**Q1.How are inline and block elements different from each other?.**

**Answer**

|  |  |
| --- | --- |
| Block Elements | Inline Elements |
| Always start on a new line. | Doesn’t start on a new line. |
| Takes up full width available. (stretches from left to right as far as it can) | Takes up as much width as necessary |
| Eg- Paragraphs (<p>), headings(h1 through h6),Divisions(<div>),list and list items (<ul>,<ol>, and <li>), Forms (<form>) | Eg- Spans (<span>), Images(<img) and Anchors(<a>) |
| Causes a line break before and after element. |  |

* Inline and block properties can be switched using display property in CSS.

**Example**

**HTML**

<html>

<head>

<title>DIV vs SPAN</title>

</head>

<body>

<div style='background-color:red'>Div 1</div>

<div style='background-color:yellow'>Div 2</div>

<span style='background-color:blue'>Span 1</span>

<span style='background-color:green'>Span 2</span>

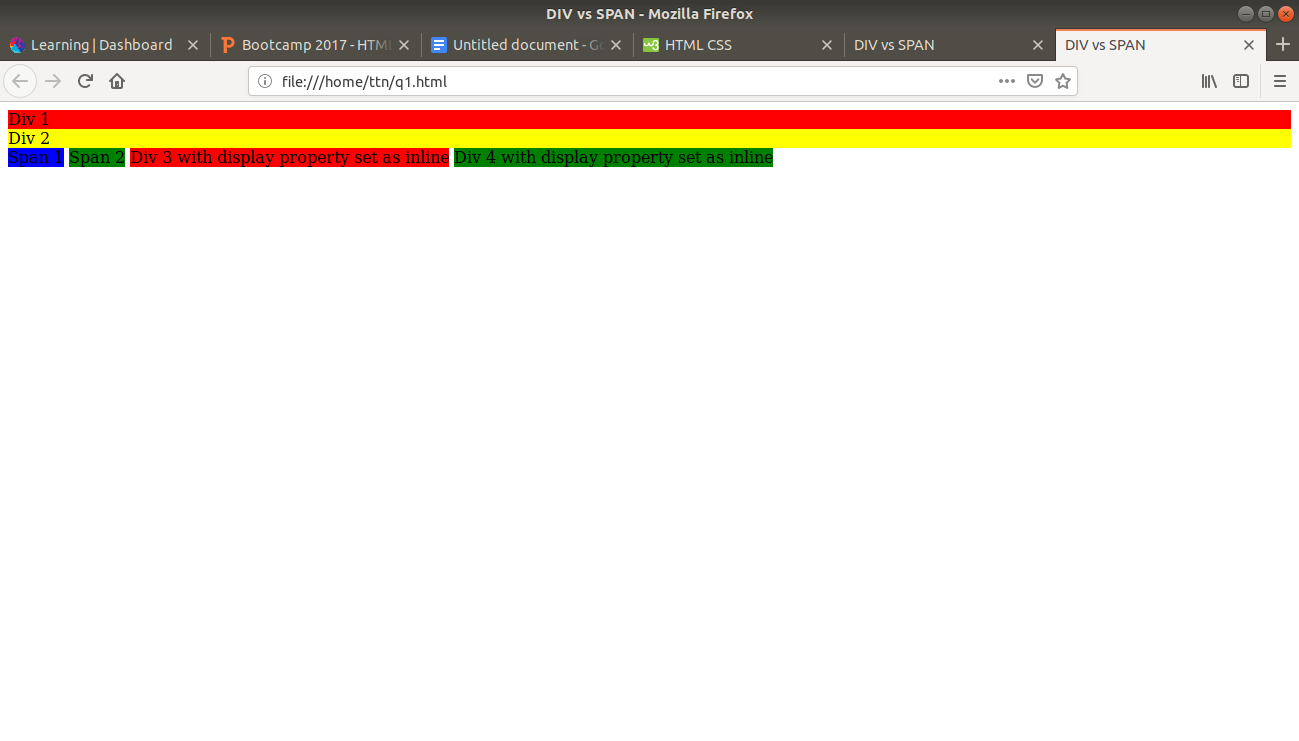
<div style='background-color:red;display:inline'>Div 3 with display property set as inline</div>

<div style='background-color:green;display:inline'>Div 4 with display property set as inline</div>

</body>

</html>

**Output**



**Q2.Explain the difference between visibility:hidden and display:none**

**Answer**

Display:none → The tag will not appear on the page at all, i.e., there will be no space allocated for it between other tags. (however, we can interact through such elements using DOM).

Visibility:hidden → The tag is not visible, but space is allocated for it on the page.

**Example**

<html>

<head>

<title> Display VS Visibility</title>

</head>

<body>

<h2>Visibility hidden for second paragraph</h2>

<p>Hi, This is Surbhi Garg</p>

<p style='visibility:hidden'>I am pursuing MCA </p>

<p>Currently I am a part of TTN bootcamp</p>

<hr/>

<h2>Display none for second paragraph</h2>

<p>Hi, This is Surbhi Garg</p>

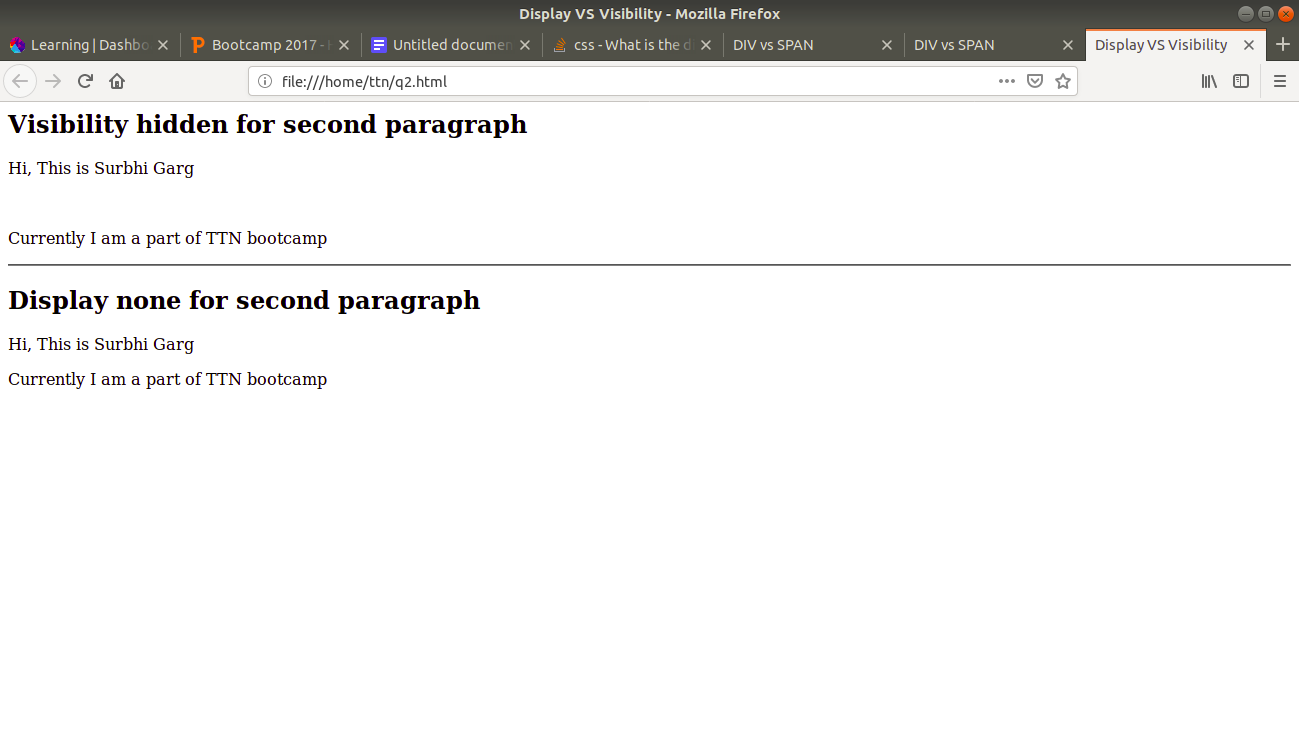
<p style='display:none'>I am pursuing MCA </p>

<p>Currently I am a part of TTN bootcamp</p>

</body>

</html>

**Output**



**Q3. Explain the clear and float properties.**

**Answer**

**Float:**

* It is a positioning property in css.
* It is used to flow an element to the left or right corner of parent container and next element wrap around it. A floating element doesn’t wrap occupy space on the normal flow. Thus, we have to use clear:both after last floating element to avoid wrapping.
* By default, all html elements are non-floating.

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).

→ inherit - The element inherits the float value of its parent

**Clear:**

* It is used to stop the next element to wrap around the adjacent floating elements.

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

**Example**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Float And Clear</title>

<style>

.floattry

{

float:left;

}

</style>

</head>

<body>

<div class="floattry">div1</div>

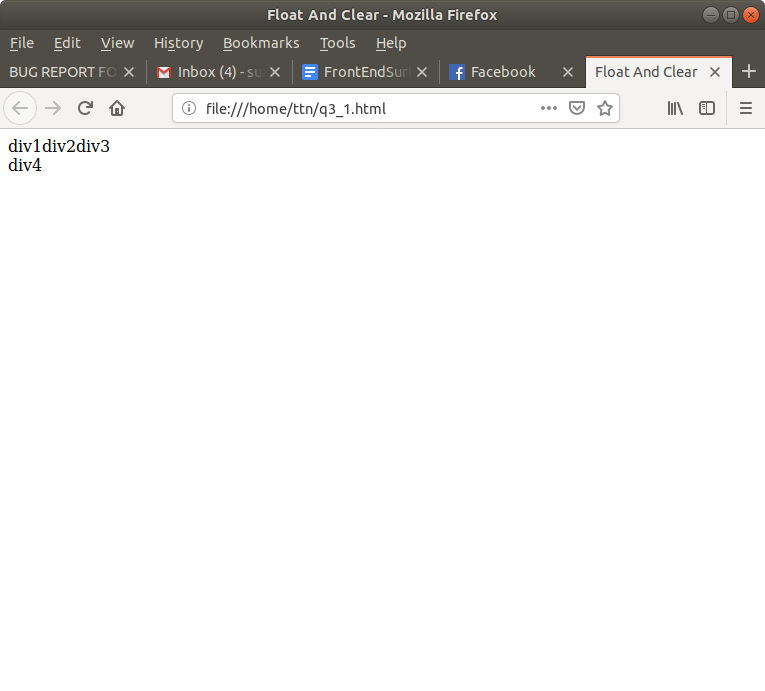
<div class="floattry">div2</div>

<div class="floattry">div3</div>

<div class="floattry" style="clear:left">div4</div.

</body>

</html>

**Output**

**Q4. Explain difference between absolute, relative,fixed and static.**

**Answer**

* **All are possible values of position attribute.**

**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.

**Example**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Position</title>

<style>

div.relative {

position: relative;

width: 400px;

height: 200px;

border: 3px solid rgb(233, 121, 56);

}

div.static

{

position: static;

border:2px solid salmon;

}

div.absolute {

position: absolute;

top: 60px;

left: 10px;

right: 0;

width: 200px;

height: 100px;

border: 3px solid #73AD21;

}

div.fixed

{

position:fixed;

bottom: 0;

right:0;

height: 40px;

border:2px solid springgreen;

}

</style>

</head>

<body>

<div class="static">

Static Div

</div>

<div class="relative">Relative div

<div class="absolute">Absolute div</div>

</div>

<div class="fixed">

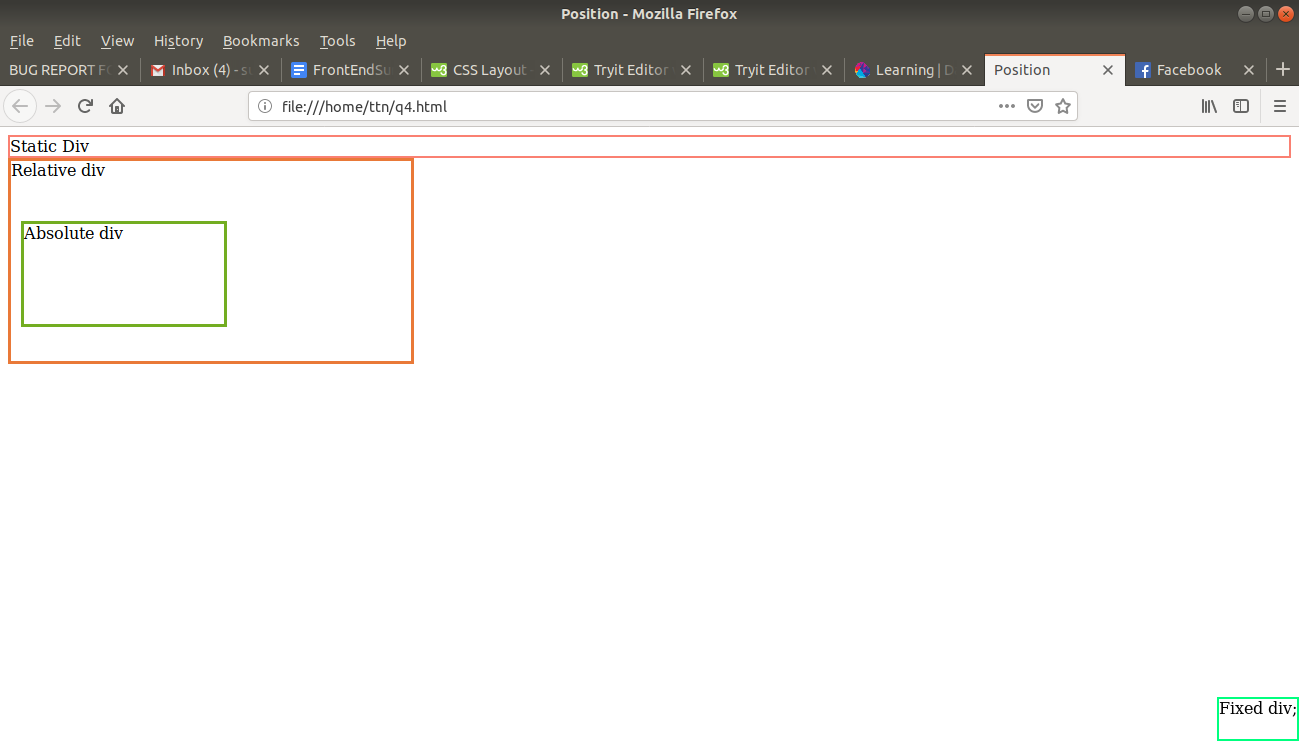
Fixed div;

</div>

</body>

</html>

**Output**



**Q5. 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.**

**HTML**

<html lang="en">

<head>

<title>Table Design</title>

<style>

#emp

{

border:2px solid;

border-collapse:collapse;

overflow:scroll;

}

#emp>thead,td,th

{

border:2px solid black;

padding: 20px;

background-color: burlywood;

font-family: 'Courier New', Courier, monospace;

}

#emp>tr,td:hover

{

background-color:chocolate;

font-size: 15px;

}

</style>

</head>

<body>

<table id="emp">

<thead>

<tr>

<th>ID</th>

<th>Employee Name</th>

<th>Designation</th>

<th>Department</th>

</tr>

</thead>

<tr>

<td>1001</td>

<td>Surbhi Garg</td>

<td>Trainee</td>

<td>IT</td>

</tr>

<tr>

<td>1002</td>

<td>Vagish Dixit</td>

<td>Trainee</td>

<td>IT</td>

</tr>

<tr>

<td>1003</td>

<td>Rishabh Rajput</td>

<td>Software Developer</td>

<td>JVM</td>

</tr>

<tr>

<td>1004</td>

<td>Gagan Kushwaha</td>

<td>Trainee</td>

<td>JVM</td>

</tr>

<tr>

<td>1005</td>

<td>Sakshi</td>

<td>Trainee</td>

<td>FEEN</td>

</tr>

<tr>

<td>1006</td>

<td>Yukti Sharma</td>

<td>Trainee</td>

<td>JVM</td>

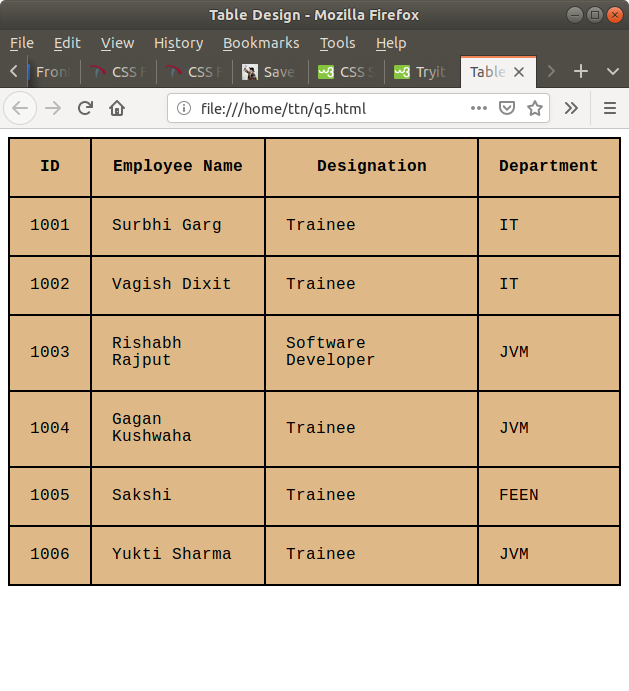
</tr>

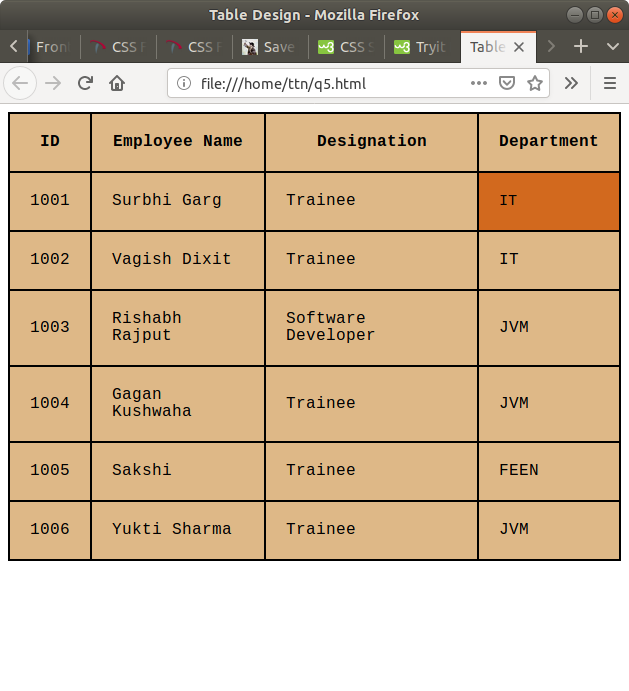
</table>

</body>

</html>

**Output**





**Q6. Why do we use meta tags?**

**Answer**

The <meta> tag provides metadata about the html document. Metadata will not be displayed on the page, but will be machine parsable.

* Metadata tags are typically used to specify page description, keywords, author of the document, last modified, and other metadata;
* 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 <head> elements.

**Example**

<html>

<head>

<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript"> (For Search engines)

<meta name="description" content="Free Web tutorials on HTML and CSS"> ( Description of web page)

<meta name="author" content="Surbhi Garg"> (Author of web page)

<meta name="viewport" content="width=device-width, initial-scale=1.0"> (Responsiveness)

</head>

**Q7. Explain box model.**

**Answer**

All elements can be considered as boxes. In CSS terms, the “Box model” is used when talking about design and layout.

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

Parts

Content→ The content of box, where text and image appears.

Padding→ Clears an area around the content.

Border→ A border that goes around content and padding.

Margin→ Clears an area outside the border. The margin is transparent.

Margin

Border

Padding

Content

**Q8.What are the different types of CSS selectors?**

**Answer**

Selectors are patterns used to select the elements we want to style.

|  |  |  |
| --- | --- | --- |
| Selector | Example | Description |
| .class | .tab | Selects all elements with class=”tab” |
| #id | #firstname | Select the element withid=”first name” |
| \* | \* | Selects all elements |
| element | h1 | Selects all <h1> elements. |
| element1,element2 | div,p | Selects all div and all p elements. |
| Element element | Div p | Selects all <p> elements inside <div> elements. |
| element>element | div>p | Selects all <p> elements where the parent is a <div> elements. |
| element+element | div+p | Selects all <p> elements that are placed immediately after <div> elements. |
| element~element2 | p~ul | Selects every <ul> element preceded by a <p> element. |
| [attribute] | [target] | Selects all elements with target attribute. |
| [attribute=value] | [target=\_blank] | Selects all elements with target=”\_blank”. |
| [attribute~=value] | [title~=flower] | Selects all elements with title attribute containing the word flower. |
| [attribute|=value] | [lang|=en] | Selects all elements with a lang attribute value starting with en. |
| [attribute ^= value] | a[href^=”https” | Selects every <a> element whose href element begins with https. |
| [attribute$=value] | a[href$=”.pdf” | Selects every <a> element whose href element ends with “.pdf”. |
| [attribute\*=value] | a[href\*=”ToTheNew”] | Selects every a element whose href element contains the substring “ToTheNew”. |
| :active | a:active | Selects the active link. |
| ::after | p::after | Insert something after the content of each <p> element. |
| ::before | p::before | Insert something before the content of each <p> element. |
| :checked | input:checked | Selects every checked input element |
| :default | input:default | Selects the default <input>element |
| :input | input:disabled | Selects every disabled <input>element. |
| :empty | p:empty | Selects every <p> element that has no children (including text nodes). |
| :enabled | input:enabled | Selects every enabled <input.element. |
| :first-child | p:first-child | Selects every <p> element that is the first-child of its parent element. |
| ::first-letter | p::first-letter | Selects the first letter of every <p> element. |
| ::first-line | p::first-line | Selects the first line of every <p> element. |
| :first-of-type | p:first-of-type | Selects every <p> element that is the first <p> element of its parent. |
| :focus | input:focus | Selects the input element that has focus. |
| :hover | a:hover | Selects link on mouse hover. |
| :in-range | input:in-range | Selects input element with a value within specified range. |
| :indeterminate | input:indeterminate | Selects input elements that are in an indeterminate state. |
| :invalid | input:invalid | Selects all input having invalid value. |
| :lang(langage) | p:lang(it) | Selects every <p> element where language=”it” (italian). |
| :last-child | p:last-child | Selects every <p> element that is last child of its parent |
| :last-of-type | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| :link | a:link | Selects all unvisited links |
| :not(selector) | :not(p) | Selects every element that is not a <p> element. |
| :nth-child(n) | p:nth-child(2) | Selects every <p> element that is the second child of its parent. |
| :nth-last-child(n) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent,counting from the last child. |
| :nth-of-type(n) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent. |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p> element of its parent. |
| :only-child | p:only-child | Selects every <p> element that is the only child of its parent. |
| :optional | input:optional | Selects input element with no “required” attribute. |
| :out-of-range | p:out-of-range | Selects input elements with a value outside a specified range. |
| ::placeholder | input::placeholder. | Selects input elements with placeholder text. |
| :read-only | input:read-only | Selects all input elements with “read-only” attribute specified. |
| :read-write | input:read-write | Selects all input elements with “read-only” attribute NOT specified. |
| :required | input:required | Selects all input elements with “required” attribute specified. |
| :root | :root | Selects the document’s root element. |
| ::selection | ::selection | Selects the portion of an element that is selected by a user. |
| :target | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name). |
| :valid | input:valid | Selects all input elements with a valid value. |
| :visited | a:visited | Selects all visited links. |

**Q9. Define doctype.**

**Answer**

The <!Doctype> declaration must be the very first thing in HTML document, before html tag.

The <!Doctype> declaration is not a HTMLtag; 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 browser renders the content correctly.

HTML5 is not based on SGML, and therefore, does not require a reference to DTD.

**Q10.Explain 5 HTML5 semantic tags.**

**Answer.**

**Semantic Elements**

A semantic element is an element that clearly defines it meaning to both developer and browser.

Eg- form, table and article.

Examples of non-semantic elements are-form and div.

**Semantic tags in HTML5**

**<header> element**

→ It specifies a header for a document or section.

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

→ We can have several header elements in a document.

**<article> element**

→ It specifies independent, self contained content.

→ An article should make sense on it’s own, and it should be possible to read it independently from the rest of the site.

→ Example where it can be used:

* Forum post
* Blog post
* Newspaper article

**<footer> element**

→ It specifies a footer for a document or section.

→ It should contain information about it’s containing element.

→ A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.

→ We may have several <footer> elements in one document.

**<nav> element**

→ It defines a set of navigation links.

**<figure>and <figcaption>element.**

The purpose of a figure caption is to add a visual explanation to an image.

In HTML5, an image and a caption can be grouped together in a <figure> element.

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>HTML5 semantic elements</title>

</head>

<body>

<nav>

<a href="/home/">HOME</a> |

<a href="/aboutus/">About Us</a> |

<a href="/quicklinks/">Important contacts</a> |

<a href="/gallery/">Gallery</a>

</nav>

<article>

<header>

<h1>What does TTN do?</h1>

</header>

<p>

It is a service based organization, which works on product

engineering and digital transformation.

</p>

</article>

<figure>

<img src=”/home/ttn/Desktop/TTN-logo.jpg">

<figcaption>Fig1.This is the logo</figcaption>

</figure>

<footer>

Author:Surbhi Garg

Email:surbhi.garg@tothenew.com

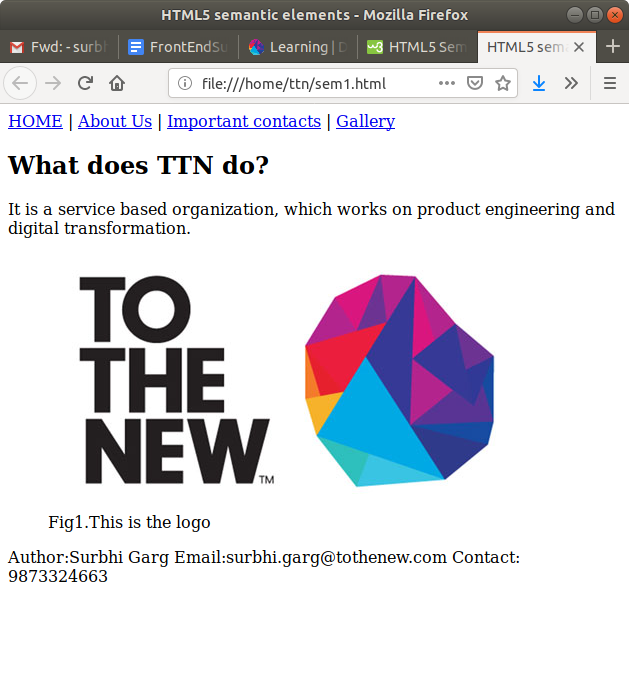
Contact: 9873324663

</footer>

</body>

</html>

**Output**



**Q11. Create HTML for web-page.jpg**

**Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>To The New</title>

<style>

#main-container

{

height: 100%;

background-color: #383838;

}

.pagebody

{

width:80vw;

background-color: #323232;

margin:0 auto;

position:relative;

}

#header

{

padding-left:2em;

padding-right:2em;

}

.main-heading

{

color:#c4b6ab;

padding-top:0.5em;

}

nav

{

background-color: #222222;

position: relative;

padding:0.6em;

width:auto;

}

a

{

color:#c4b6ab;

text-decoration:none;

margin:0.2em;

}

#inside-body

{

padding-top: 1em;

background-color: #ffff;

padding-bottom: 20px;

}

.content

{

color:#c4b6ab;

background-color: #323232;

width: 960px;

height:360px;

margin:auto;

font-family:Verdana, Geneva, Tahoma, sans-serif;

text-align: center;

position: relative;

}

.content>p

{

position: absolute;

padding:170px 300px 150px 380px ;

}

footer

{

color:#666666;

background-color: #202020;

padding:1em 0em 1em 0em;

}

#second

{

background-color:#eeeeee;

margin:2em auto;

width:960px;

font-size: small;

color:#b5b5b5;

padding:0.3em;

}

.image

{

background-color: #323232;

flex-direction: column;

color:#ffff;

text-align: center;

height:313px;

position: relative;

}

.image>p

{

position:absolute;

padding:8em 2em 8em 1em;

color:white;

}

#flex-container

{

display: flex;

}

p{

color:#9d9ca4;

margin-left: 40px;

}

#cr,#fr

{

display: inline;

}

#fr

{

margin-left:300px;

}

hr{

color:silver;

margin-left: 40px;

margin-right: 35px;

}

.img-container

{

background-color: white;

padding:0.2em;

flex-direction: column;

border: 0.5px solid gainsboro;

margin-right: 35px;

width:215px;

}

.caption

{

background-color: #eeeeee;

color: #959595;

font-size: 15px;

text-align: center;

}

</style>

</head>

<body>

<div id="main-container">

<div class="pagebody">

<div id="header">

<h3 class="main-heading">To the New</h3>

<nav>

<a href="/text-link/">TEXT LINK</a>

<a href="/text-link/">TEXT LINK</a>

<a href="/text-link/">TEXT LINK</a>

<a href="/text-link/">TEXT LINK</a>

<a href="/text-link/">TEXT LINK</a>

</nav>

</div>

<div id="inside-body">

<div class="content">

<p>960\*360 px</p>

</div>

<div id="second">

VESTIBULUMAACCUMASAN EGESTIBULUM EU JUSTO CONVALIS AUGO ESTAS ANEAN ELIT INTESQUE SED.FACILISPEDE ESTIBULUM NULLA ORNA NSIL VELIT ELIT AC ELIQUIT NON TINCIDUNT NAMUJUSTO CRAS URNA

</div>

<p>LATEST FROM GALLERY<hr/></p>

<div id="flex-container">

<div class="img-container" style="margin-left:2.7em;">

<div class="image"><p>215\*315px</p></div>

<article class="caption">IMAGE CAPTION HERE</article>

</div>

<div class="img-container">

<div class="image"><p>215\*315px</p></div>

<article class="caption">IMAGE CAPTION HERE</article>

</div>

<div class="img-container">

<div class="image"><p>215\*315px</p></div>

<article class="caption">IMAGE CAPTION HERE</article>

</div>

<div class="img-container">

<div class="image"><p>215\*315px</p></div>

<article class="caption">IMAGE CAPTION HERE</article>

</div>

</div>

</div>

<footer>

<p id="cr">Copyright &copy; 2009-2017-All rights reserved-To the New</p>

<p id="fr">Web layout by To the New.</p>

</footer>

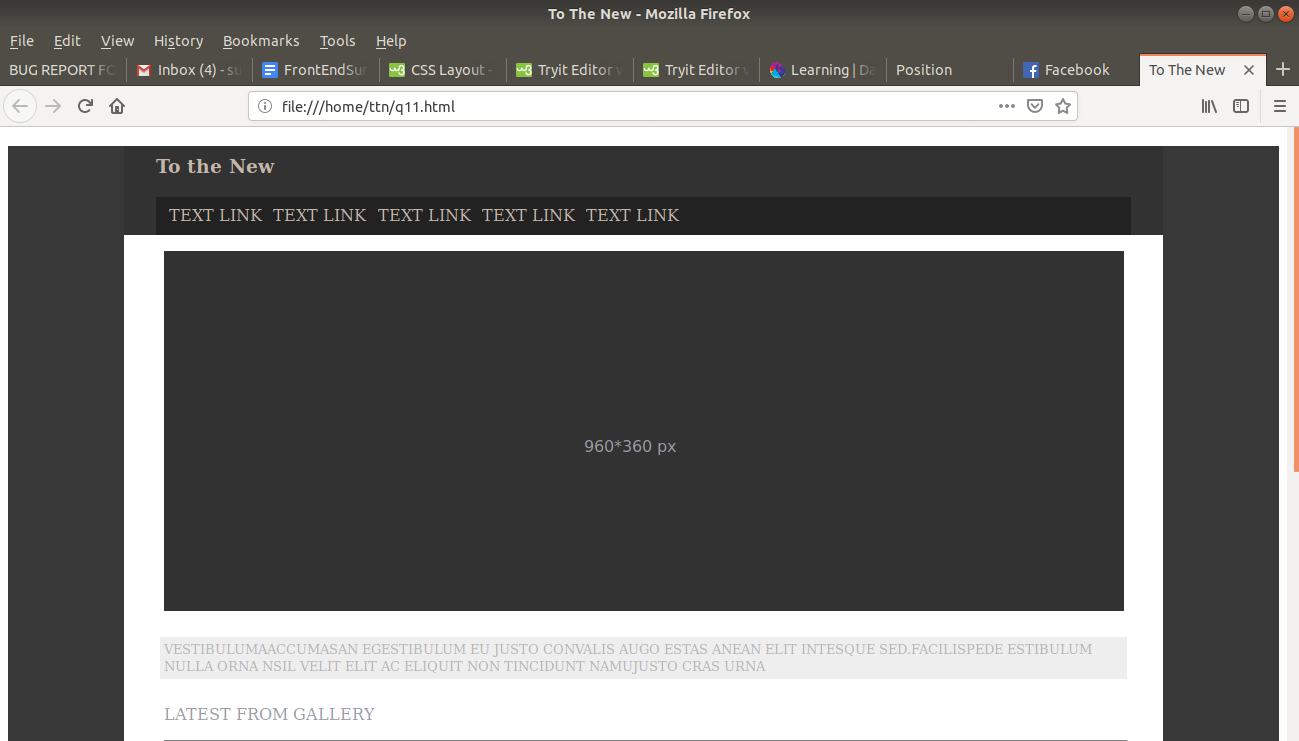
</div>

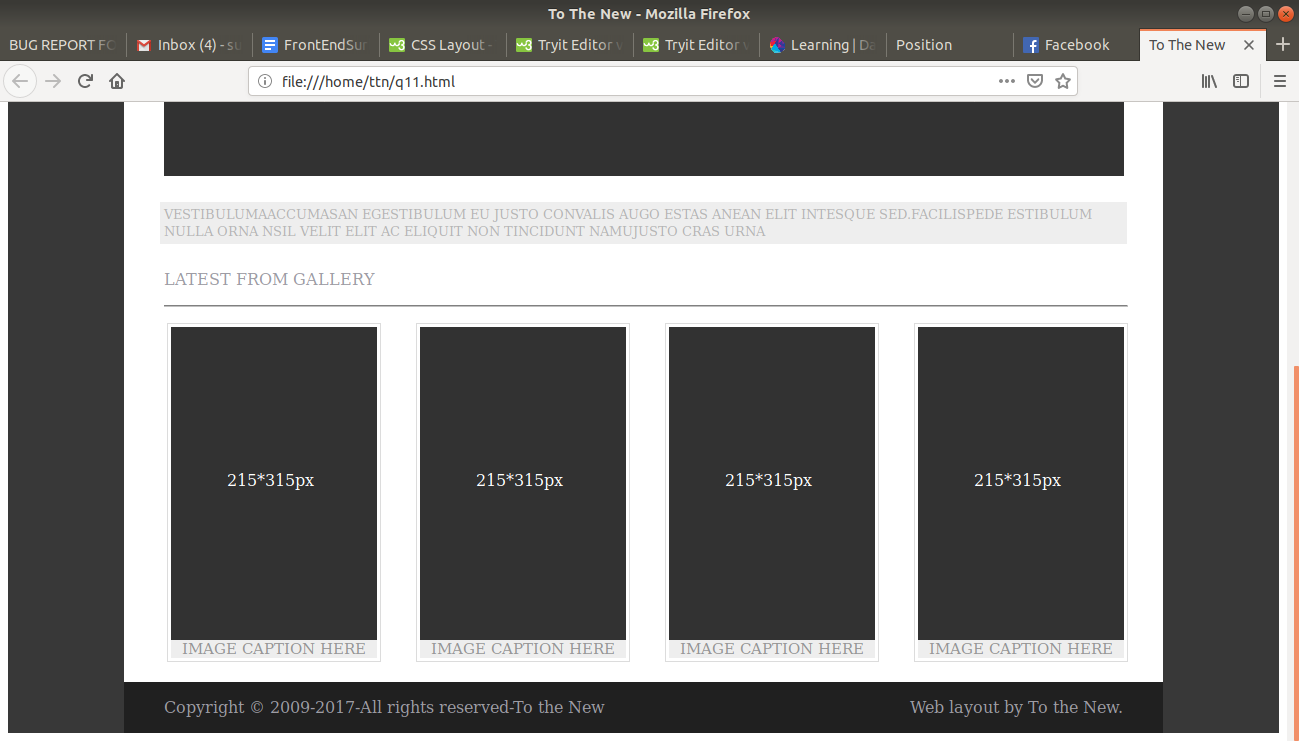
</div>

</body>

</html>

**Output**





**Q12. Create HTML for form.png**

**Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>BUG REPORT FORM</title>

<style>

#main-container

{

width:80%;

margin:0 auto;

color:#1b9596;

}

h2{

display: inline;

}

nav

{

display: inline;

margin-left:680px;

}

nav>a{

color: #1b9596;

text-decoration: none;

}

hr

{

color:rgb(217, 217, 224);

}

#form-title

{

background-color: #bcc3ca;

color:white;

height: 10px;

padding-left: 2em;

padding-top:1em;

padding-bottom: 2em;

border-radius:6px 6px 0px 0px;

}

form

{

color:black;

margin:0 auto;

width:95%

}

.formele

{

width: 100%;

}

select

{

background-color:#1abc9c;

color:white;

padding:0.2em;

}

footer

{

background-color: #edeff1;

padding:1em;

margin-top: 1em;

}

input[type="button"]

{

background-color:#1abc9c;

color:white;

margin-left:90%;

padding: 0.5em;

}

input[placeholder]

{

width:77%;

}

button{

padding: 2px;

background-color:#bdc3c7;

color:white;

}

</style>

</head>

<body>

<div id="main-container">

<h2>TO THE NEW</h2>

<nav>

<a href="#">Home</a>

<a href="#">Quick Help</a>

</nav>

<hr/>

<div id="form-container">

<div id="form-title">

<h2>Bug Report</h2>

</div>

<form action="#" method="#">

Title:\*

<br/>

<input type="text" required class="formele">

<br/>

Description:\*

<br/>

<textarea required class="formele"></textarea>

<br/>

Operating system:

<br/>

<select class="formele">

<option value="WindowsXP">Windows XP</option>

<option value="Ubuntu">Ubuntu</option>

<option value="Vistas">Vistas</option>

</select>

<br/>

Product:

<br/>

<select class="formele">

<option value="formold">Formold</option>

</select>

<br/>

Version:\*<br/>

<input type="text" required style="width:30%;height:20px;">

<br/>

License:

<br/>

<input type="radio" name="license">Free

<input type="radio" name="license" style="margin-left:40%">Business

<br/>

Severity:

<br/>

<select class="formele">

<option value="critical">Critical</option>

</select>

<br/>

Attachment:

<br/>

<input type="text" placeholder="No file choosen">

<button value="Choose File">Choose File</button>

</form>

</div>

<footer>

<input type="button" value="Send">

</footer>

</div>

</body>

</html>

**Output**

