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

An inline element has, but may not be limited to, the following characteristics:

* Flows along with text content, thus
* Will not clear previous content to drop to the next line like block elements
* Is subject to white-space settings in CSS
* Will ignore top and bottom margin settings, but will apply left and right margins, and any padding
* Will ignore the width and height properties
* If floated left or right, will automatically become a block-level element, subject to all block characteristics
* Is subject to the vertical-align property

Examples of Inline Elements:

<a>, <span>, <b>, <em>, <i>, <cite>, <mark>, and <code>.

A block element is an element that has the following characteristics:

* If no width is set, will expand naturally to fill its parent container
* Can have margins and/or padding
* If no height is set, will expand naturally to fit its child elements (assuming they are not floated or positioned)
* By default, will be placed below previous elements in the markup (assuming no floats or positioning on surrounding elements)
* Ignores the vertical-align property

Examples of Block Elements:

<p>, <div>, <form>, <header>, <nav>, <ul>, <li>, and <h1>.

<html>

<head>

<title>

Exercise

</title>

</head>

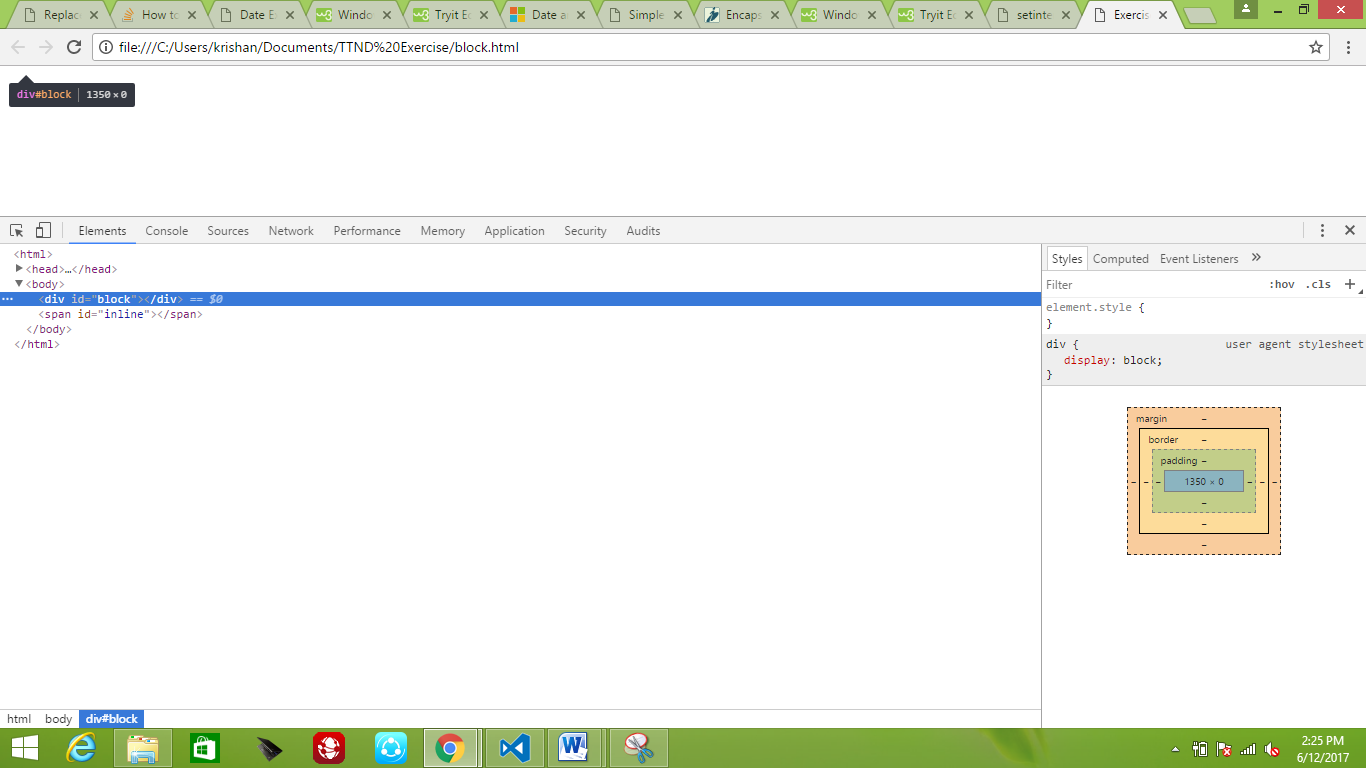
<body>

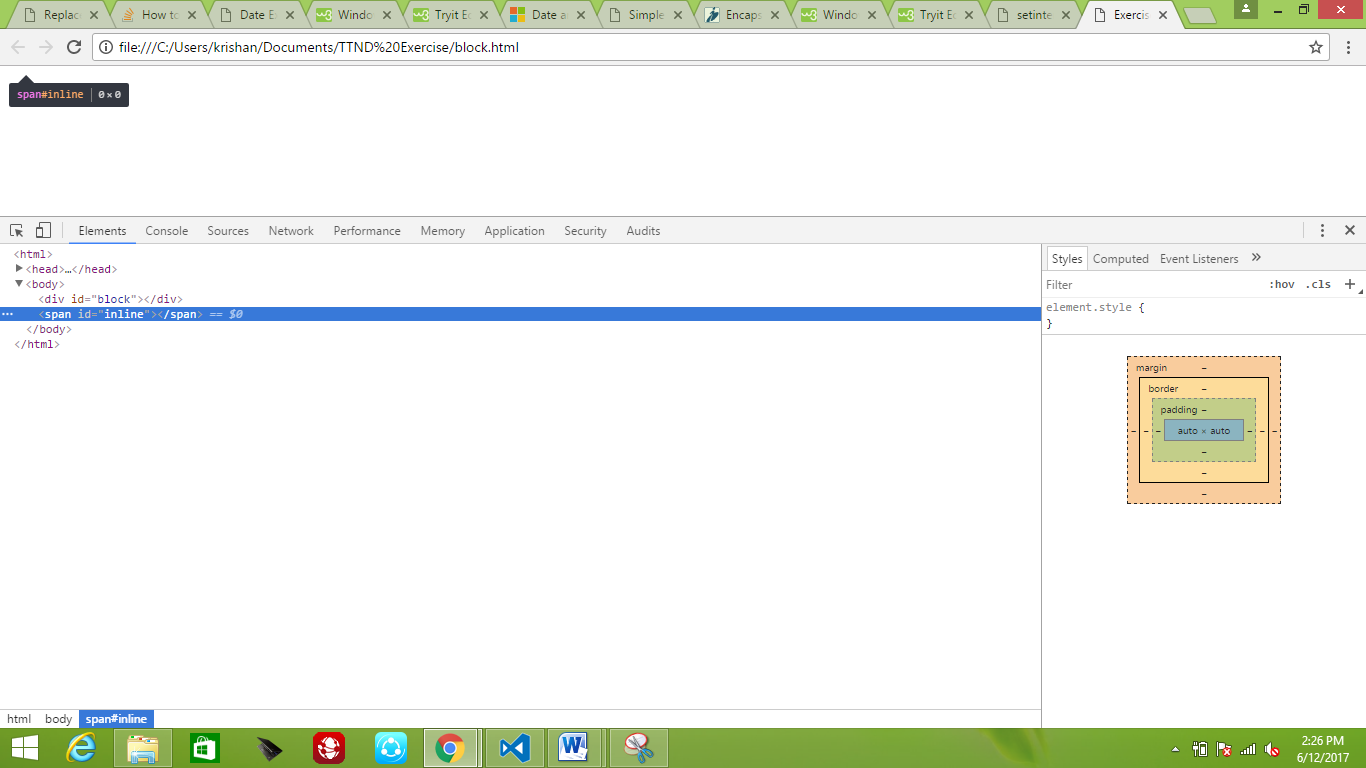
<div id="block"></div>

<span id="inline"></span>

</body>

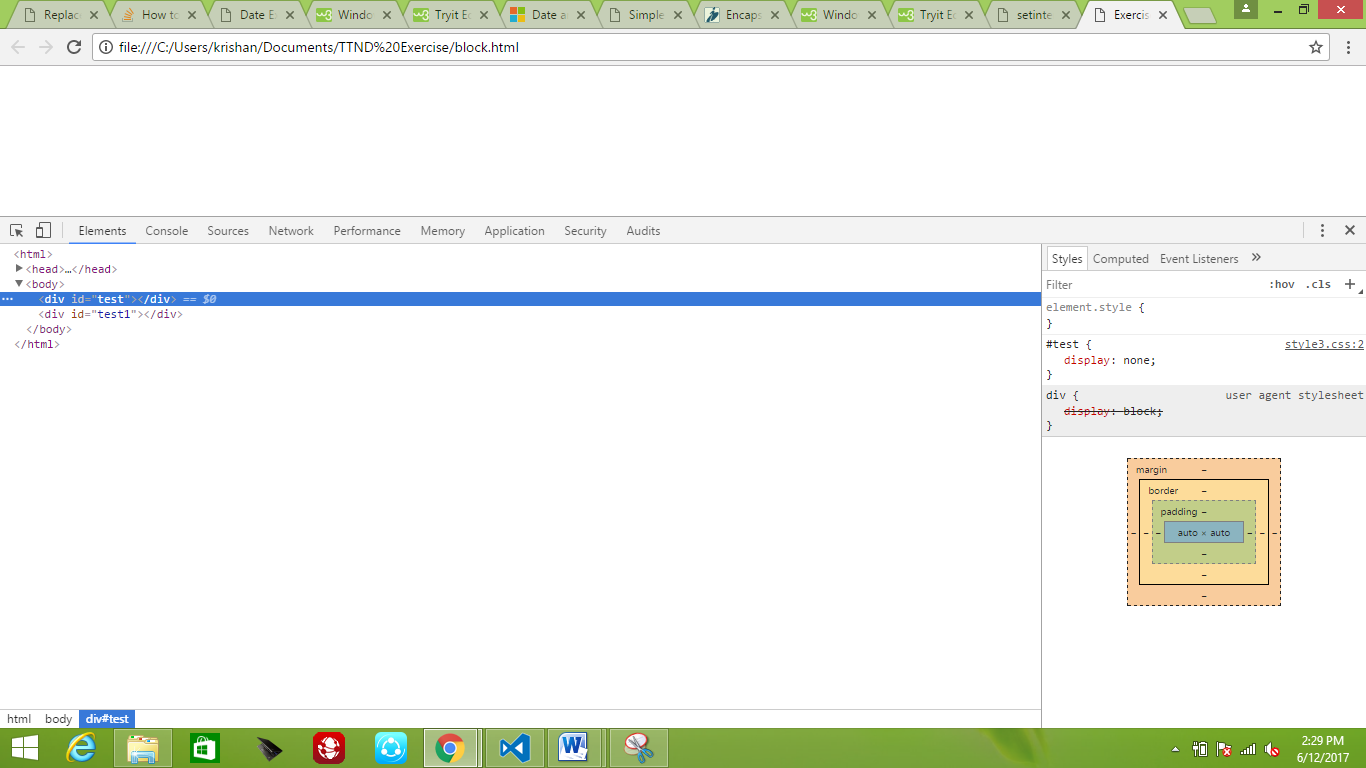
</html>



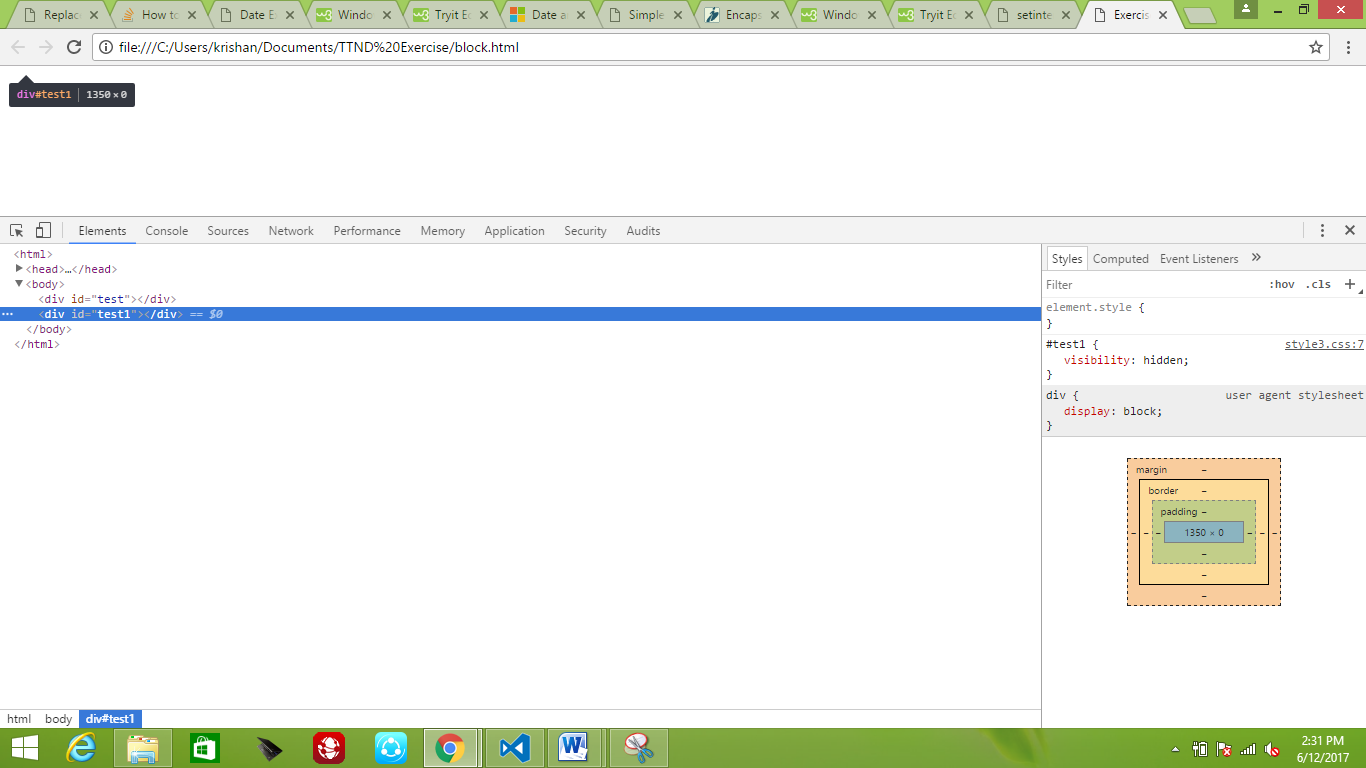


**2.Explain the difference between visibility:hidden and display:none**

**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 specifies whether or not an element should float.

In its simplest use, the float property can be used to wrap text around images.

The clear property is used to control the behavior of floating elements.

Elements after a floating element will flow around it. To avoid this, use the clear property.

The clear property specifies on which sides of an element floating elements are not allowed to float.

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

**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

**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 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 it's position 10 pixels down from where it would normally be.

It introduces the ability to use z-index on that element, which doesn't really work with statically positioned elements. Even if you don't set a z-index value, this element will now appear on top of any other statically positioned element. You can't fight it by setting a higher z-index value on a statically positioned element.

**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

**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. The only reason you would ever set an element to position: static is to forcefully-remove some positioning that got applied to an element outside of your control. This is fairly rare, as positioning doesn't cascade.

**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.**

<html>

<head>

<link rel="stylesheet" href="style.css">

<title>

Title Example

</title>

</head>

<body>

<table class = "boundary">

<thead>

<th>ID</th>

<th>Employee Name</th>

<th>Designation</th>

<th>Department</th>

</thead>

<tbody>

<tr>

<td>1</td>

<td>A</td>

<td>Manager</td>

<td>Information Technology</td>

</tr>

<tr>

<td>2</td>

<td>B</td>

<td>HR</td>

<td>Information Technology</td>

</tr>

<tr>

<td>3</td>

<td>C</td>

<td>Employee</td>

<td>Information Technology</td>

</tr>

<tr>

<td>4</td>

<td>D</td>

<td>Manager</td>

<td>Marketing</td>

</tr>

<tr>

<td>5</td>

<td>E</td>

<td>HR</td>

<td>Marketing</td>

</tr>

<tr>

<td>6</td>

<td>F</td>

<td>Employee</td>

<td>Marketing</td>

</tr>

</tbody>

</table>

</body>

</html>

**Style.css**

table, th, td

{

border: 1px solid black;

border-collapse: collapse;

padding: 5px 10px;

text-align: left;

}

td

{

color:black;

}

th

{

background-color: black;

color: white;

}

tr:nth-child(even)

{

background-color:#fff;

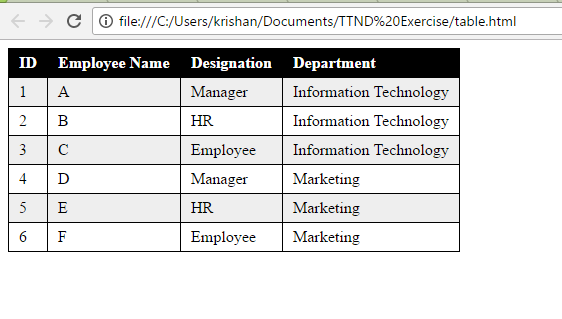
}

tr:nth-child(odd)

{

background-color:#eee;

}

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

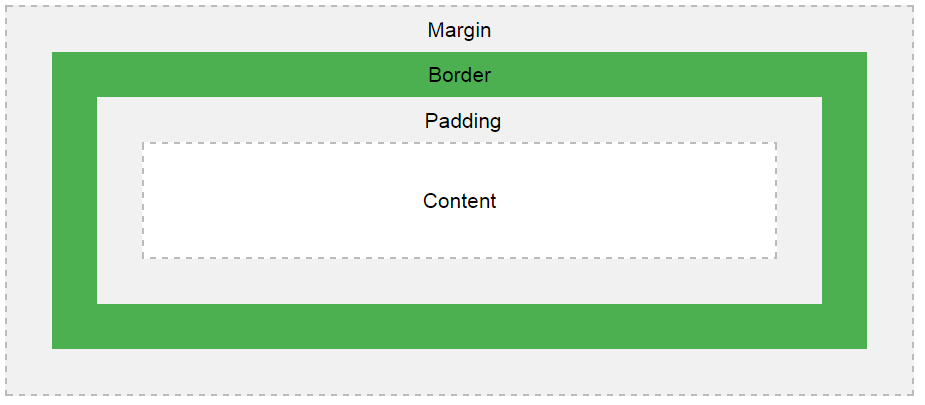
Metadata is data (information) about data.

* 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.
* 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

**7. Explain box model.**

All HTML elements can be considered as boxes.

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:



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

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?**

There are three ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style

**External Style Sheet**

With an external style sheet, you can change the look of an entire website by changing just one file.

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

**<head>**

**<link rel="stylesheet" type="text/css" href="mystyle.css">**

**</head>**

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.

**Internal Style Sheet**

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

<head>

<style>

body {

background-color: linen;

}

h1 {

color: maroon;

margin-left: 40px;

}

</style>

</head>

**Inline Styles**

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

**9. Define Doctype.**

Basically, the DOCTYPE describes the HTML that will be used in your page.

Browsers also use the DOCTYPE to determine how to render a page. Not including a DOCTYPE or including an incorrect DOCTYPE can trigger quirks mode. The kicker here is that quirks mode in Internet Explorer is quite different from quirks mode in Firefox (and other browsers), meaning that you'll have a *much* harder job trying to ensure your page works consistently in all browsers if pages are rendered in quirks mode than you will if they are rendered in standards mode.

**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**.**

**Section:**

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.

**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.

**Footer**

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.

**Figure**

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:

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

**Web1.html**

<html>

<head>

<link rel="stylesheet" href="style2.css">

<title>Web</title>

</head>

<body>

<div id="container">

<div id = "header">

To The New

<div id = "menu">

<a href="">TEXT LINK</a>

<a href="">TEXT LINK</a>

<a href="">TEXT LINK</a>

<a href="">TEXT LINK</a>

<a href="">TEXT LINK</a>

</div>

</div>

<div id = "body">

<div id="main\_image">

<img src="" alt="960 X 360px">

</div>

<div id="text">

VESTIBULUMACCUMSAN EGESTIBULUM EU JUSTO CONVALLIS AUGUE ESTAS AENEAN ELIT INTESQUE SED.<br/>

FACILISPEDE ESTIBULUM NULLA ORNA NISL VELIT ELIT AC ALIQUAT NON TINCIDUNT NAMJUSTO CRAS URNA.

</div>

LATEST FROM THE GALLERY<br/>

<hr/>

<div id="latest\_gallery">

<span><img src="" alt="215 X 315px"></span>

<span><img src="" alt="215 X 315px"></span>

<span><img src="" alt="215 X 315px"></span>

<span><img src="" alt="215 X 315px"></span>

</div>

</div>

<div id = "footer">

Copyright &copy; 2009-2017 - All Rights Reserved - To the new

</div>

</div>

</body>

</html>

**Style2.css**

body

{

background-color: #1C2833 ;

}

#container

{

background-color:white ;

margin: 0px 90px;

}

#body

{

margin: 0px 90px;

}

#header

{

background-color: #212F3C ;

color:white;

font-weight: bold;

font-size: 25px;

height:70px;

padding:0px 90px;

}

#menu

{

background-color: #17202A;

}

#menu a

{

color:white;

font-size:14px;

padding:8px;

}

#main\_image img

{

width:960px;

height:360px;

margin:20px 0px;

}

#text

{

background-color: #D7DBDD;

padding: 20px 0px;

margin:10px 0px 20px;

}

div#latest\_gallery span img

{

width: 215px;

height: 315px;

margin:20px 12px;

}

div img

{

background-color: #212F3C ;

height: 360px;

line-height: 360px;

text-align: center;

color:white;

}

span img

{

background-color: #212F3C ;

height: 315px;

line-height: 315px;

text-align: center;

color:white;

}

#footer

{

background-color: #212F3C ;

color:white;

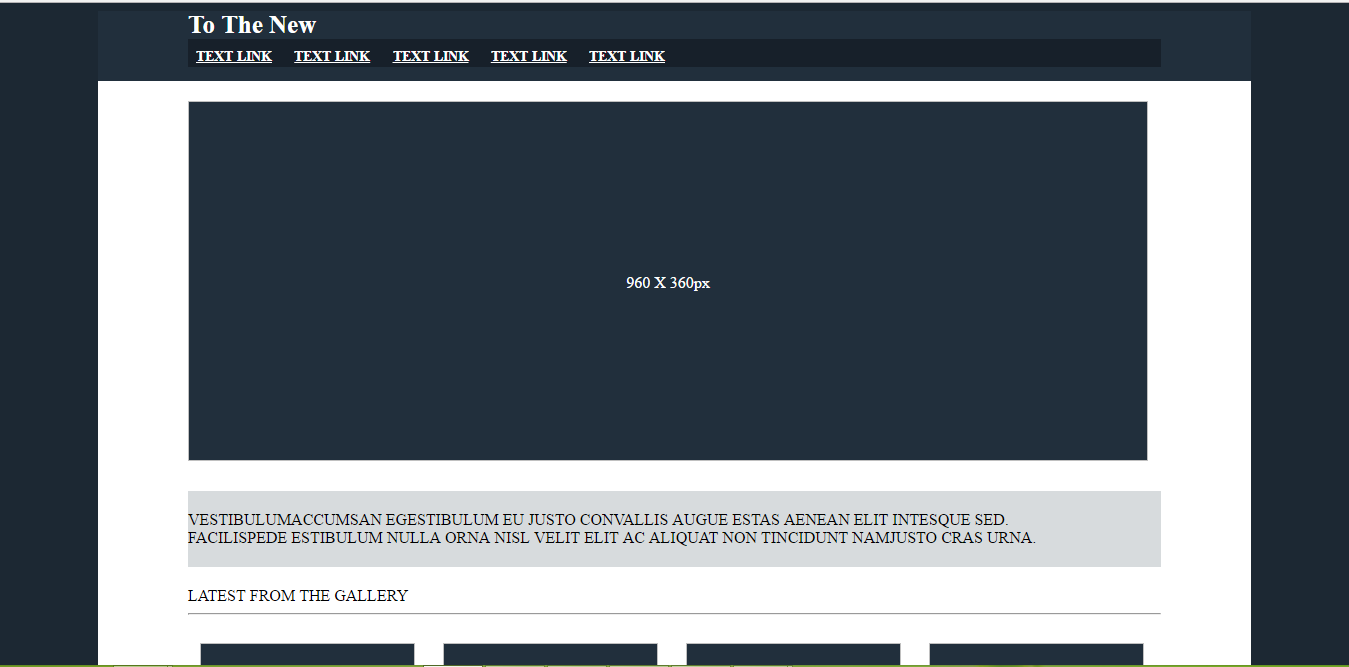
font-weight: light;

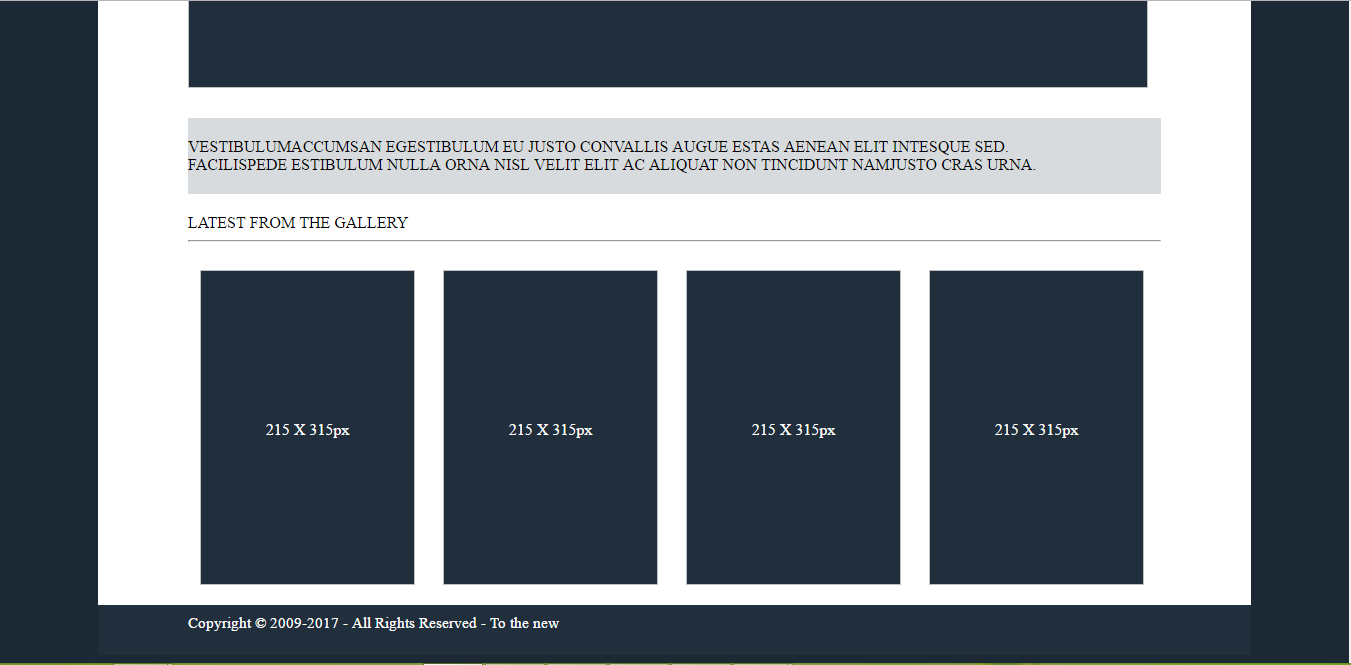
font-size: 15px;

height:30px;

padding:10px 90px;

}





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

<html>

<head>

<link rel="stylesheet" href="style1.css">

<title>

webForm

</title>

</head>

<body>

<div id = "space"> </div>

<div id = heading> TO THE NEW

<span id="right\_header">

<span id=home> Home </span>

<span id=help>Quick Help</span>

</span>

</div>

<hr/>

<form id = "form\_field">

<div id = "form\_container">

<div class = "form\_header">

<spam id = "bug"> Bug Report </spam>

</div>

<div id ="form\_input">

Title:\*<br/>

<input type="text" name="title" required><br/><br/>

Description:\*<br/>

<input type="textarea" rows="4" cols = "20" required><br/><br/>

Operating systems:<br/>

<select name="os" id="menu">

<option value="windows\_xp" selected>Windows XP</option>

<option value=""></option>

<option value=""></option>

<option value=""></option>

</select><br/><br/>

Product:<br/>

<select name="product" id="menu">

<option value="formoid" selected> Formoid</option>

<option value=""></option>

<option value=""></option>

<option value=""></option>

</select><br/><br/>

Version:\*<br/>

<input type="text" name= "version"><br/><br/>

License:<br/>

<span style="float:left;"><input type="radio" name="license" value="free" checked> Free<br></span>

<span style="float:right;"><input type="radio" name="license" value="business" > Business<br></span>

<br/><br/>

Severity:<br/>

<select name="severity" id="menu">

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

<option value=""></option>

<option value=""></option>

<option value=""></option>

</select><br/><br/>

Attachments:<br/>

<!-- <input type="text" placeholder="No file selected">-->

<input type="file"><br/><br/>

</div>

<div class = "form\_footer">

<button type = "submit">Send</button>

</div>

</div>

</form>

<div id = "space"> </div>

</body>

</html>

#space

{

height:100px;

background-color:#EBF5FB ;

}

input[type = text],select,input[type=textarea]

{

width:280px;

height:30px;

align-self: center;

border-radius: 3px;

}

#heading

{

margin:15px;

color:#138D75 ;

font-weight: bolder;

}

span#right\_header

{

float:right;

padding:4px;

}

span#home

{

float:left;

padding :2px 4px;

}

span#help

{

float:right;

padding: 2px 4px;

}

#menu

{

background-color: #17A589;

color:white;

width:280px;

height:25px;

}

#form\_container

{

width:350px;

border-radius: 3px;

margin:auto;

}

#menu, input[type=text],input[type=textarea]

{

width:320px;

border-radius: 3px;

padding: 10px 10px;

}

#form\_container

{

border-style: solid;

border-width: 1px;

height:auto;

}

.form\_header, .form\_footer

{

height:50px;

color:white;

padding:0px;

}

spam#bug {

font-size: 22px;

}

.form\_header

{

background-color: #D5D8DC ;

width:350px;

}

.form\_footer

{

background-color: #EAECEE;

width:350px;

}

.form\_footer button

{

float: right;

margin:10px 5px;

background-color: #17A589;

color:white;

}

#form\_field

{

text-align: left;

}

input[type = radio]

{

height:1.5em;

width:100%;

}

#form\_input

{

margin:10px;

}

span input[type=radio]

{

width:35px;

height:20px;

padding: 6px;

display:inline;

}

