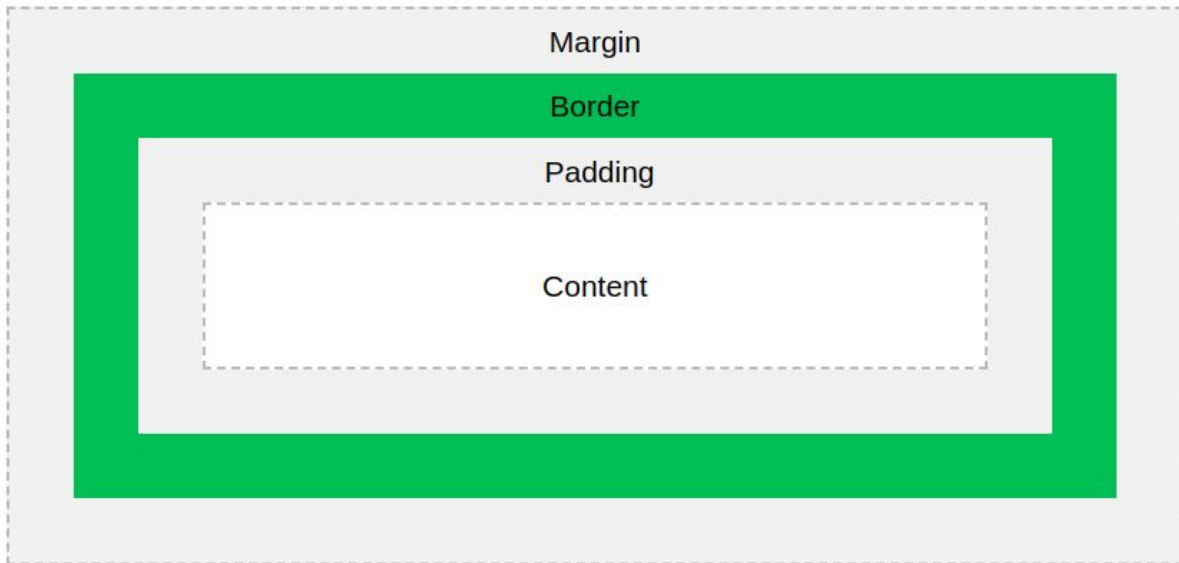
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
home > ashita > Downloads > ques6.html > html
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Meta tags example</title>
5      <meta name = "keywords used "
6        |         |         | content = "Meta Tags, Metadata" />
7      <meta name = "description about"
8        |         |         | content = "Meta tags learning." />
9      <meta name = "author" content = "Ashita Kumar" />
10   </head>
11
12   <body>
13     <p>Hello World!</p>
14   </body>
15 </html>
```

Hello World!

7. Explain box model.

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:



- 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. What are the different types of CSS Selectors?

A CSS selector is the part of a CSS rule set that actually selects the content you want to style. Let's look at all the different kinds of selectors available, with a brief description of each.

Universal Selector

The *universal selector* works like a wildcard character, selecting all elements on a page. Every HTML page is built on content placed within HTML tags. Each set of tags represents an element on the page.

Element Type Selector

Also referred to simply as a "type selector," this selector must match one or more HTML elements of the same name. Thus, a selector of `nav` would match all HTML `nav` elements, and a selector of `` would match all HTML unordered lists, or `` elements.

ID Selector

An ID selector is declared using a hash, or pound symbol (`#`) preceding a string of characters. The string of characters is defined by the developer. This selector matches any HTML element that has an ID attribute with the same value as that of the selector, but minus the hash symbol.

Class Selector

The class selector is the most useful of all CSS selectors. It's declared with a dot preceding a string of one or more characters. Just as is the case with an ID selector, this string of characters is defined by the developer. The class selector also matches all elements on the page that have their class attribute set to the same value as the class, minus the dot.

9. Define 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.
- In HTML 4.01, the `<!DOCTYPE>` declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.
- HTML5 is not based on SGML, and therefore does not require a reference to a DTD.

NOTE: Always add the `<!DOCTYPE>` declaration to your HTML documents, so that the browser knows what type of document to expect.

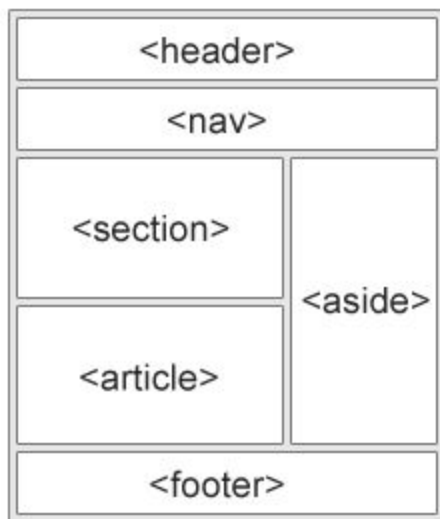
10. Explain 5 HTML5 semantic tags.

Semantic HTML elements clearly describe it's meaning in a human and machine readable way. Elements such as `<header>`, `<footer>` and `<article>` are all considered semantic because they accurately describe the purpose of the element and the type of content that is inside them.

List of new semantic elements

The semantic elements added in HTML5 are:

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



1. HTML5 `<section>` Element

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."

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

2. HTML5 `<article>` Element

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

3. HTML5 `<header>` Element

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

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

You can have several `<header>` elements in one document.

4. HTML5 `<footer>` Element

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.

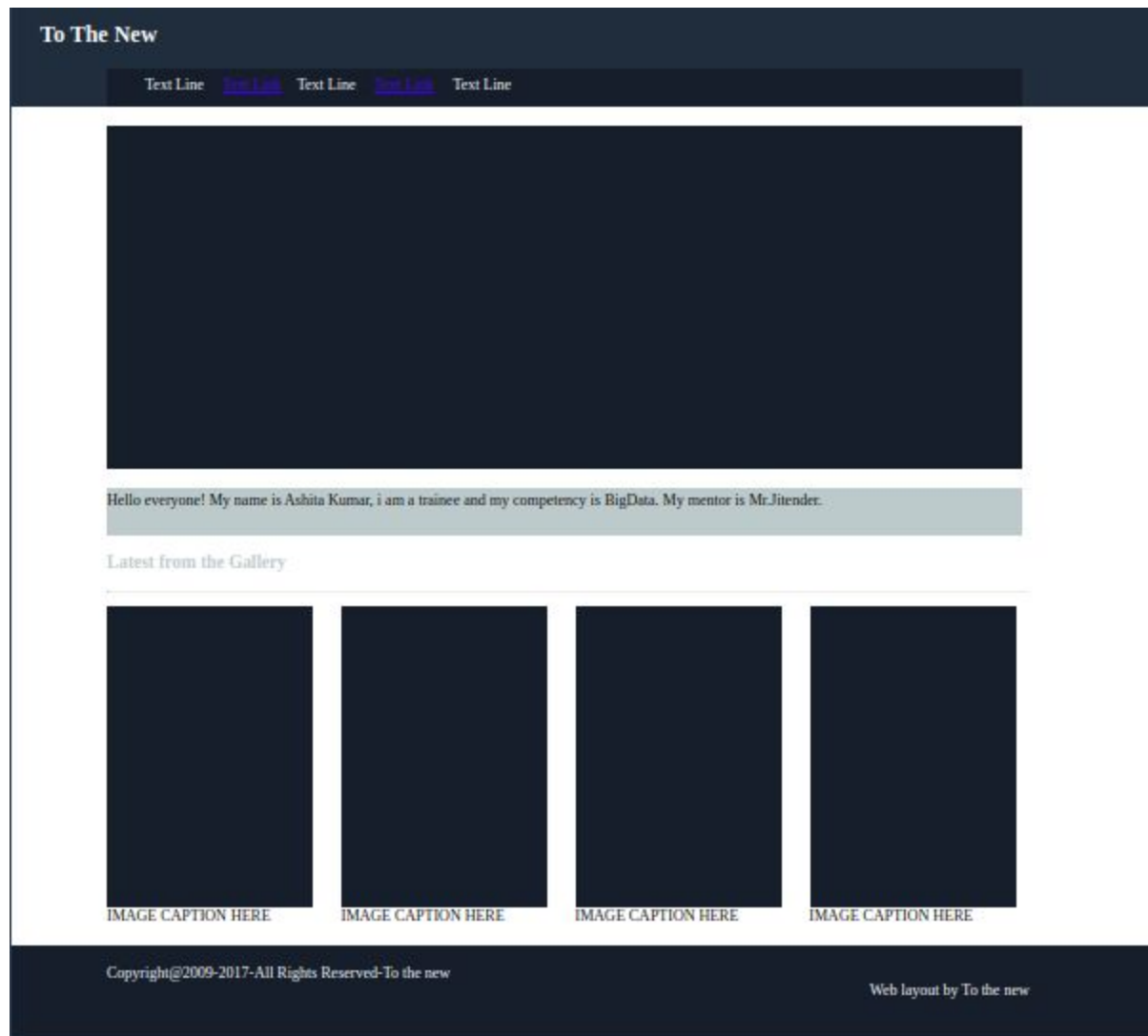
You may have several `<footer>` elements in one document.

5. HTML5 <nav> Element

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

NOTICE: Not all links of a document should be inside a `<nav>` element. The `<nav>` element is intended only for major blocks of navigation links.

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



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

TO THE NEW

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

Bug Report

Title:*

Description:*

Operating system

Windows XP

Product:*

Laptop

Version:*

License:

☒ Free ☐ Business

Severity

Critical

Attachments:

No file chosen