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

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). eg.<div>,<p>,<h1> elements are block elements.

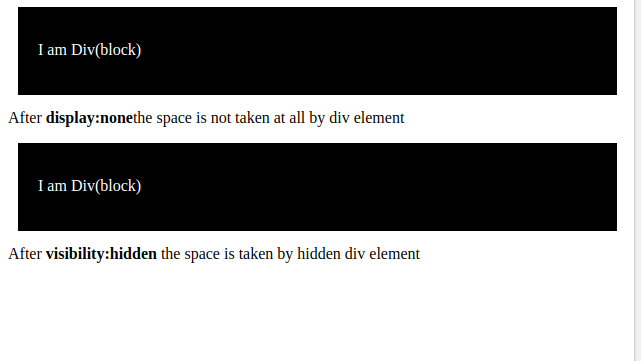
An inline element does not start on a new line and only takes up as much width as necessary.eg.<span>,<img>,<a>.

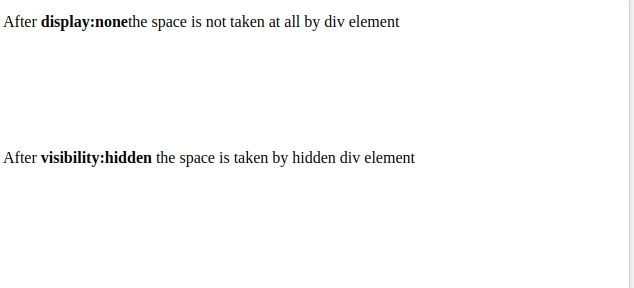


**2.Explain the difference between visibility:hidden and display:none.**

**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** means that unlike display:none , the tag is not visible, but space

is allocated for it on the page.



**3. Explain the clear and 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

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

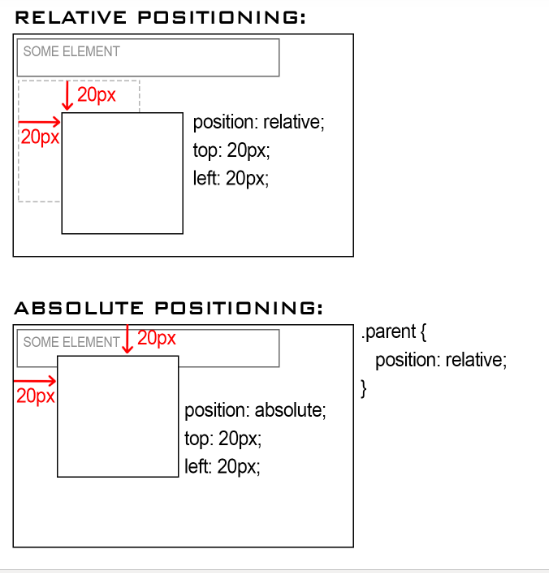
**4. explain difference between absolute, relative,fixed and static.**

**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.

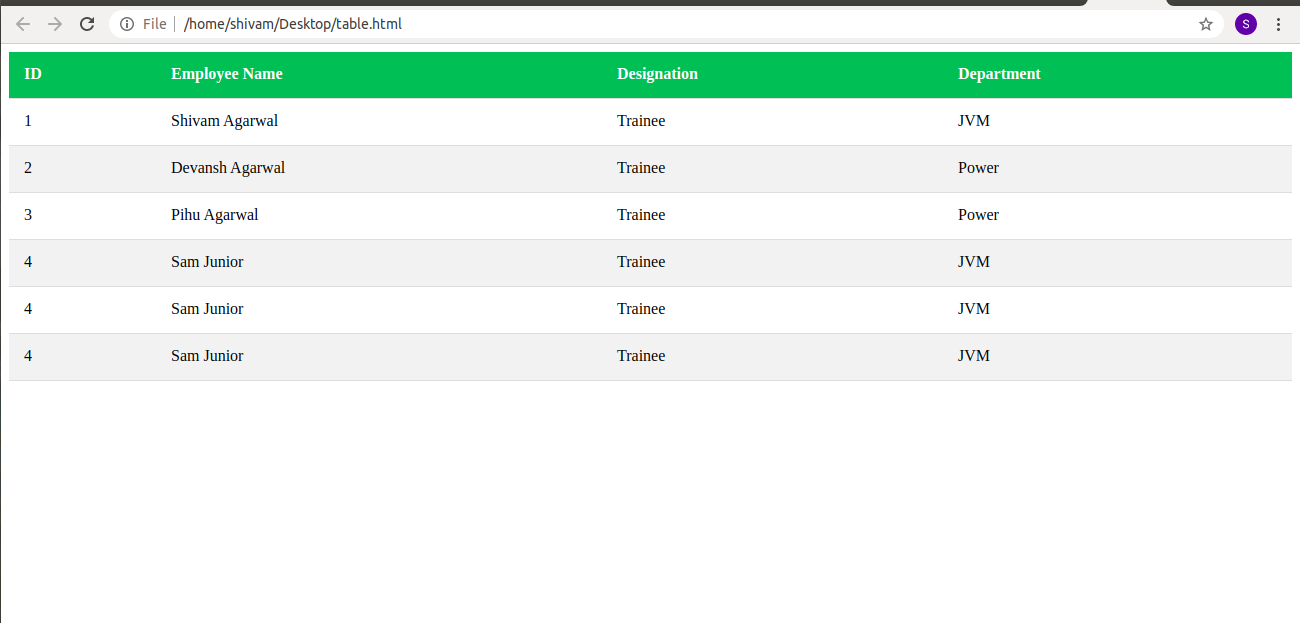
**Relative.** This type of positioning is probably the most confusing and misused. What it really means is "relative to itself". If you set position: relative; on an element but no other positioning attributes (top, left, bottom or right), it will have no effect on it's positioning at all, it will be 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.

**Fixed.** This type of positioning is fairly rare but certainly has its uses. A fixed position element is positioned relative to the viewport, or the browser window itself. The viewport doesn't change when the window is scrolled, so a fixed positioned element will stay right where it is when the page is scrolled, creating an effect a bit like the old school "frames" days.

**Absolute.** This is a very powerful type of positioning that allows you to literally place any page element exactly where you want it. You use the positioning attributes top, left, bottom. and right to set the location. Remember that these values will be relative to the next parent element with relative (or absolute) positioning. If there is no such parent, it will default all the way back up to the <html> element itself meaning it will be placed relative to the page itself.

****

**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.**

****

**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.

**<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">**

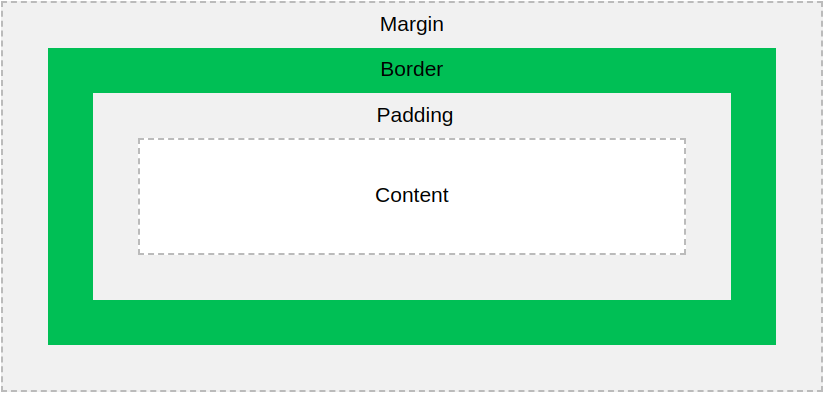
**<meta name="description" content="Session On HTML and CSS">**

**<meta name="author" content="Shivam Agarwal">**

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

**7. Explain 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.



* 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
* **Total element width = width + left padding + right padding + left border + right border + left margin + right margin**

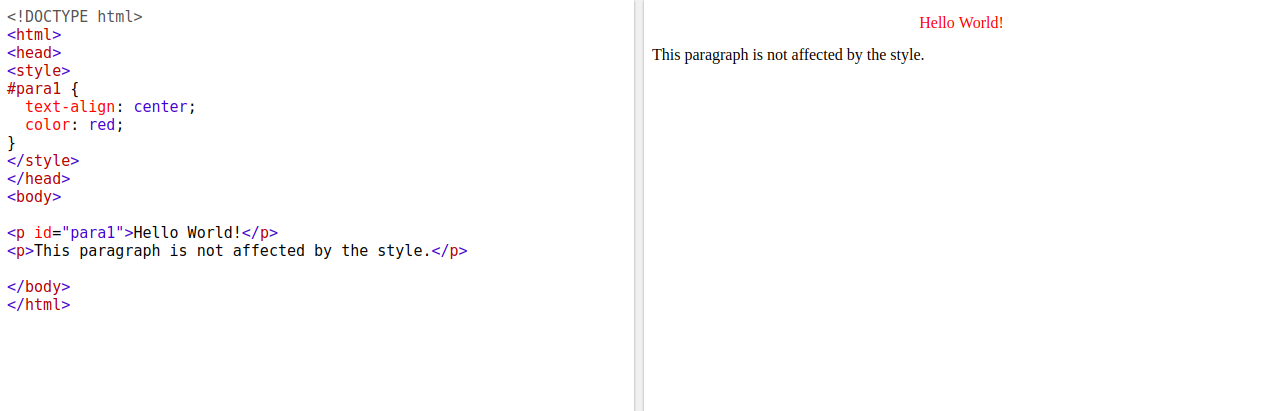
**8. What are the different types of CSS Selectors?**

**The 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!

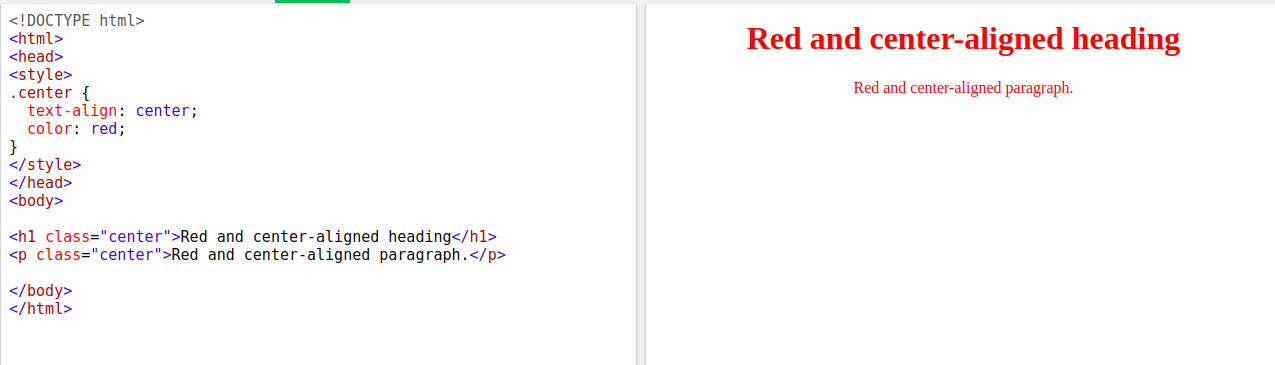
To select an element with a specific id, write a hash (#) character, followed by the id of the element.

****

**The 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.



**The universal selector:**

The universal selector (\*) selects all HTML elements on the page.

**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.

**10. 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.

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer">

to indicate navigation, header, and footer.

HTML5 offers new semantic elements to define different parts of a web page:

<article>

<aside>

<details>

<figcaption>

<figure>

<footer>

<header>

<main>

<mark>

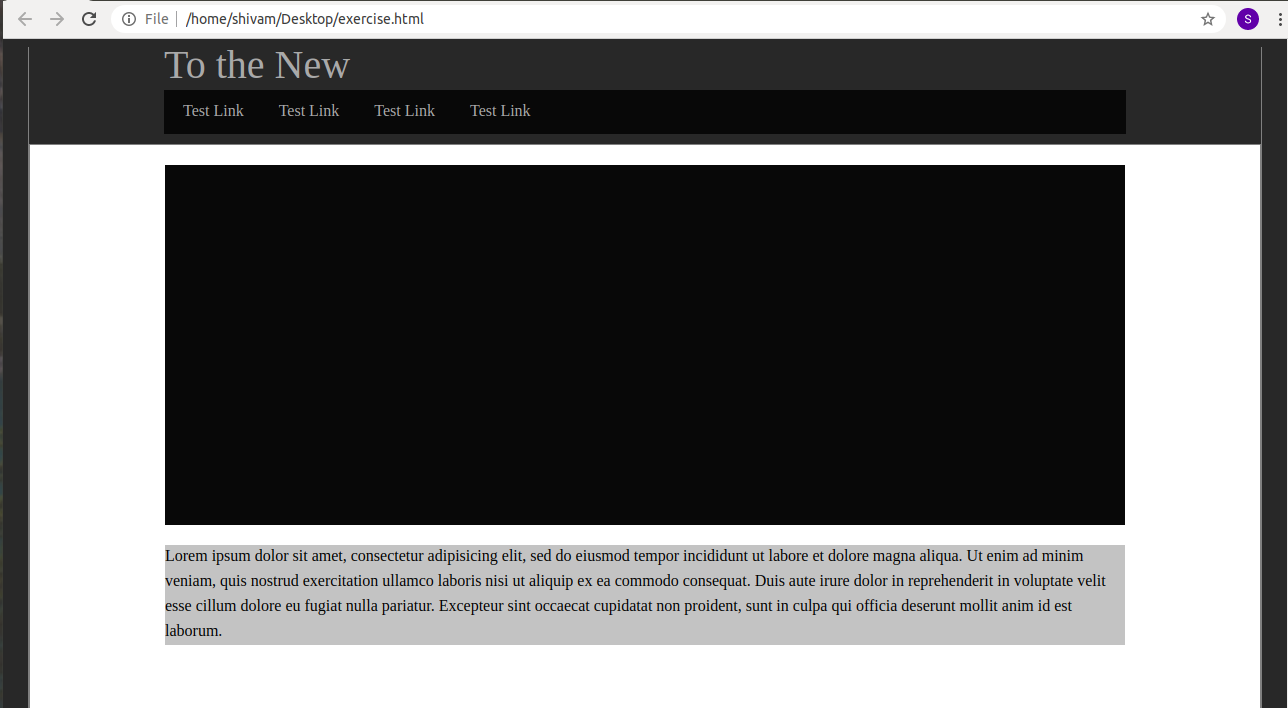
<nav>

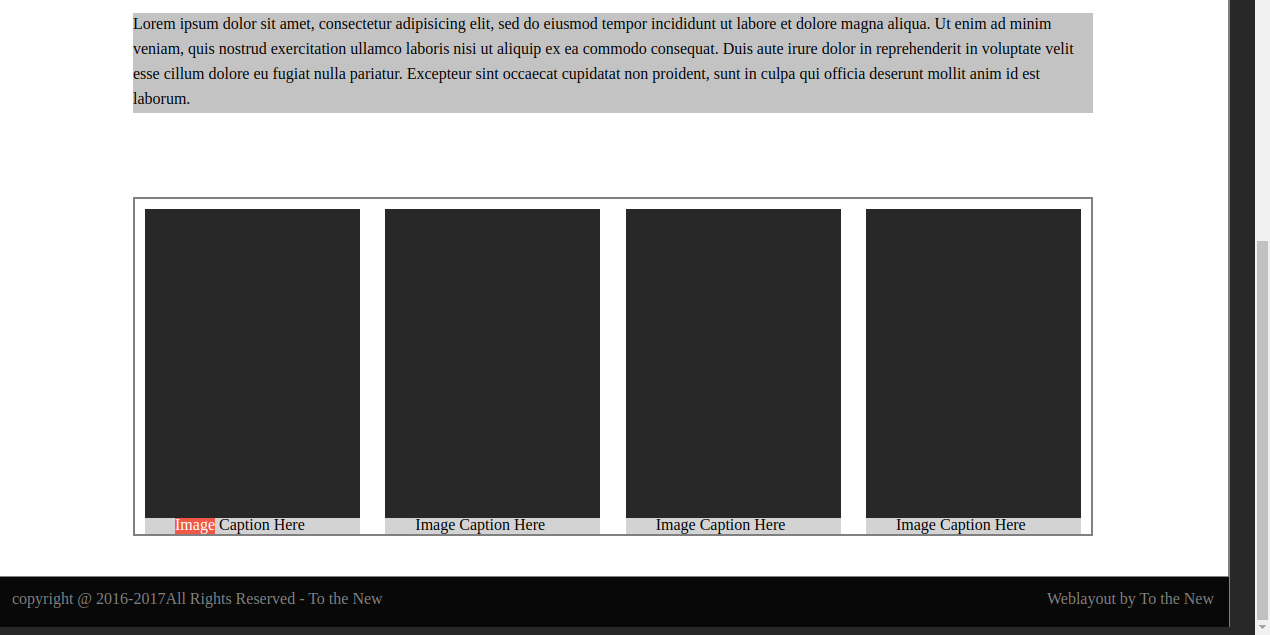
<section>

<summary>

<time>

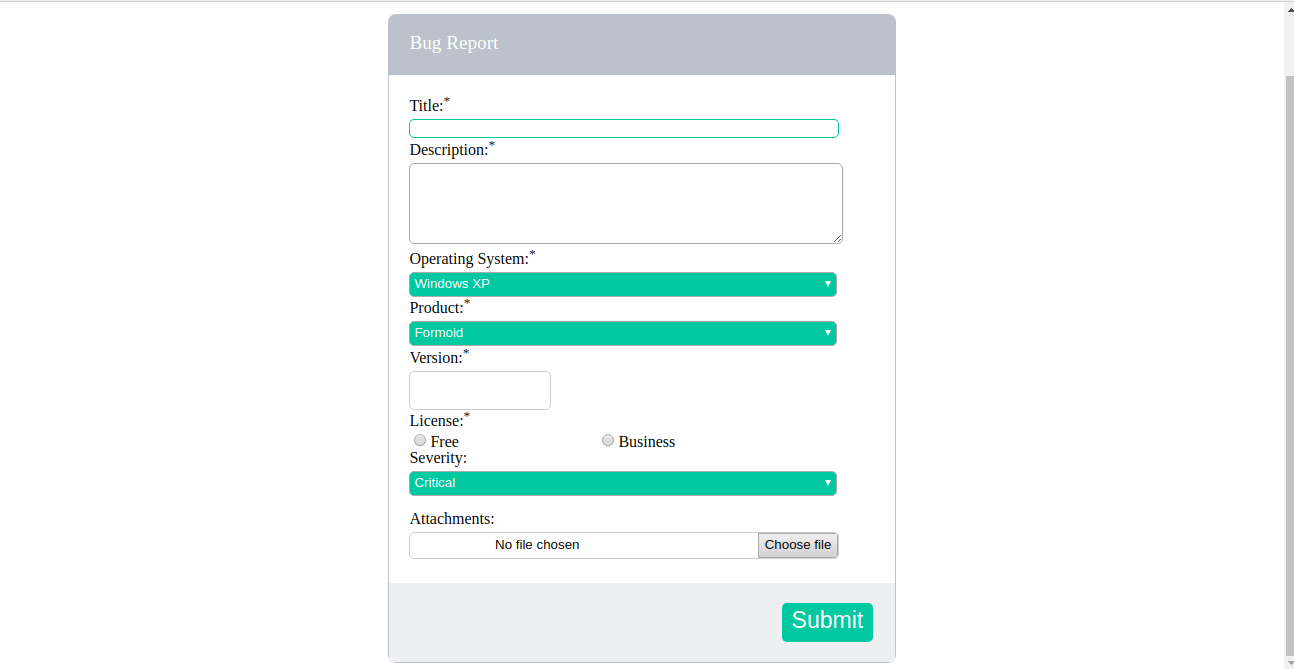
**11. Create HTML for web-page.jpg (check resources, highest weightage for answers)**





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

****

****