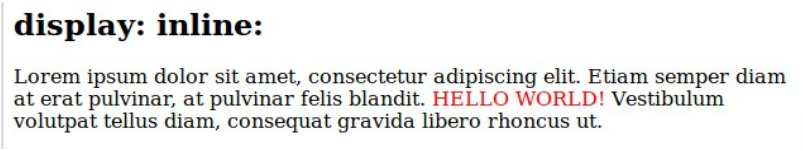
1.How are inline and block elements different from each other?

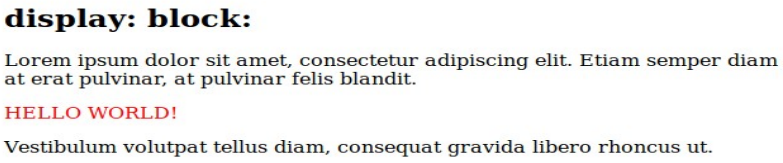
Inline:

Displays an element as an inline element (like). Any height and width properties will have no effect.



Block:

Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width.



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`

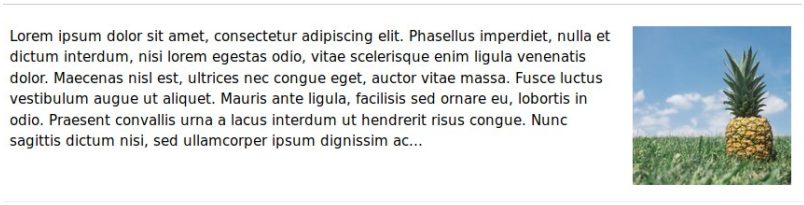
means that unlike **`display:none`,** the tag is not visible, but space is allocated for it on the page. The tag is rendered, it just isn't seen on the page.

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:

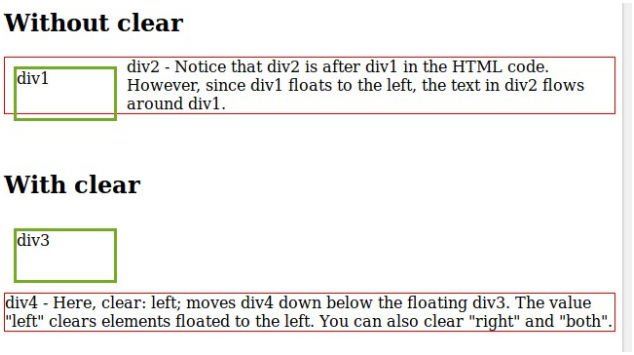
- left - The element floats to the left of its container
- right- The element floats to the right of its container
- none - The element does not float (will be displayed just where it occurs in the text). This is default
- inherit - The element inherits the float value of its parent.



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



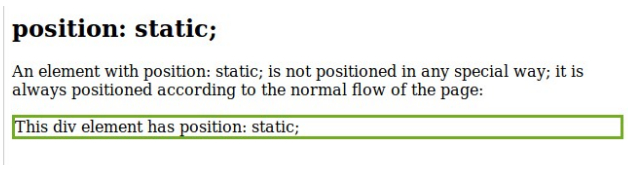
4. explain difference between absolute, relative,fixed and static.

Static:

HTML elements are positioned static by default.

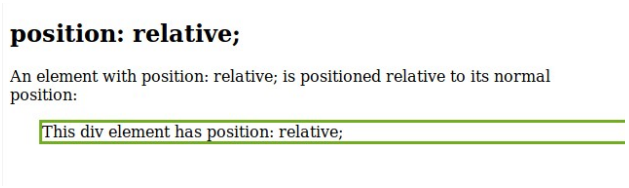
Static positioned elements are not affected by the top, bottom, left, and right properties.

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



Relative:

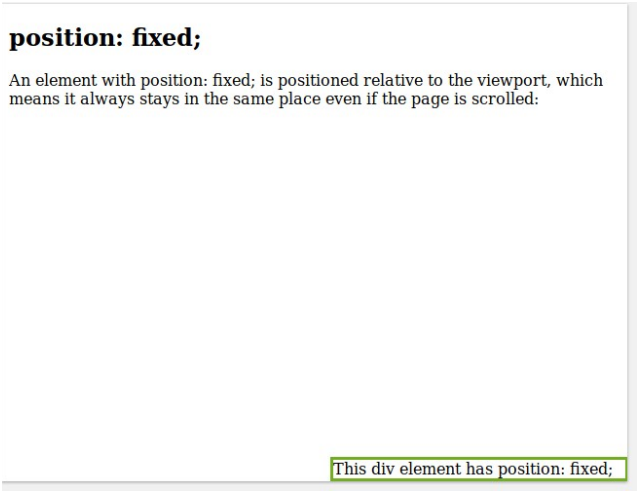
An element with **`position: relative;`** is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the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



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.

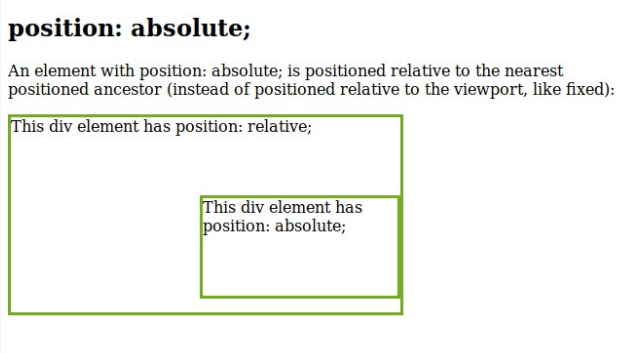
A fixed element does not leave a gap in the page where it would normally have been located.



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.



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.

Id	Employee Name	Designation	Department
1000	Gaurav Gandhi	Trainee	JVM
1001	Harish	Student	Mainframe
1002	Abhishek Anand	Student	Telecom
1003	Mehak	Trainee	JVM
1004	Sachin Tendulkar	Sportsman	Cricket
1005	Akshay Kumar	Actor	Bollywood

Code:

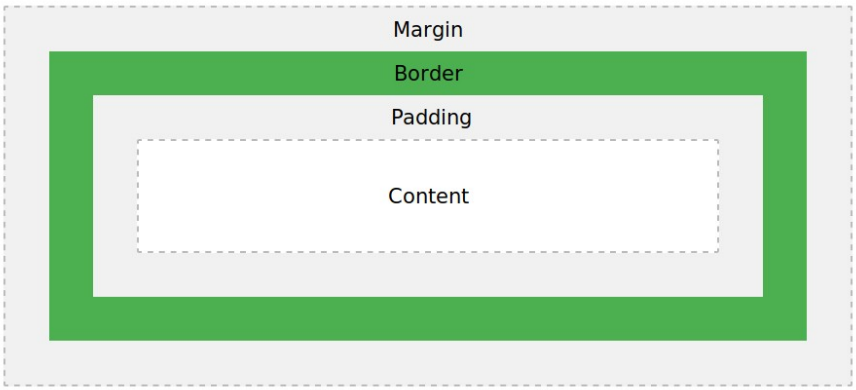
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Table Example</title>
    <style>
        table, th, td{
            border: 1px solid lawngreen;
            border-collapse: collapse;
            padding: 10px;
            text-align: center;
        }
        thead{
            background: lawngreen;
        }
        tr:nth-child(even){
            background-color: #f2f2f2
        }
    </style>
</head>
<body>
    <table style="border: 2px solid black">
        <thead id="heading">
            <th>Id</th>
            <th>Employee Name</th>
            <th>Designation</th>
            <th>Department</th>
        </thead>
        <tr id="1000">
            <td>1000</td>
            <td>Gaurav Gandhi</td>
            <td>Trainee</td>
            <td>JVM</td>
        </tr>
        <tr id="1001">
            <td>1001</td>
            <td>Harish</td>
            <td>Student</td>
```

```
            <td>Mainframe</td>
        </tr>
        <tr id="1002">
            <td>1002</td>
            <td>Abhishek Anand</td>
            <td>Student</td>
            <td>Telecom</td>
        </tr>
        <tr id="1003">
            <td>1003</td>
            <td>Mehak</td>
            <td>Trainee</td>
            <td>JVM</td>
        </tr>
        <tr id="1004">
            <td>1004</td>
            <td>Sachin Tendulkar</td>
            <td>Sportsman</td>
            <td>Cricket</td>
        </tr>
        <tr id="1005">
            <td>1005</td>
            <td>Akshay Kumar</td>
            <td>Actor</td>
            <td>Bollywood</td>
        </tr>
    </table>
</body>
</html>
```

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.

7. Explain 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

8. What are the different types of CSS Selectors?

The different types of CSS selectors are:

. (dot) : A dot is used to select a class and to apply the properties to all the tags which are present inside that class.

#(hash) : A hash is used to resemble id. A tag can be given a unique identity within a html page. To apply css properties to a particular tag id is used. We can refer an id using hash.

***(asterisk) :** It acts as a universal selector. It used to select all the elements present inside an html page. It is a wild card selector.

Element selector: it is represented using the name of the tag. We just have to use the html tag name to apply the css properties to that tag. Those properties will be applied to all the tags with that name present in the html 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.

10. Explain 5 HTML5 semantic tags.

<Section> :

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

<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

<Header> :

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.

<nav> :

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

<aside> :

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.

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

This html page is present in **web-page.html** file.

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

This html page is present in **form.html** file.