

HTML Assignment (By Siddharth Gupta)

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

Ans.

A `` element is used as an inline element and a `<div>` element as a block level element.

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
- emphasis `` tag
- image `` tag

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 ``, ``, `<dl>`, ``

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

Ans.

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. It removes the element from the normal flow of the page, allowing other elements to fill in.

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. It leaves the element in the normal flow of the page such that it still occupies space.

3. Explain the clear and float properties.

Ans.

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.

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.

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

Ans.

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

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

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:

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.

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.

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.

Note: A "positioned" element is one whose position is anything except static.

An element with ***position: sticky;*** is positioned based on the user's scroll position.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position: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.

Ans.

```
<!DOCTYPE html>
<html>
<head>
    <title>
        Table Example
    </title>
    <style type="text/css">
        table
        {
            border: 1px solid;
            border-collapse: collapse;
            width: 500px;
            padding: 4px;
        }
        th
        {
            border: 1px solid;
            height: 40px;
            background-color: green;
            color: white;
```

```
    }
    td
    {
        border: 1px solid;
        height: 25px;
        text-align: center;
    }
    tr:hover
    {
        background-color: lightgrey;
    }
</style>
```

</head>

<body>

<table>

<tr>

<th>ID</th>

<th>Employee Name</th>

<th>Designation</th>

<th>Department </th>

</tr>

<tr>

<td>101</td>

<td>Siddharth Gupta</td>

<td>Trainee</td>

<td>JVM</td>

</tr>

<tr>

<td>103</td>

<td>Gaurav Gandhi</td>

<td>Trainee</td>

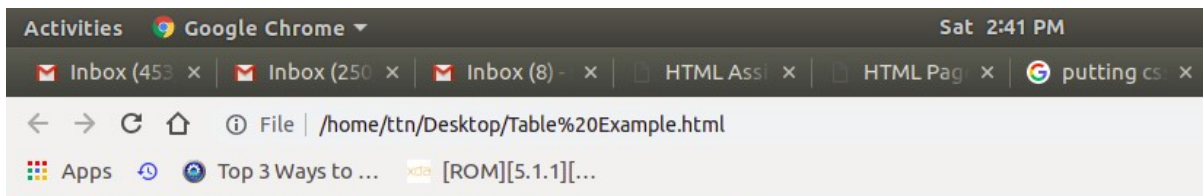
<td>JVM</td>

```
</tr>
<tr>
    <td>103</td>
    <td>Mehak Adhlakha</td>
    <td>Trainee</td>
    <td>iOS</td>
</tr>
<tr>
    <td>104</td>
    <td>Yatin</td>
    <td>Trainee</td>
    <td>JVM</td>
</tr>
<tr>
    <td>105</td>
    <td>Vishakha</td>
    <td>Trainee</td>
    <td>AMC</td>
</tr>
<tr>
    <td>106</td>
    <td>Arpit Gupta</td>
    <td>Trainee</td>
    <td>Python</td>
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```



6. Why do we use meta tags?

Ans.

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. Its mainly used for the Search Engine Optimization(SEO). Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

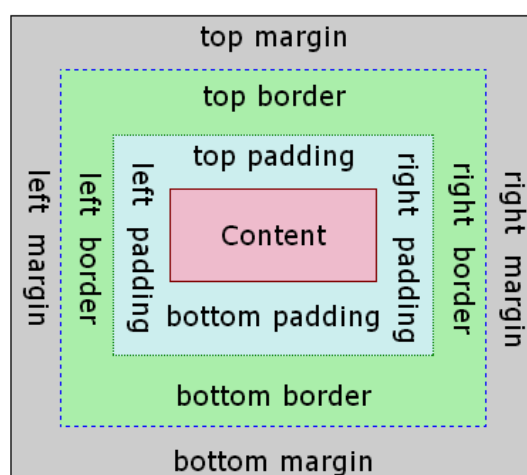
The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

HTML5 introduced a method to let web designers take control over the viewport (the user's visible area of a web page), through the <meta> tag

<meta> tags always go inside the <head> element. Metadata is always passed as name/value pairs.

7. Explain box model.

Ans.



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.

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

8. What are the different types of CSS Selectors?

Ans.

In CSS, selectors are patterns used to select the element(s) you want to style.

The **.class** selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

Syntax:

```
.class{  
  css declarations;  
}
```

The **#id** selector styles the element with the specified id.

Syntax:

```
#id{  
  css declarations;  
}
```

The ***** selector selects all elements.

The ***** selector can also select all elements inside another element

Syntax:

```
* {  
  css declarations;  
}
```

The **element** selector selects all elements with the specified element name.

Syntax:

```
element {  
  css declarations;  
}
```

To style **several elements** with the same style, separate each element name with a comma.

Syntax:

```
element,element {  
css declarations;  
}
```

The **element element selector** is used to select elements inside elements.

Syntax:

```
element element{  
css declarations;  
}
```

The **element>element selector** is used to select elements with a specific parent.

Note: Elements that are not directly a child of the specified parent, are not selected.

Syntax:

```
element > element {  
css declarations;  
}
```

9. Define Doctype.

Ans.

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(Document Type Definition), 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(Standard Generalized Markup Language), and therefore does not require a reference to a DTD.

Tip: 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.

Ans.

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

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

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

The **<article>** tag 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 site.

Potential sources for the <article> element:

- Forum post
- Blog post
- News story
- Comment

The **<aside>** tag defines some content aside from the content it is placed in.

The aside content should be related to the surrounding content.

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

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

Note: A <header> tag cannot be placed within a <footer>, <address> or another <header> element.

The **<footer>** tag defines a footer for a document or section.

A <footer> element should contain information about its containing element.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

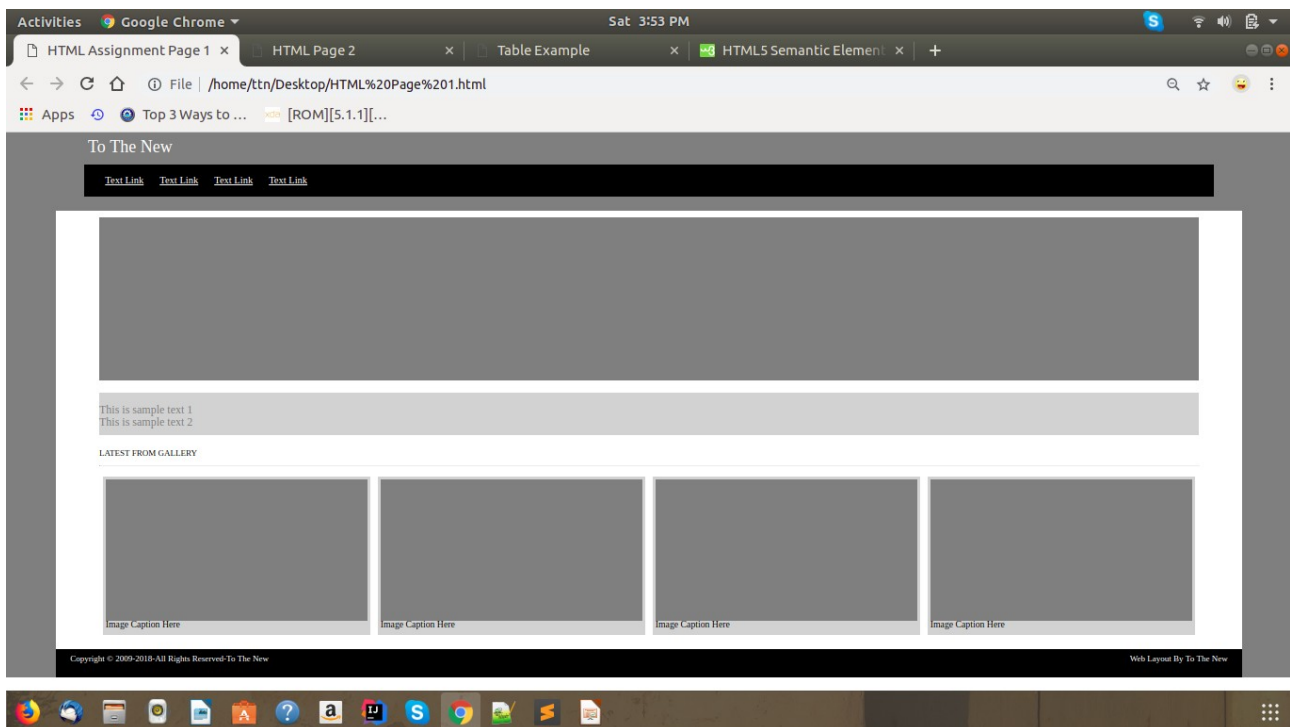
The **<nav>** tag defines a set of navigation links.

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.

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

Ans.



HTML Page 1.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>HTML Assignment Page 1</title>
```

```
</head>
```

```
<body style="margin: 0">
```

```
    <div style="background-color: grey; ">
```

```
        <div style="margin: 0px auto;background-color: grey; width: 90%; height: 100px; padding: 10px 20px 0px 20px;">
```

```
            <div style="color: white ; padding-left: 25px; border-bottom: 0px">
```

```
                <span style="font-size: 24px;">To The New<br></span>
```

```
            </div>
```

```
        <div style="color:white;background-color: black ;padding: 15px 10px 0px 20px;margin: 10px 20px 0 20px;height: 30px" >
```

```
14px">Text Link</a>
14px">Text Link</a>
14px">Text Link</a>
14px">Text Link</a>
</div>
</div>
```

```
<div style="margin: 0px auto;background-color: white; width: 90%; height:
600px; padding: 10px 20px 10px 20px">
  <div style="height: 230px ; width: 95%;background-color: grey;margin:
0 auto"></div><br>
  <div style="height: 60px ; width: 95%;background-color:
lightgrey;color:grey;margin: 0 auto;"><br>This is sample text 1<br>This is sample text
2</div><br>
  <div style="height: 25px ; width: 95%;margin: 0 auto"><span
style="font-size: 12px">LATEST FROM GALLERY</span>
  <hr style="width: 100%;color: lightgrey">
  </div><br>
  <div style="width: 95%;display: flex;margin: 0 auto;">
    <div style="height: 215px ; background-color: grey;margin: 0px
5px 0px 5px;flex: 1;border: solid;border-color:lightgrey">
      <div style="height: 200px"></div>
      <div style="height: 15px;background-color: lightgrey;font-
size: 13px">
        Image Caption Here
      </div>
    </div>
    <div style="height: 215px ; background-color: grey;margin: 0px
5px 0px 5px;flex: 1;border: solid;border-color:lightgrey">
      <div style="height: 200px"></div>
      <div style="height: 15px;background-color: lightgrey;font-
size: 13px">
```

Image Caption Here

</div>

</div>

<div style="height: 215px ; background-color: grey;margin: 0px 5px 0px 5px;flex: 1;border: solid;border-color:lightgrey">

<div style="height: 200px"></div>

<div style="height: 15px;background-color: lightgrey;font-size: 13px">

Image Caption Here

</div>

</div>

<div style="height: 215px ; background-color: grey;margin: 0px 5px 0px 5px;flex: 1;border: solid;border-color:lightgrey">

<div style="height: 200px"></div>

<div style="height: 15px;background-color: lightgrey;font-size: 13px">

Image Caption Here

</div>

</div>

</div>

</div>

<div style="margin: 0px auto;background-color: black; width: 90%; height: 20px; padding: 10px 20px 10px 20px">

Copyright © 2009-2018-All Rights Reserved-To The New

Web Layout By To The New

</div>

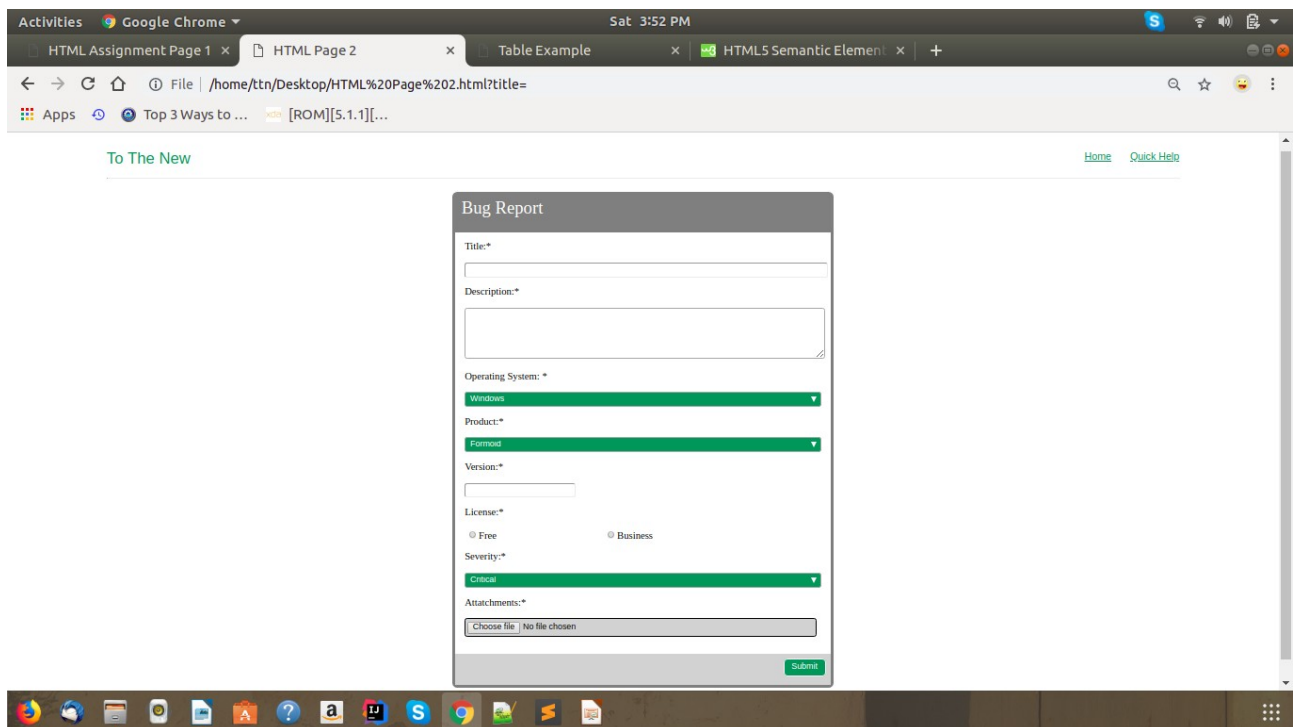
</div>

</body>

</html>

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

Ans.



HTML Page 2.html

```
<!DOCTYPE html>
<html>
<head>
  <title>
    HTML Page 2
  </title>
  <link rel="stylesheet" href="HTML2.css">
</head>
<body>
  <div class="container">
    <div class="innercontainer1">
      <br>
      <span id="heading">To The New</span>
      <a href=""><span style="margin-left: 30px">Quick Help</span></a>
      <a href=""><span style="">Home</span></a>
      <br><br>
      <hr style="width: 100%">
    </div>

    <div class="innercontainer2">
      <div style="width: 35%; margin: 0 auto;border-color: grey;border-style:
solid;border-radius: 8px;">
        <div id="divhead" style="">
          Bug Report
        </div>
        <div style="background-color: white;height: 640px;margin:
15px"><form>
          <p>Title:*</p>
```

```

required="">
<input type="text" name="title" class="text" style="width: 100%;"

<p>Description:*</p>
<textarea rows="5" style="width: 100%;border-radius: 4px"
required=""></textarea>
<p>Operating System: *</p>
<select class="select" required="">
    <option>Windows</option>
    <option>Linus/Ubuntu</option>
    <option>MacOS</option>
</select>
<p>Product:*</p>
<select class="select" required="">
    <option>Formoid</option>
    <option>Product 2</option>
    <option>Product 3</option>
    <option>Product 4</option>
</select>
<p>Version:*</p>
<input type="text" name="Version" class="text" required="">
<p>License:*</p>
<table style="width: 100%">
    <tr>
        <td><input type="radio" name="License"
>Free</td>
        <td><input type="radio" name="License"
>Business</td>
    </tr>
</table>
<p>Severity:*</p>
<select class="select" required="">
    <option>Critical</option>
    <option>Normal</option>
    <option>warning</option>
</select>
<p>Attatchments:*</p>
<input type="file" name="Attatchments" class="attatch"
required="">
</div>
<div style="background-color: lightgrey;height: 50px;">
    <input type="submit" name="Send" class="submit">
</div></form>
</div>
</body>
</html>

```

HTML Page 2.css

```

.container{
    width: 100%;
}

.innercontainer1{

```

```

        height: 50px;
        width: 85% ;
        padding: 15px;
        color: seagreen;
        margin: 0 auto
    }

    .innercontainer2{
        height: 800px;
        padding: 15px;
        width: 85%;
        margin: 0 auto
    }

    a{
        float: right;
        color: seagreen;
        font-family: sans-serif;
    }

    #heading{
        float: left;
        font-size: 25px;
        font-family: sans-serif;
    }

    #divhead{
        background-color: grey;
        height: 40px;
        padding: 10px;
        font-size: 28px;
        color:white;
        border: 10px;
    }

    .select{
        width: 100%;
        border-radius: 4px;
        color: white;
        background-color: seagreen;
        padding: 5px;
    }

    .attatch{
        width: 97%;
        border-radius: 4px;
        border: 1px solid;
        padding: 4px;
        background-color: lightgrey;
    }

    .text{
        border-radius: 4px;
        padding: 3px
    }

```

```
.submit{
    float: right;
    margin: 8px;
    border-radius: 6px;
    color: white;
    background-color: seagreen;
    border: 1px solid;
    padding: 6px 12px;
}
```