

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

Every HTML element has a default display value depending on what type of element it is. This value can either be **block** or **inline**.

1) block - 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).

Ex. <div>, <hr>, <form>, <p>, <section> etc.

2) inline - An inline element does not start on a new line and only takes up as much width as necessary.

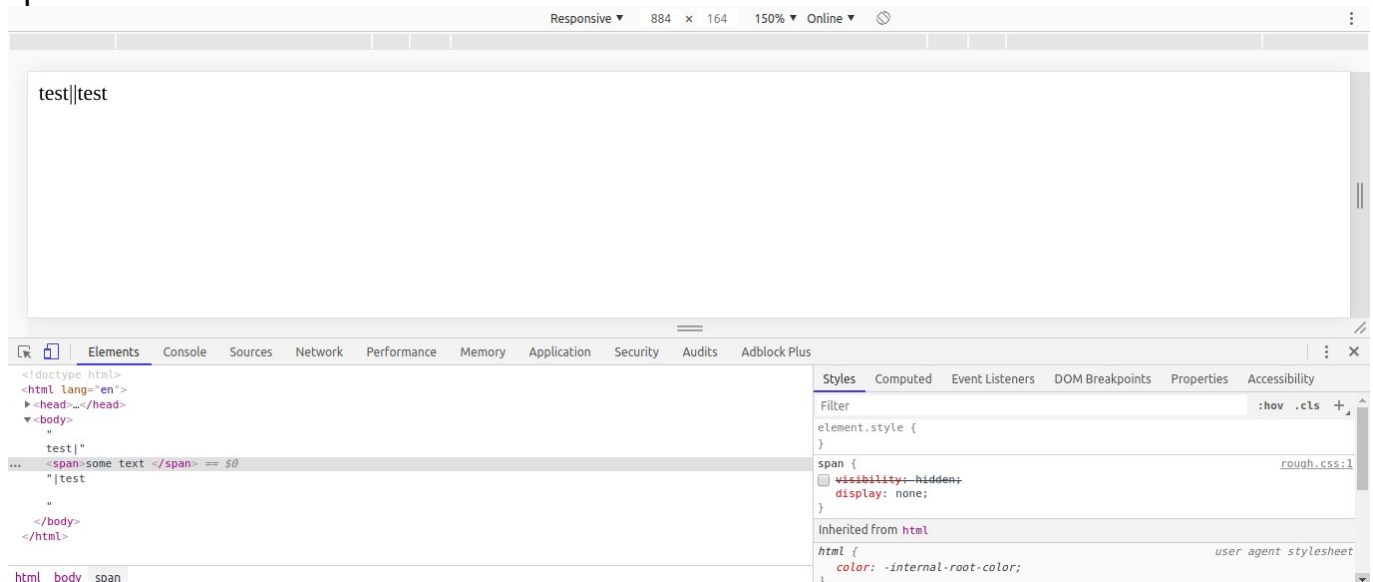
Ex. <a>, <span>, <b>, <textarea>, <strong> etc.

<p> is a block level element, so it takes up full width.

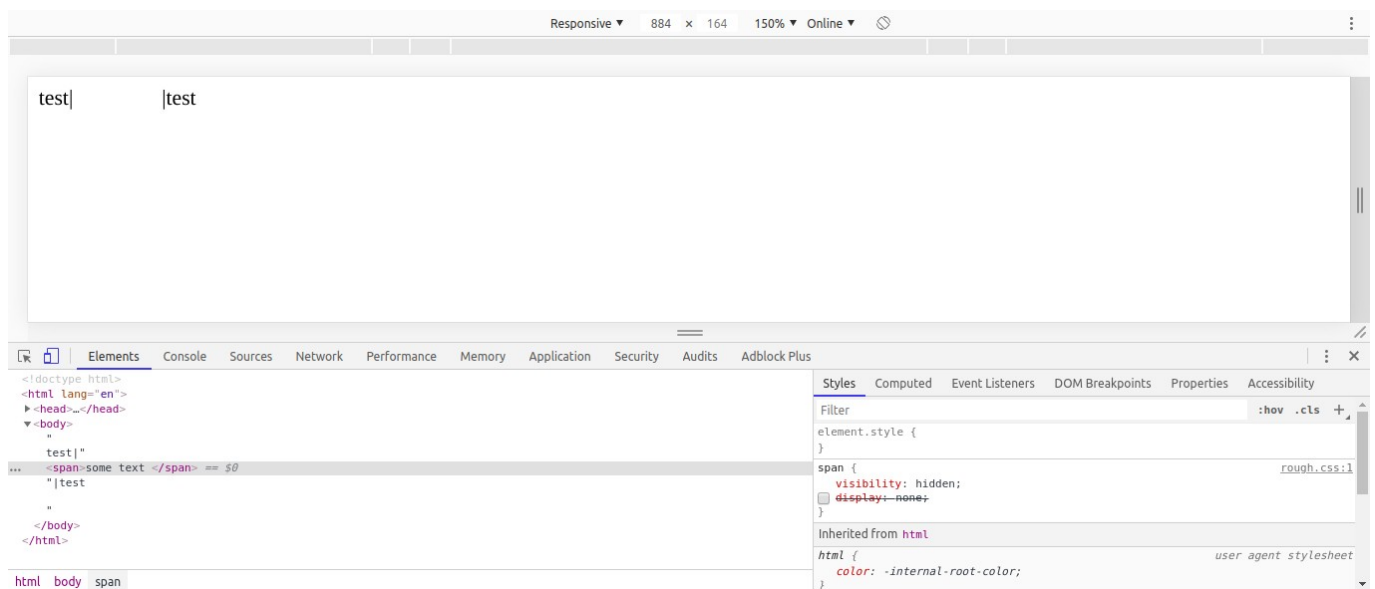
<span> is a inline element and only takes up as much width as necessary.

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

1) display:none – means that the tag will not appear on the page at all. There will be no space allocated for it as well.



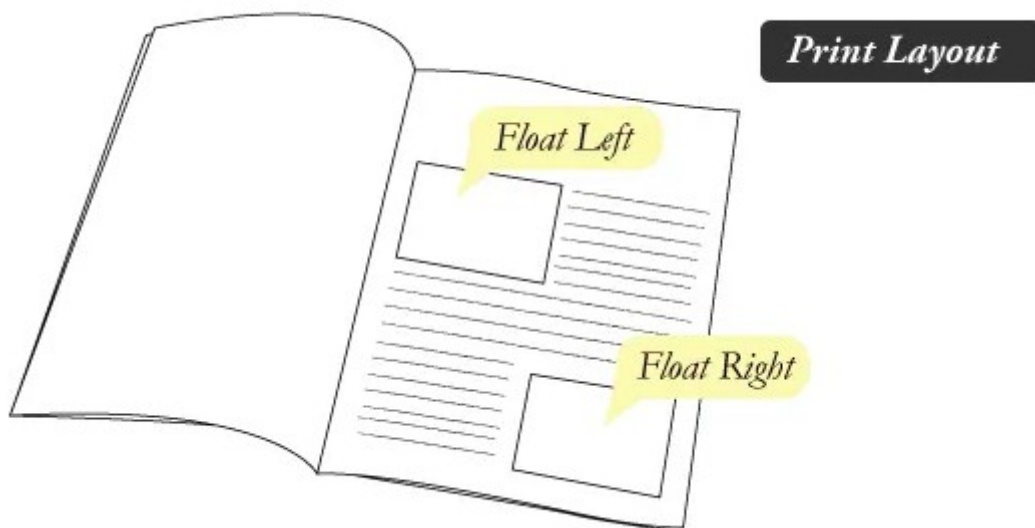
2) visibility:hidden – Tag is not visible but space is allocated for it on the page.



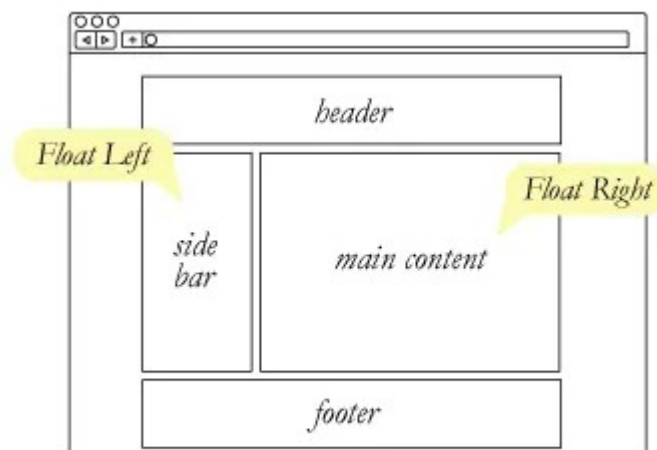
### 3. Explain the clear and float properties.

Float is a CSS positioning property. As the name suggests when applied to an element, the element 'floats' above other elements making a empty space. All other following elements move up and take up this empty space except text, text wraps around the floating element. Where we want the element to float can be specified by left of right.

Simplest use of float can be to wrap text around images as in a print layout. This is commonly known as **text wrap**.



Float can also be used to achieve such a behavior.



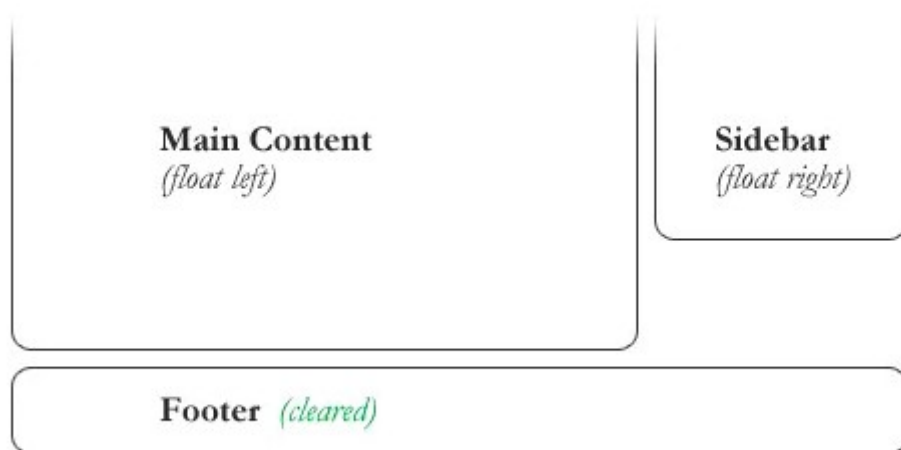
Clear is used so that an element is not affected by the empty space created by a floating element.

Clear left will not allow floating elements on its left side and clear right will not allow floating elements on its right side. To clear both the sides use both.

Ex.



The footer moves up and occupies the space created by floating elements. This can be avoided by setting its clear property as both.



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

The position property specifies the type of positioning method used for an element. Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set to anything except relative which is the default. They work differently depending on the position value.

##### 1) static

HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties, they follow the normal flow of the page.

##### 2) relative

Used to position an element '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.

##### 3) fixed

Used to position elements relative to the view port, 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.

##### 4) 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.

A positioned element is one whose position is anything except static(default).

##### 5) sticky

This is a hybrid between fixed and relative.

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.

ID	Employee	Designation	Department
1	Chirag	Trainee	JVM
2	Ajay	Trainee	JVM
3	Harshit	SDE 1	DevOps
4	Elon	SDE 2	FEEN
5	Deepak Mittal	CEO	Management
6	Satya Sheel Sharma	CHRO	HR

Code is available in the repository (q5.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.

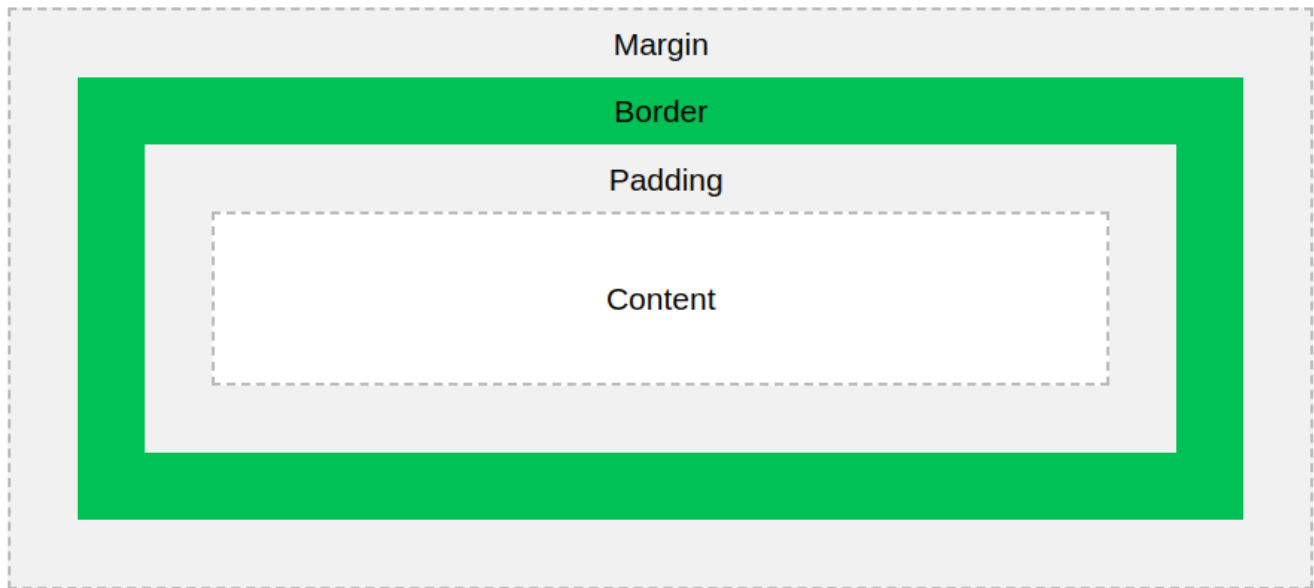
The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

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. 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.



Now using CSS we can style each of these.

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

CSS selectors are used to "find" (or select) the HTML elements you want to style. We can divide CSS selectors into five categories:

- 1) Simple selectors (select elements based on name, id, class)
- 2) Combinator selectors (select elements based on a specific relationship between them)
- 3) Pseudo-class selectors (select elements based on a certain state)
- 4) Pseudo-elements selectors (select and style a part of an element)
- 5) Attribute selectors (select elements based on an attribute or attribute value)

## 9. Define Doctype.

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.

For a HTML 5 page use `<!DOCTYPE html>`

## 10. Explain 5 HTML5 semantic tags.

HTML 5 introduced new tags whose meaning are clear to both the developer and browser. The word semantic literally means 'relating to meaning'.

5 semantic HTML tags are as follows :

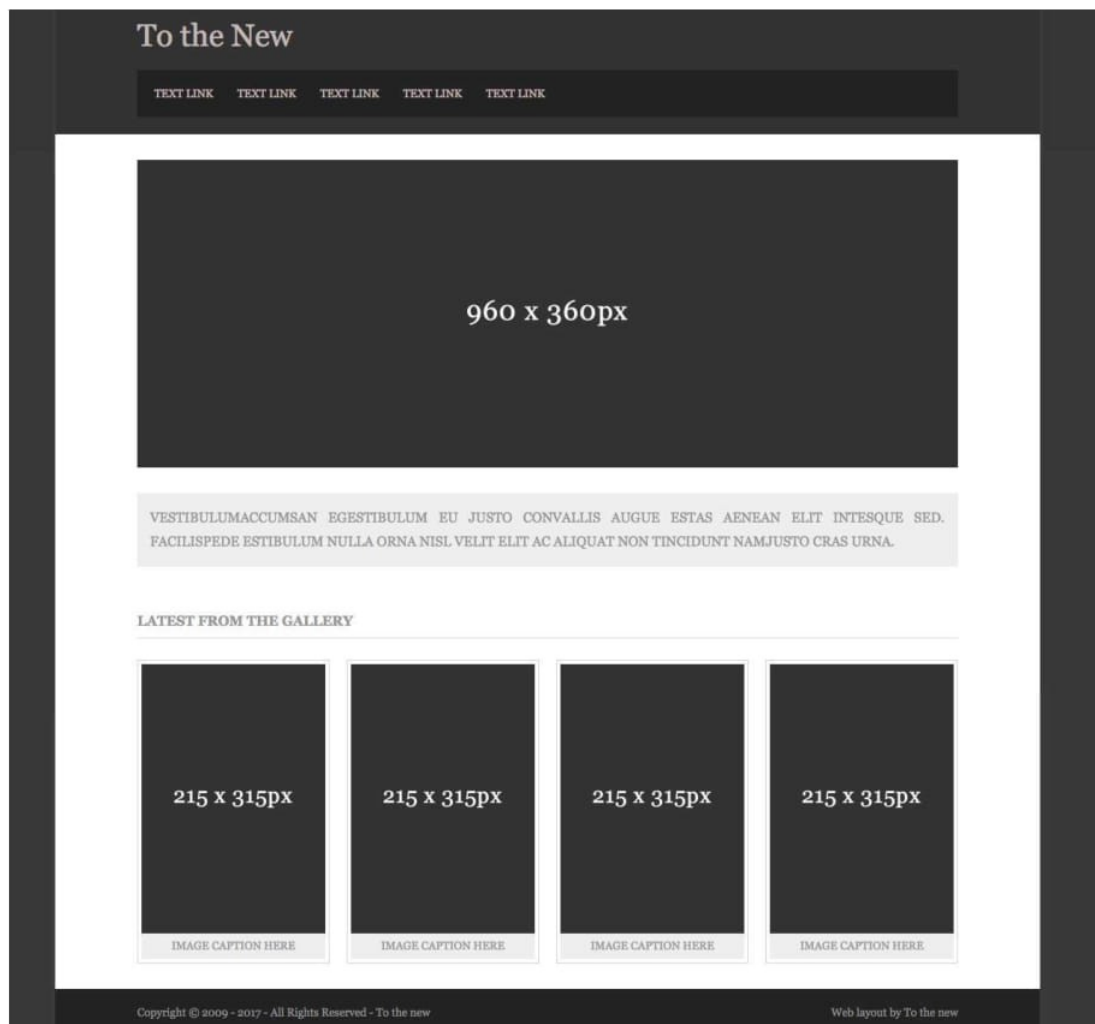
- 1) <section> - defines a section in a document.
- 2) <header> - specifies a header for a document or section.
- 3) <footer> - specifies a footer for a document or section.
- 4) <nav> - defines a set of navigation links.
- 5) <figcaption> - caption a image.

### Why semantic tags?

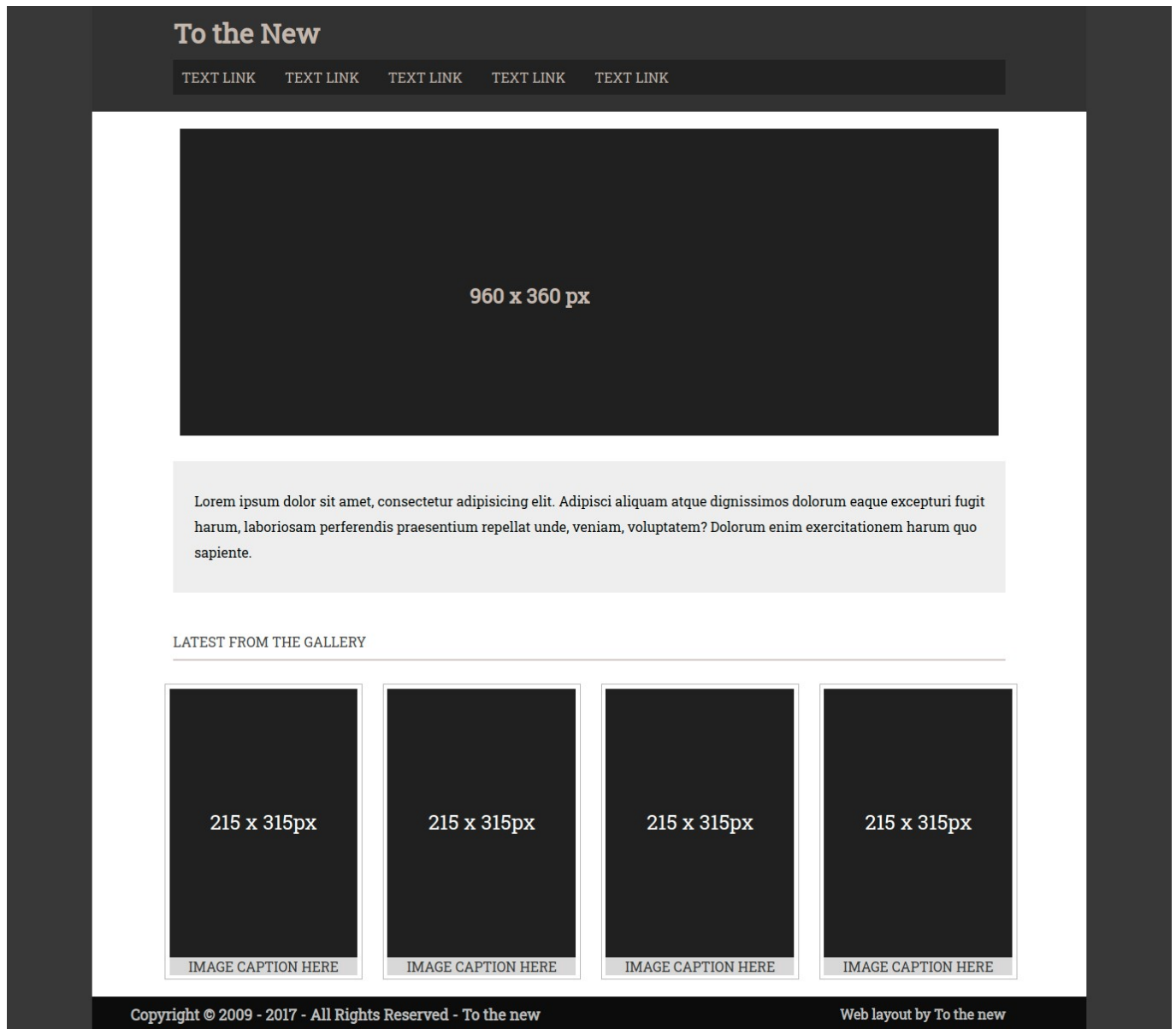
With HTML4, developers used their own id/class names to style elements: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc. This made it impossible for search engines to identify the correct web page content. With the new HTML5 elements (<header> <footer> <nav> <section> <article>), this will become easier.

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

### Expected



# Actual



Code is available in the repository. (q11.html, q11.css)



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

## Expected

TO THE NEW

[Home](#) [Quick Help](#)

### Bug Report

Title:\*

Description:\*

Operating system:  

Windows XP

Product:\*  

Formoid

Version:\*

License:  
☐ Free ☐ Business

Severity:  

Critical

Attachments:  

No file selected

Choose File

Send

### Bug Report

Title:\*

Description:\*

Operating system:

Windows XP

Product:\*

Formoid

Version:\*

License:

☒ Free ☐ Business

Severity

Critical

Attachments

No file chosen

Choose file

Send

**Code is available in the repository. (q12.html, q12.css)**