Assignment

HTML AND CSS

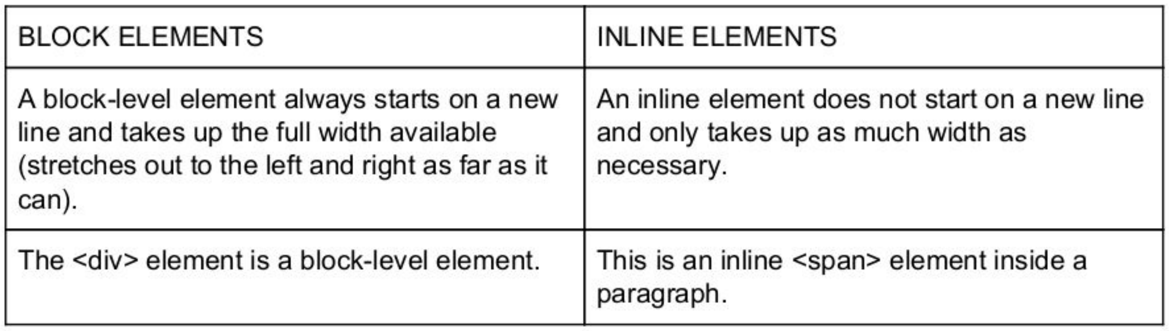
Q1.How are inline and block elements different from each other?

Answer:

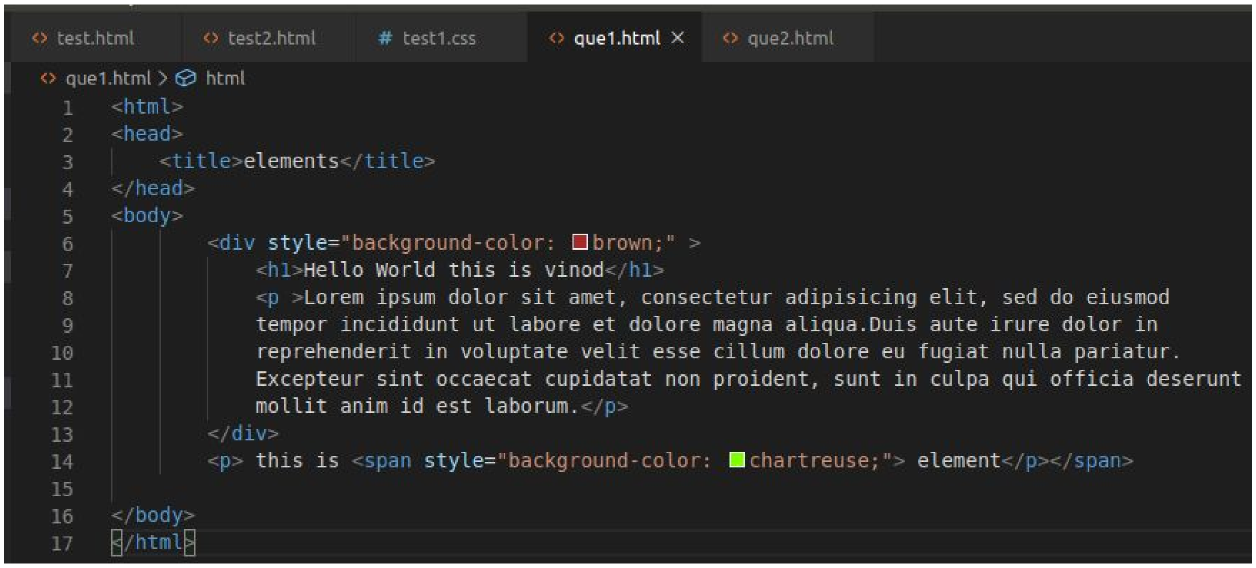
a <span> element is used as an inline element and a <div> element as a block level element.

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.

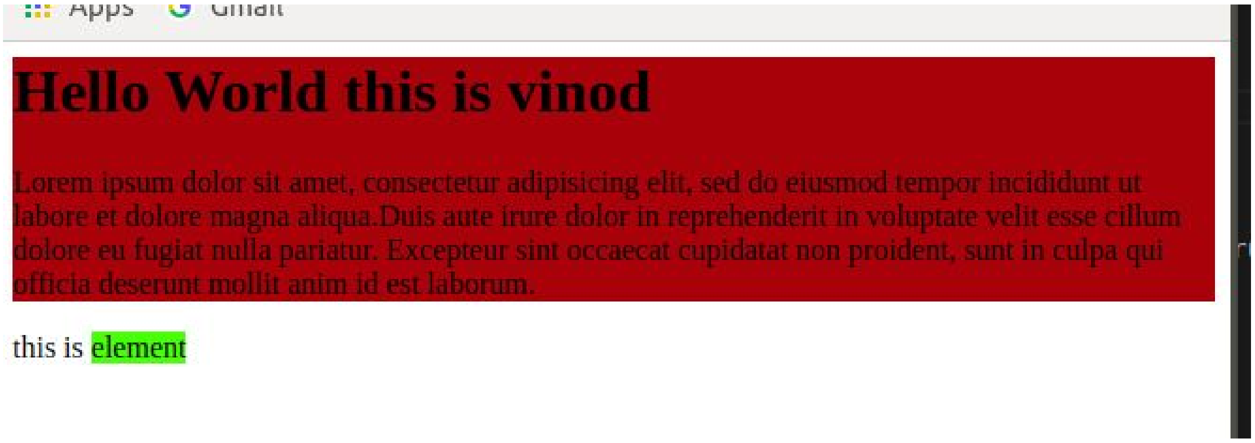
A block-level element always starts on a new line and takes up the full width of a page, from left to right. A block-level element can take up one line or multiple lines and has a line break before and after the element.



BLOCK ELEMENTS	INLINE ELEMENTS
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).	An inline element does not start on a new line and only takes up as much width as necessary.
The <div> element is a block-level element.	This is an inline <span> element inside a paragraph.



```html
<html>
<head>
    <title>elements</title>
</head>
<body>
        <div style="background-color: brown;" >
            <h1>Hello World this is vinod</h1>
            <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
            tempor incididunt ut labore et dolore magna aliqua.Duis aute irure dolor in
            reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
            Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt
            mollit anim id est laborum.</p>
        </div>
        <p> this is <span style="background-color: chartreuse;"> element</p></span>

</body>
</html>
```



Hello World this is vinod

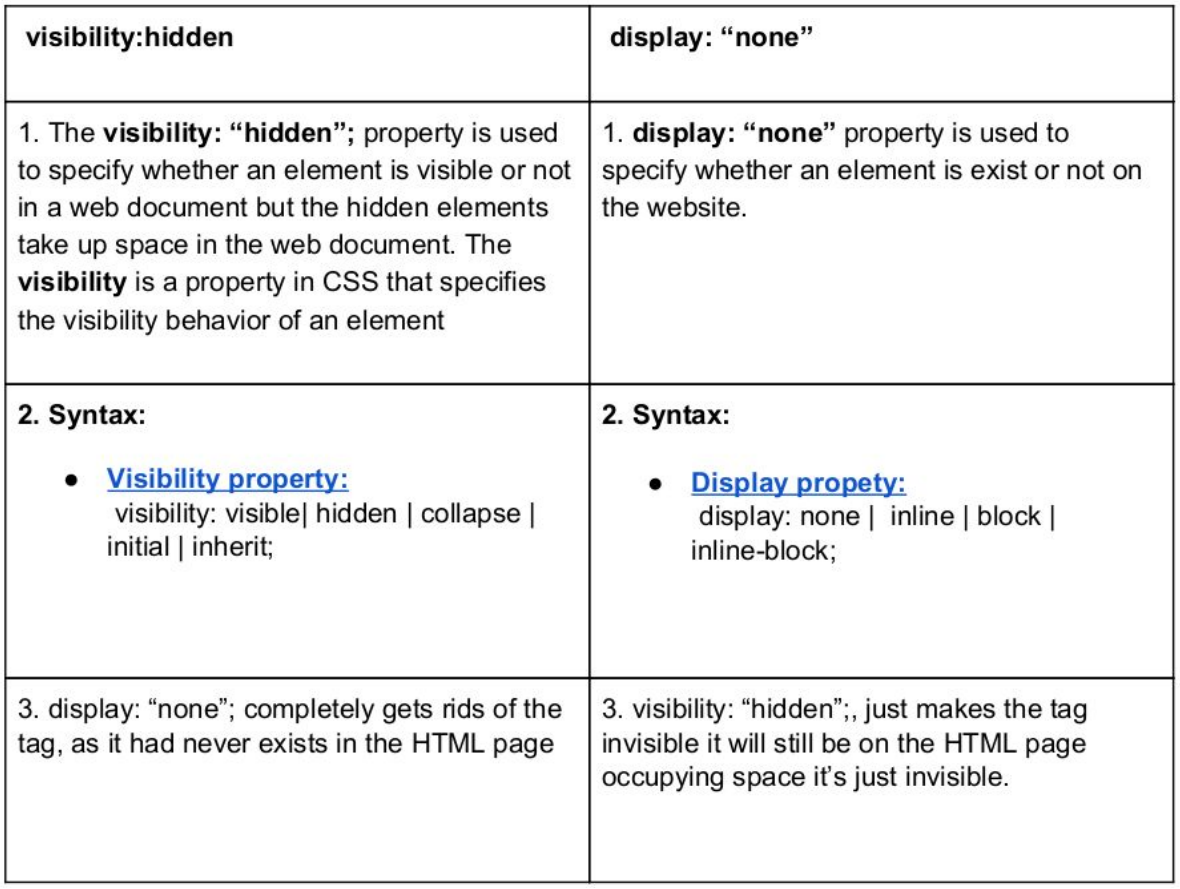
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

this is element

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

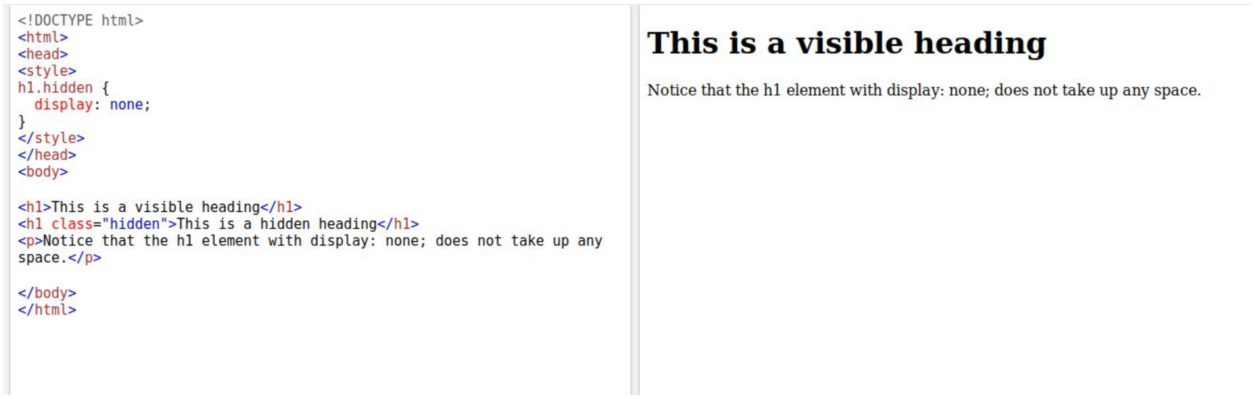
Answer:

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them.



visibility:hidden	**display: “none”**
1. The **visibility: “hidden”;** property is used to specify whether an element is visible or not in a web document but the hidden elements take up space in the web document. The **visibility** is a property in CSS that specifies the visibility behavior of an element	1. **display: “none”** property is used to specify whether an element is exist or not on the website.
2. Syntax: • **Visibility property:** visibility: visible| hidden | collapse | initial | inherit;	**2. Syntax:** • **Display propety:** display: none | inline | block | inline-block;
3. display: “none”; completely gets rids of the tag, as it had never exists in the HTML page	3. visibility: “hidden”;, just makes the tag invisible it will still be on the HTML page occupying space it’s just invisible.

The <script> element uses display: none; as default.



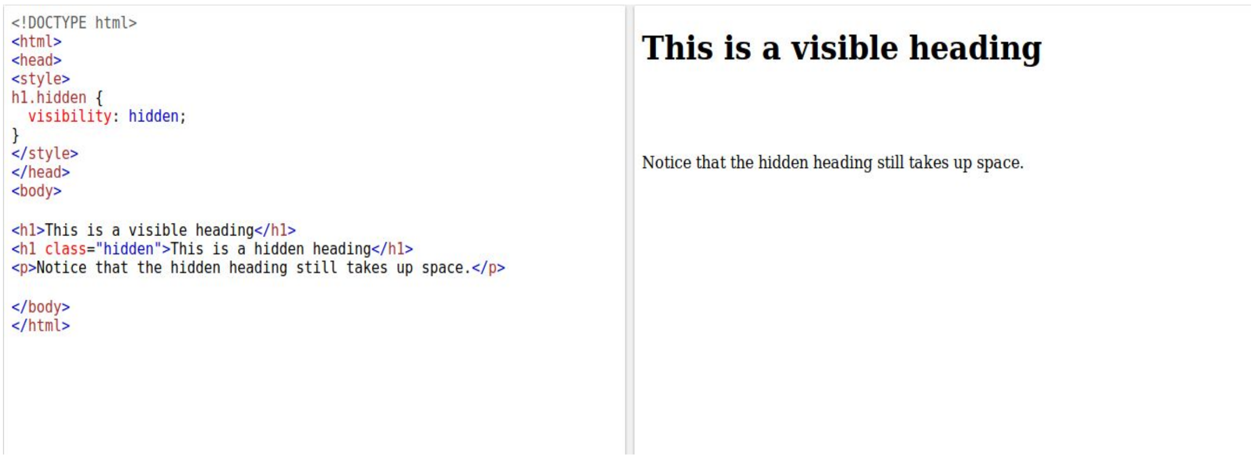
```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  display: none;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the h1 element with display: none; does not take up any
space.</p>

</body>
</html>
```

This is a visible heading

Notice that the h1 element with display: none; does not take up any space.

visibility:hidden; also hides an element.

The element will still take up the same space as before. The element will be hidden, but still affect the layout:



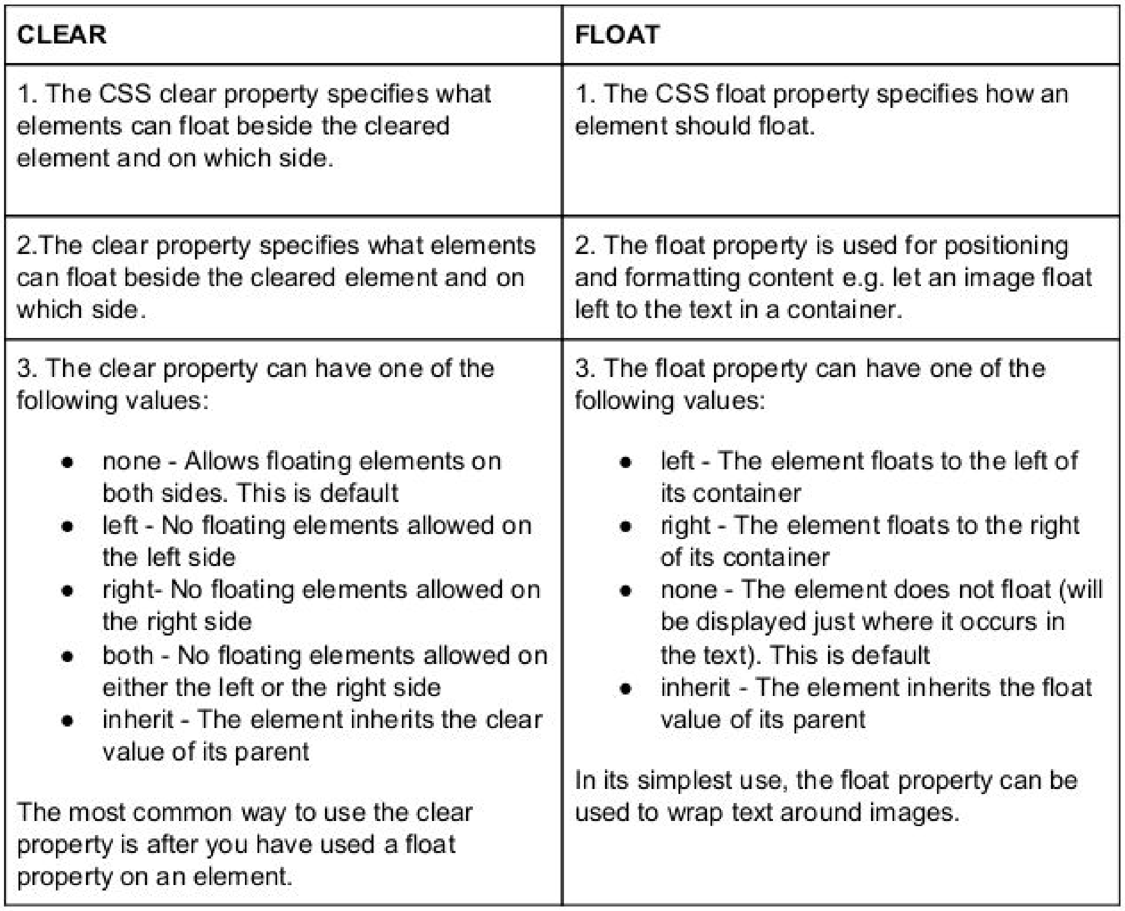
```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  visibility: hidden;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>

</body>
</html>
```

This is a visible heading

Notice that the hidden heading still takes up space.

Q3. Explain the clear and float properties.

Answer:

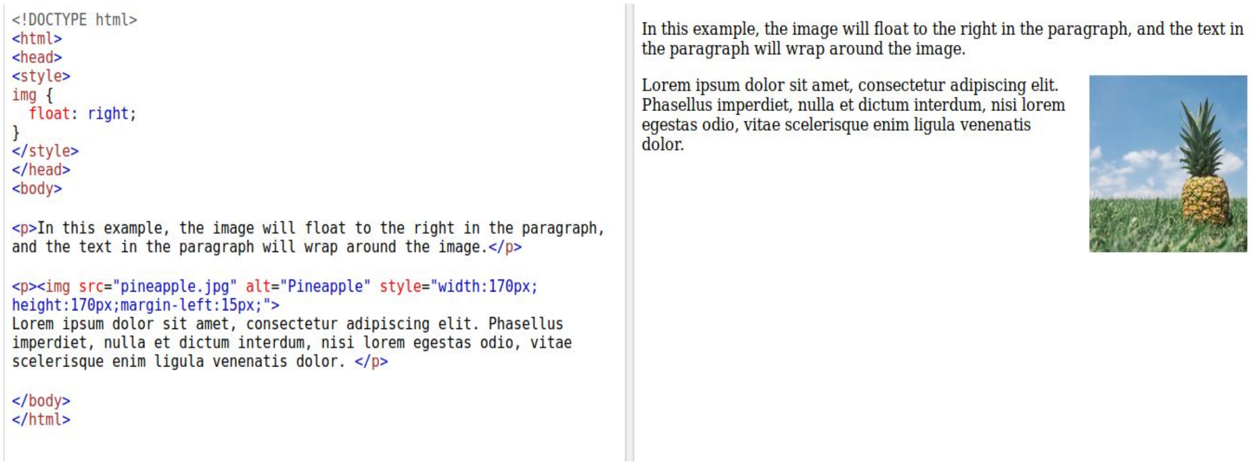


CLEAR	FLOAT
1. The CSS clear property specifies what elements can float beside the cleared element and on which side.	1. The CSS float property specifies how an element should float.
2.The clear property specifies what elements can float beside the cleared element and on which side.	2. The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.
3. 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.	3. 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 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



```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>

<p>In this example, the image will float to the right in the paragraph,
and the text in the paragraph will wrap around the image.</p>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;
height:170px;margin-left:15px;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae
scelerisque enim ligula venenatis dolor. </p>

</body>
</html>
```

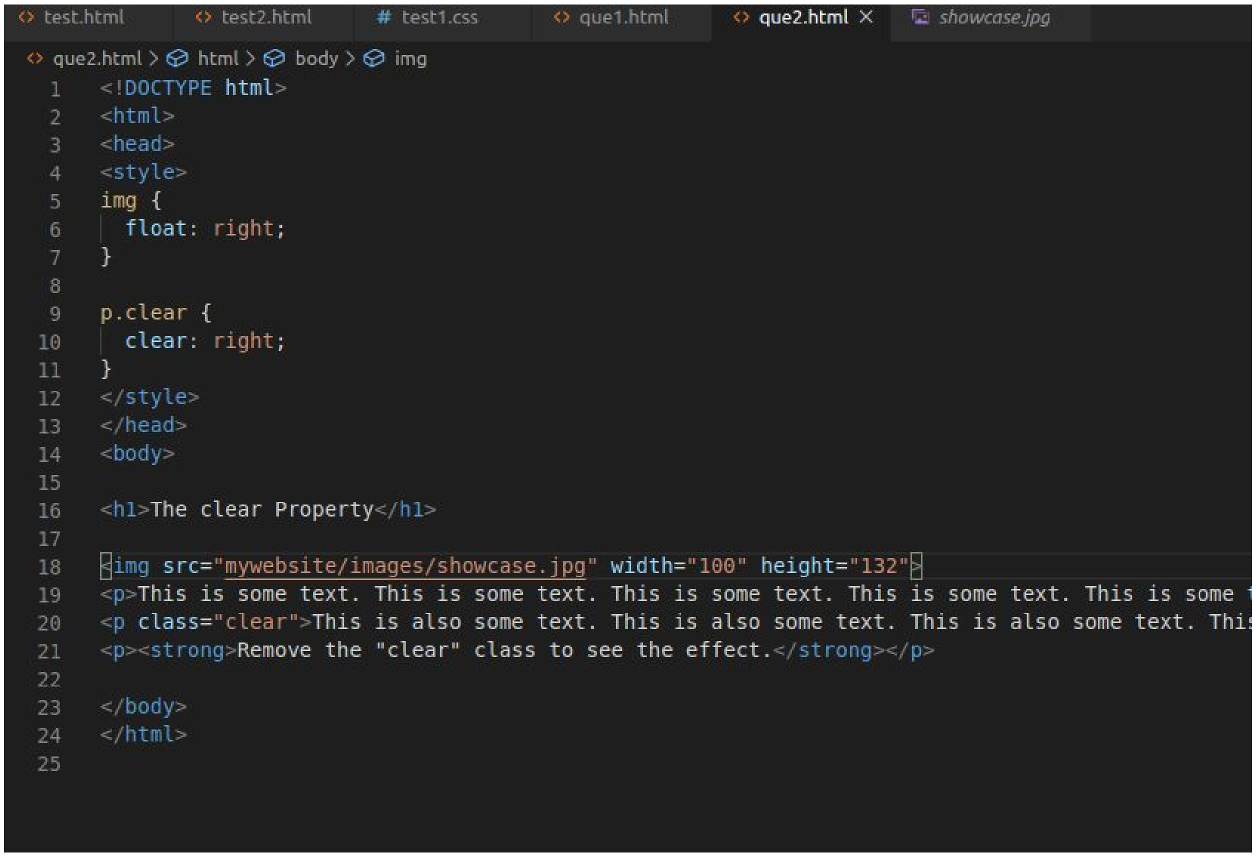
In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

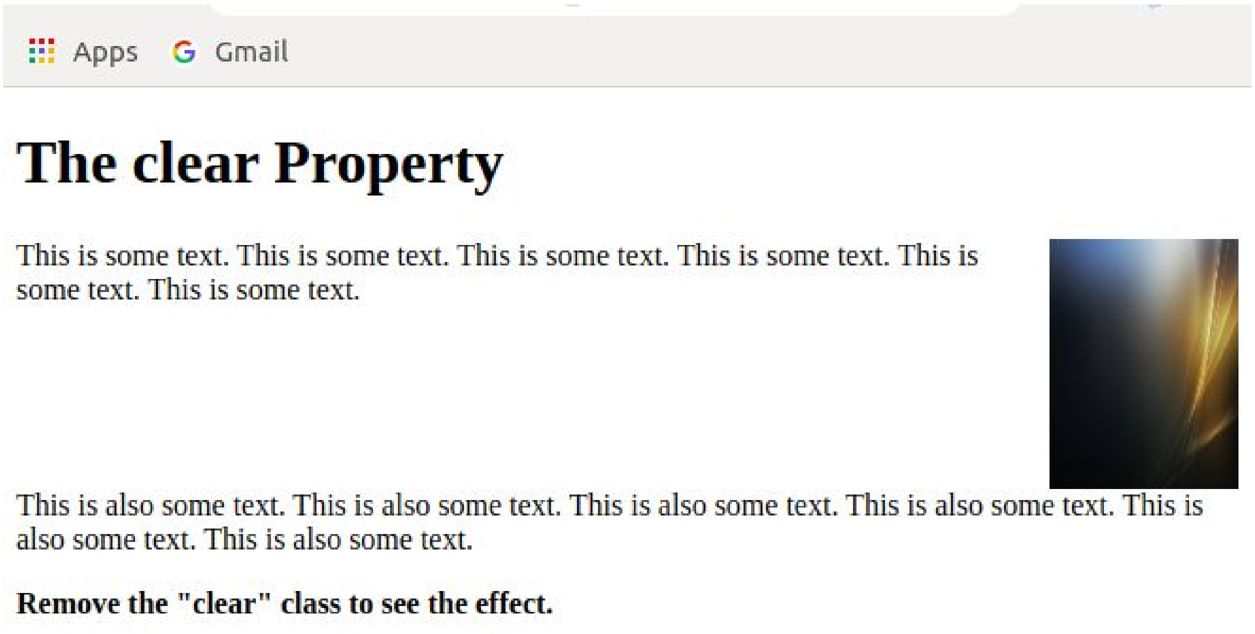
The clear property specifies what elements can float beside the cleared element and on which side.

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



```html
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}

p.clear {
  clear: right;
}
</style>
</head>
<body>

<h1>The clear Property</h1>

<img src="mywebsite/images/showcase.jpg" width="100" height="132">
<p>This is some text. This is some text. This is some text. This is some text. This is some t
<p class="clear">This is also some text. This is also some text. This is also some text. This
<p><strong>Remove the "clear" class to see the effect.</strong></p>

</body>
</html>
```



The clear Property

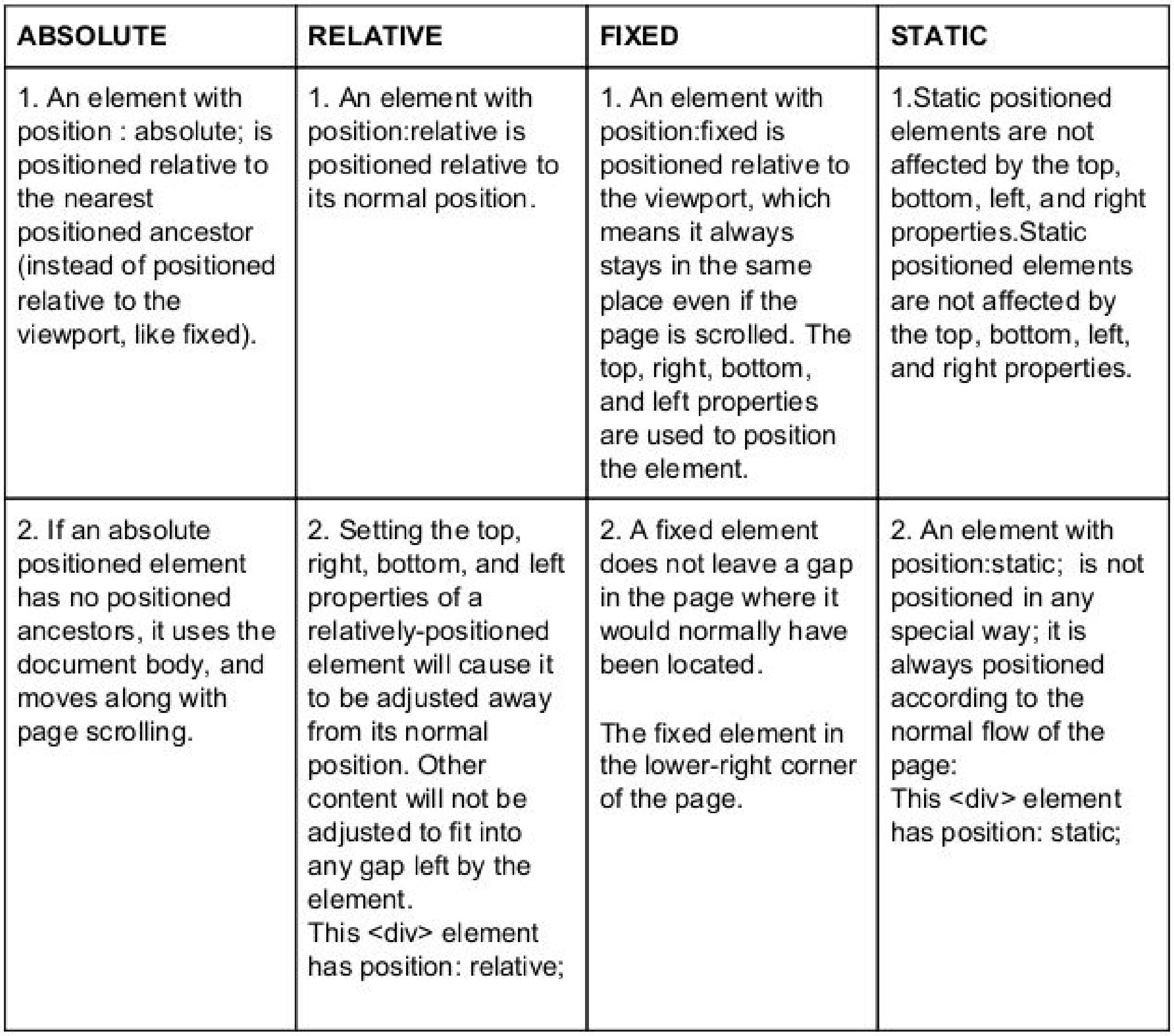
This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.

Q4. explain difference between absolute, relative,fixed and static.

Answer:

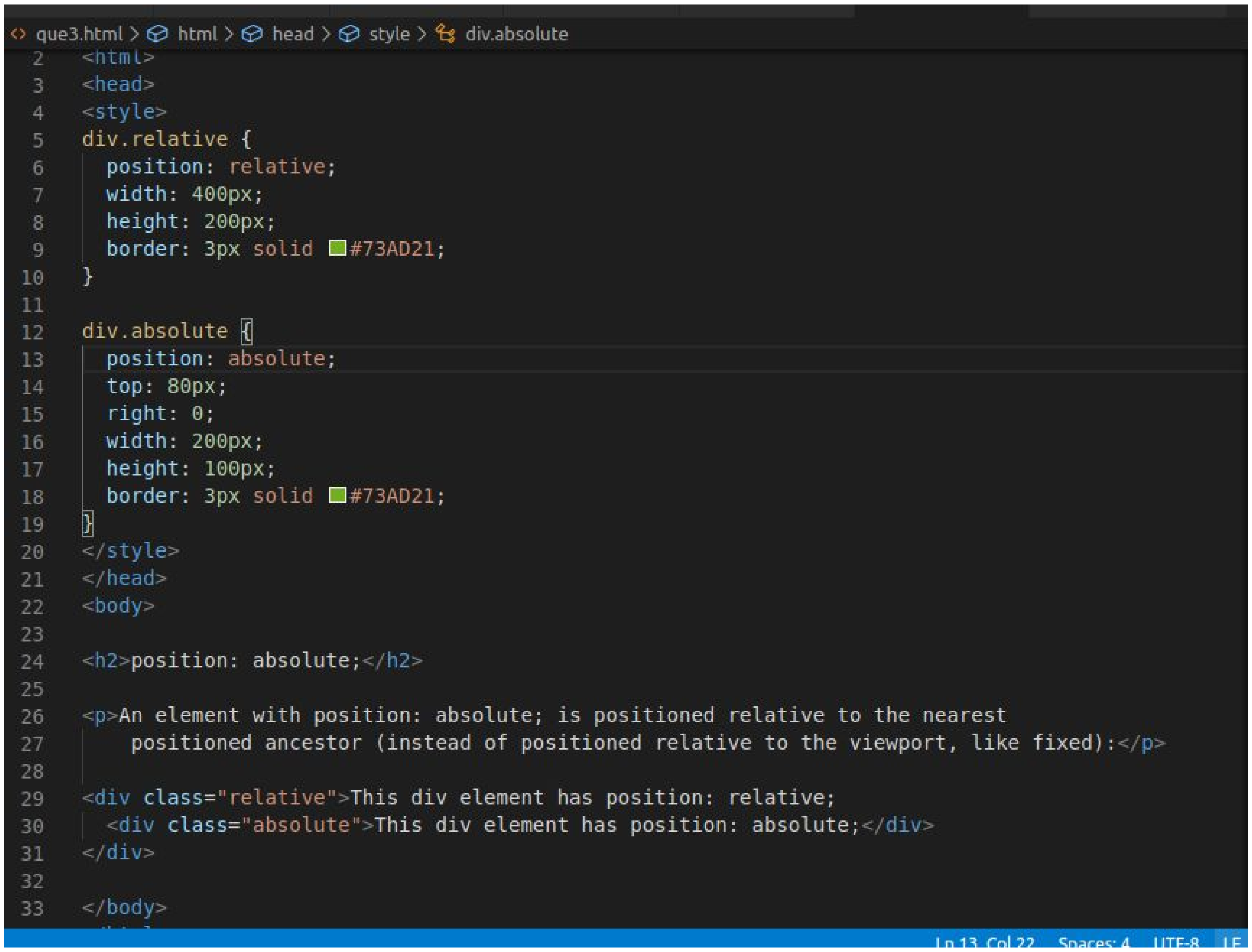


ABSOLUTE	RELATIVE	FIXED	STATIC
1. An element with position : absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).	1. An element with position:relative is positioned relative to its normal position.	1. An element with position:fixed 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.	1.Static positioned elements are not affected by the top, bottom, left, and right properties.Static positioned elements are not affected by the top, bottom, left, and right properties.
2. If an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.	2. 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. This <div> element has position: relative;	2. A fixed element does not leave a gap in the page where it would normally have been located. The fixed element in the lower-right corner of the page.	2. An element with position:static; is not positioned in any special way; it is always positioned according to the normal flow of the page: This <div> element has position: static;

position: absolute;

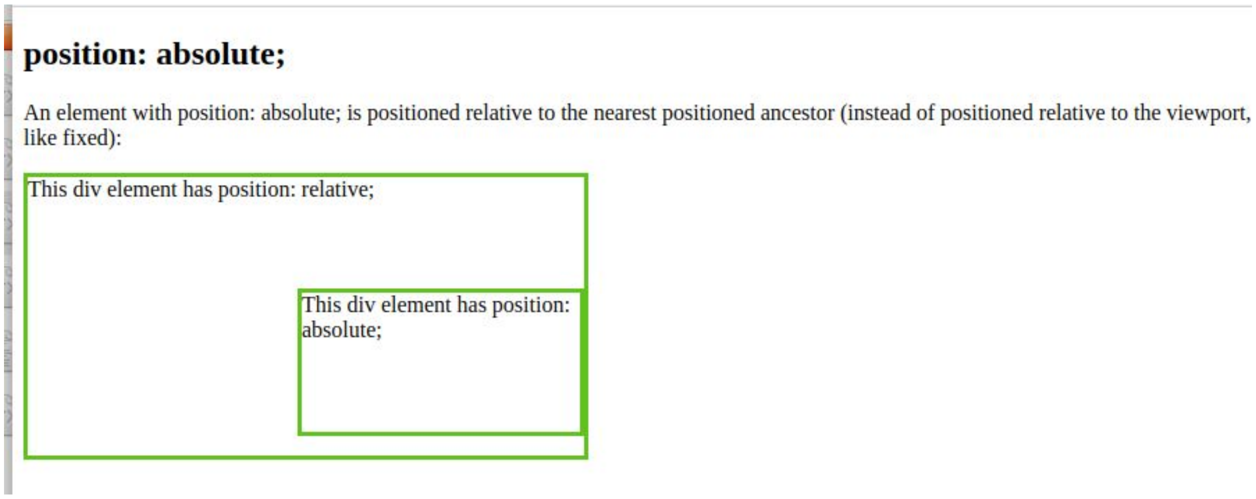
An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.


que3.html > html > head > style > div.absolute

```
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}

div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: absolute;</h2>

<p>An element with position: absolute; is positioned relative to the nearest
   positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>

<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
```

Ln 13, Col 22   Spaces: 4   UTF-8   LF


position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

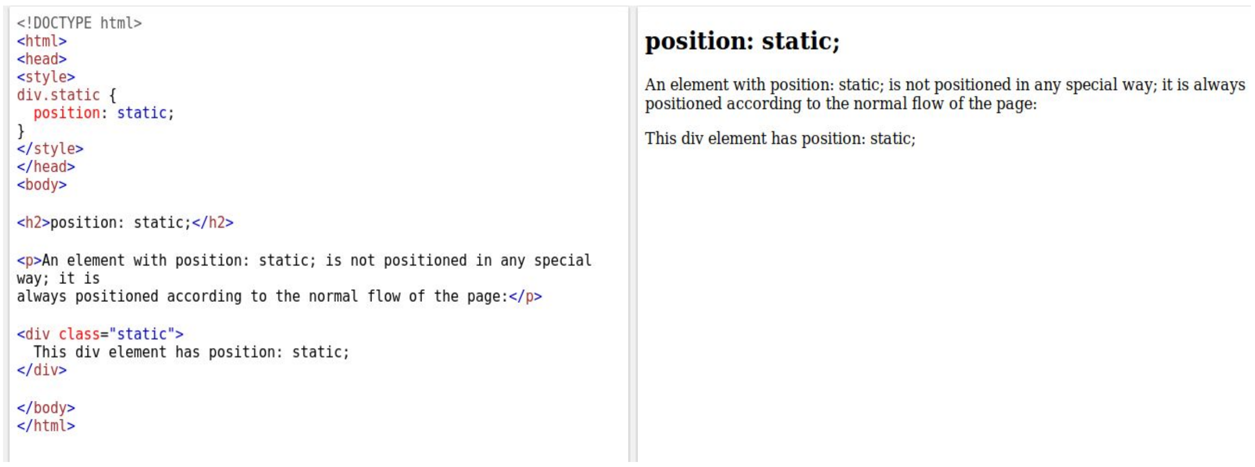
This div element has position: absolute;

position: 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:



```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
  position: static;
}
</style>
</head>
<body>

<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special
way; it is
always positioned according to the normal flow of the page:</p>

<div class="static">
  This div element has position: static;
</div>

</body>
</html>
```

position: static;

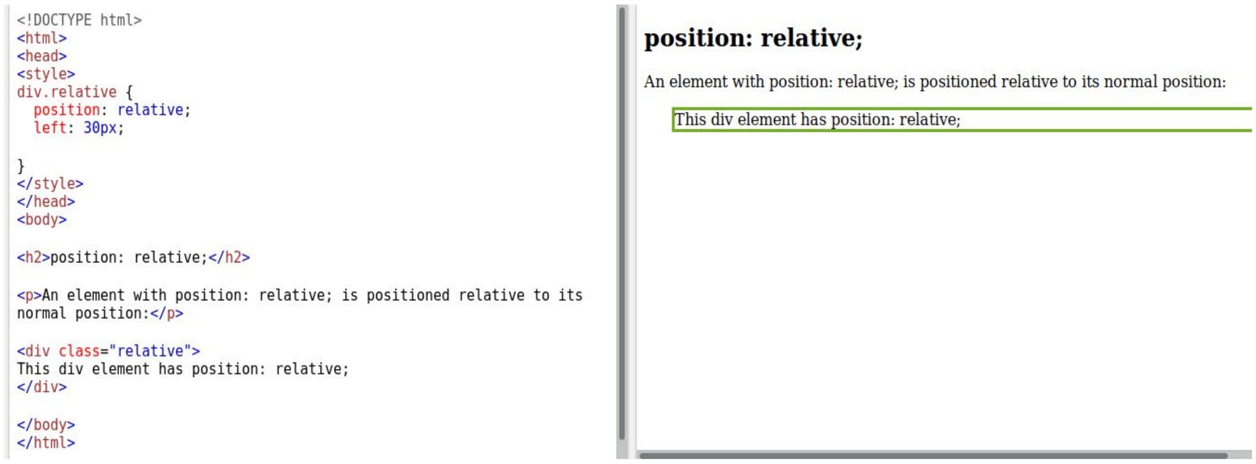
An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This div element has position: static;

position: 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.



```html
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  left: 30px;

}
</style>
</head>
<body>

<h2>position: relative;</h2>

<p>An element with position: relative; is positioned relative to its
normal position:</p>

<div class="relative">
This div element has position: relative;
</div>

</body>
</html>
```

position: relative;

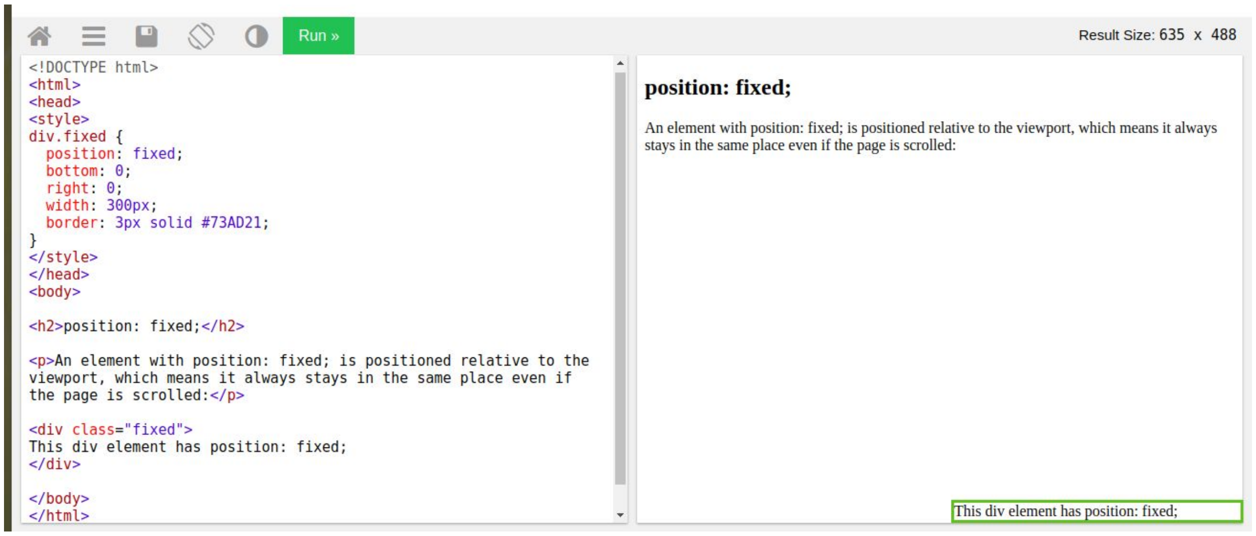
An element with position: relative; is positioned relative to its normal position:

This div element has position: relative;

position: 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.



```html
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: fixed;</h2>

<p>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:</p>

<div class="fixed">
This div element has position: fixed;
</div>

</body>
</html>
```

position: fixed;

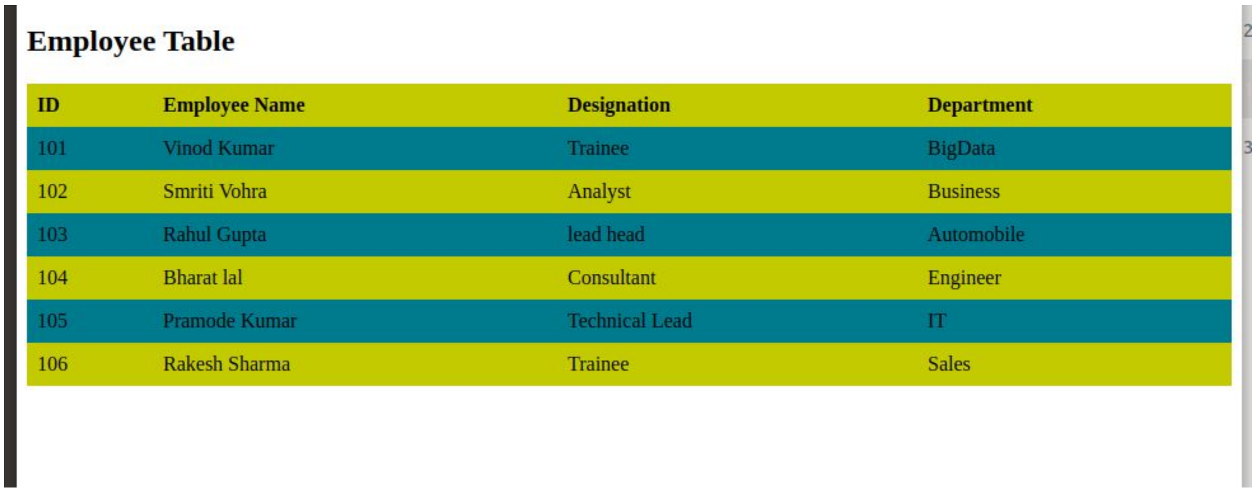
An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

This div element has position: fixed;

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.

Answer:

Code is available in five.html file which is in git rep



Employee Table

ID	Employee Name	Designation	Department
101	Vinod Kumar	Trainee	BigData
102	Smriti Vohra	Analyst	Business
103	Rahul Gupta	lead head	Automobile
104	Bharat lal	Consultant	Engineer
105	Pramode Kumar	Technical Lead	IT
106	Rakesh Sharma	Trainee	Sales

Q6. Why do we use meta tags?

Answer:

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.

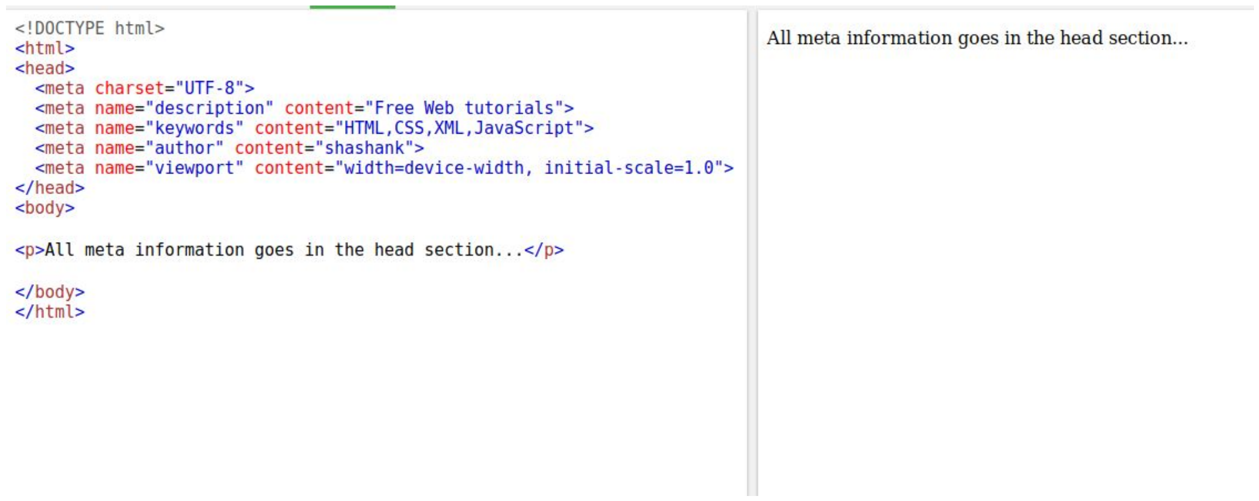
Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

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

Note: <meta> tags always go inside the <head> element.

Note: Metadata is always passed as name/value pairs.

Note: The content attribute MUST be defined if the name or the http-equiv attribute is defined. If none of these are defined, the content attribute CANNOT be defined.



```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="shashank">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<p>All meta information goes in the head section...</p>

</body>
</html>
```

All meta information goes in the head section...

Q7. Explain box model.

Answer:

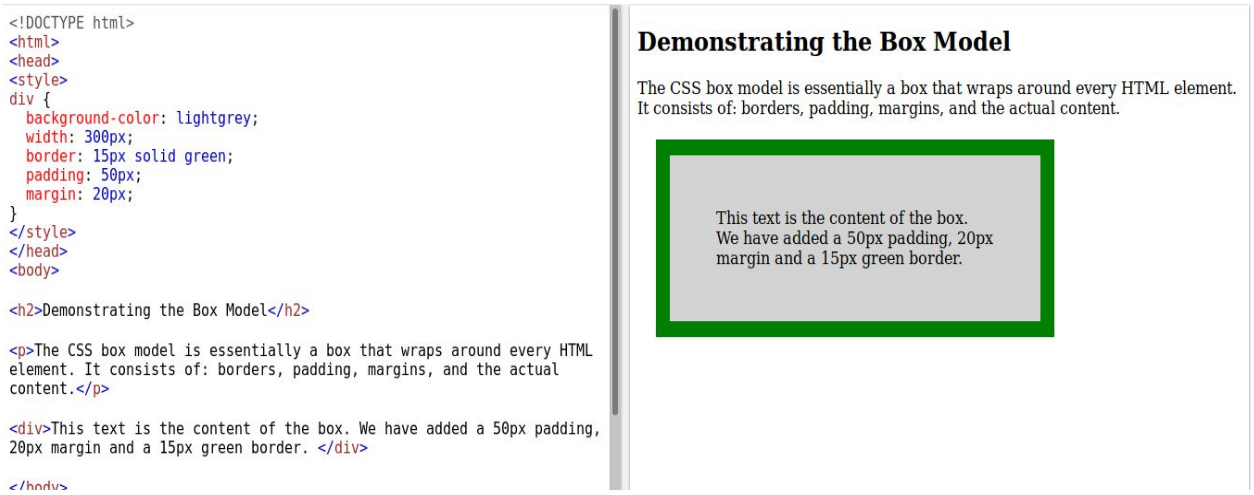
All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

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



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

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

<div>This text is the content of the box. We have added a 50px padding,
20px margin and a 15px green border. </div>

</body>
```

Demonstrating the Box Model

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

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.

Q8. What are the different types of CSS Selectors?

Answer:

A CSS selector is the part of a CSS rule set that actually selects the content you want to style.

Let’s look at all the different kinds of selectors available, with a brief description of each.

<u>Universal Selector</u>

The universal selector works like a wildcard character, selecting all elements on a page. Every

HTML page is built on content placed within HTML tags. Each set of tags represents an element on the page.

<u>Element Type Selector</u>

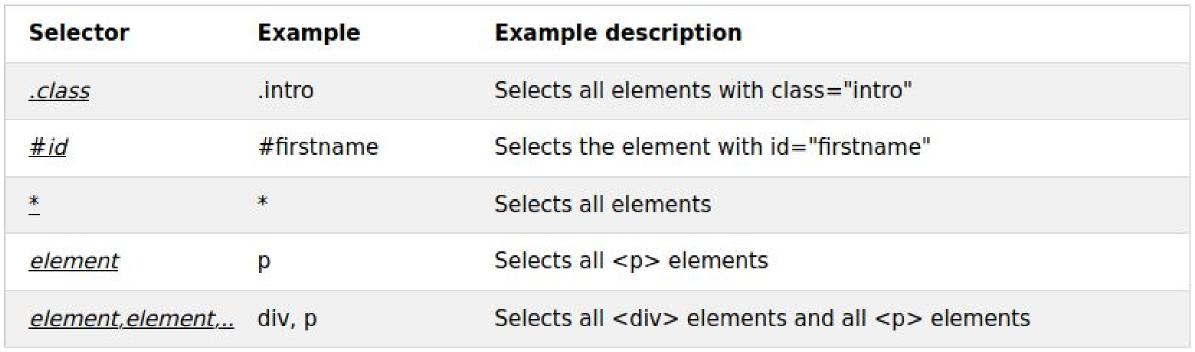
Also referred to simply as a “type selector,” this selector must match one or more HTML elements of the same name. Thus, a selector of nav would match all HTML nav elements, and a selector of <ul> would match all HTML unordered lists, or <ul> elements.

<u>ID Selector</u>

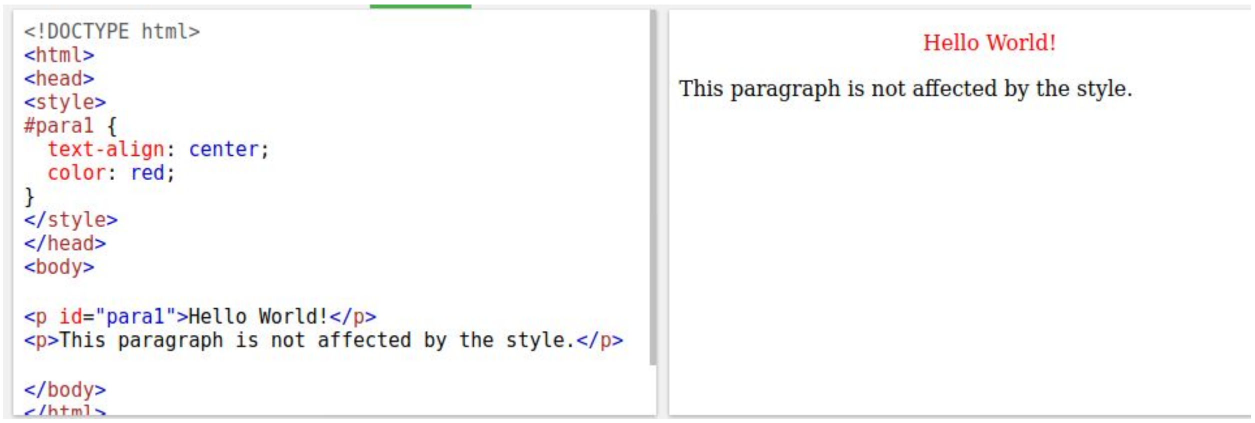
An ID selector is declared using a hash, or pound symbol (#) preceding a string of characters. The string of characters is defined by the developer. This selector matches any HTML element that has an ID attribute with the same value as that of the selector, but minus the hash symbol.

<u>Class Selector</u>

The class selector is the most useful of all CSS selectors. It’s declared with a dot preceding a string of one or more characters. Just as is the case with an ID selector, this string of characters is defined by the developer. The class selector also matches all elements on the page that have their class attribute set to the same value as the class, minus the dot.



Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element,..	div, p	Selects all <div> elements and all <p> elements



```html
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

```
Hello World!

This paragraph is not affected by the style.
```

Q9. Define Doctype.

Answer:

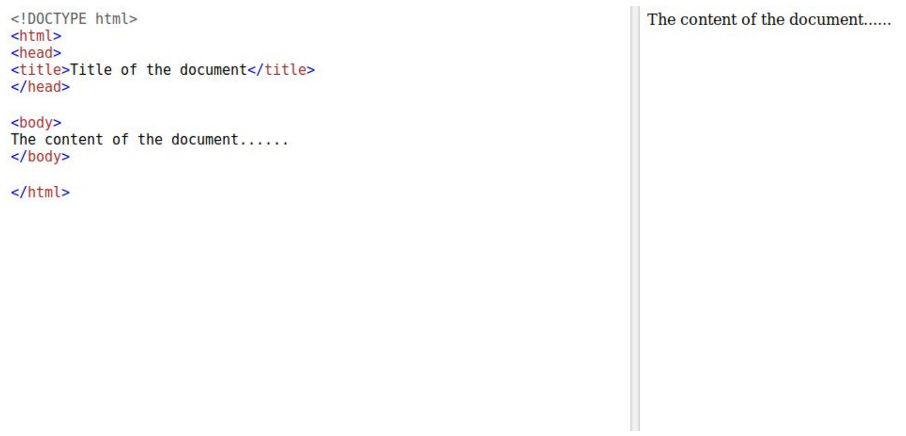
The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag.

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, 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, and therefore does not require a reference to a DTD.

NOTE: Always add the <!DOCTYPE> declaration to your HTML documents, so that the browser knows what type of document to expect.



```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document......
</body>

</html>
```

The content of the document......

Q10. Explain 5 HTML5 semantic tags.

Answer:

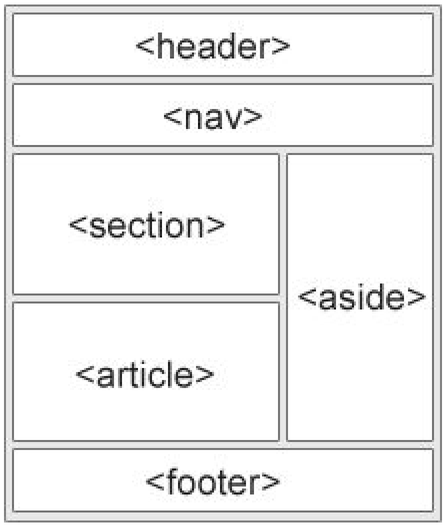
Semantic HTML elements clearly describe it's meaning in a human and machine readable way. Elements such as <header> , <footer> and <article> are all considered semantic because they accurately describe the purpose of the element and the type of content that is inside them.

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

HTML5 offers new semantic elements to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>

- <summary>
- <time>



1. HTML5 <section> Element

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.

2.HTML5 <article> Element

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

3. HTML5 <header> Element

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.

4. HTML5 <footer> Element

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.

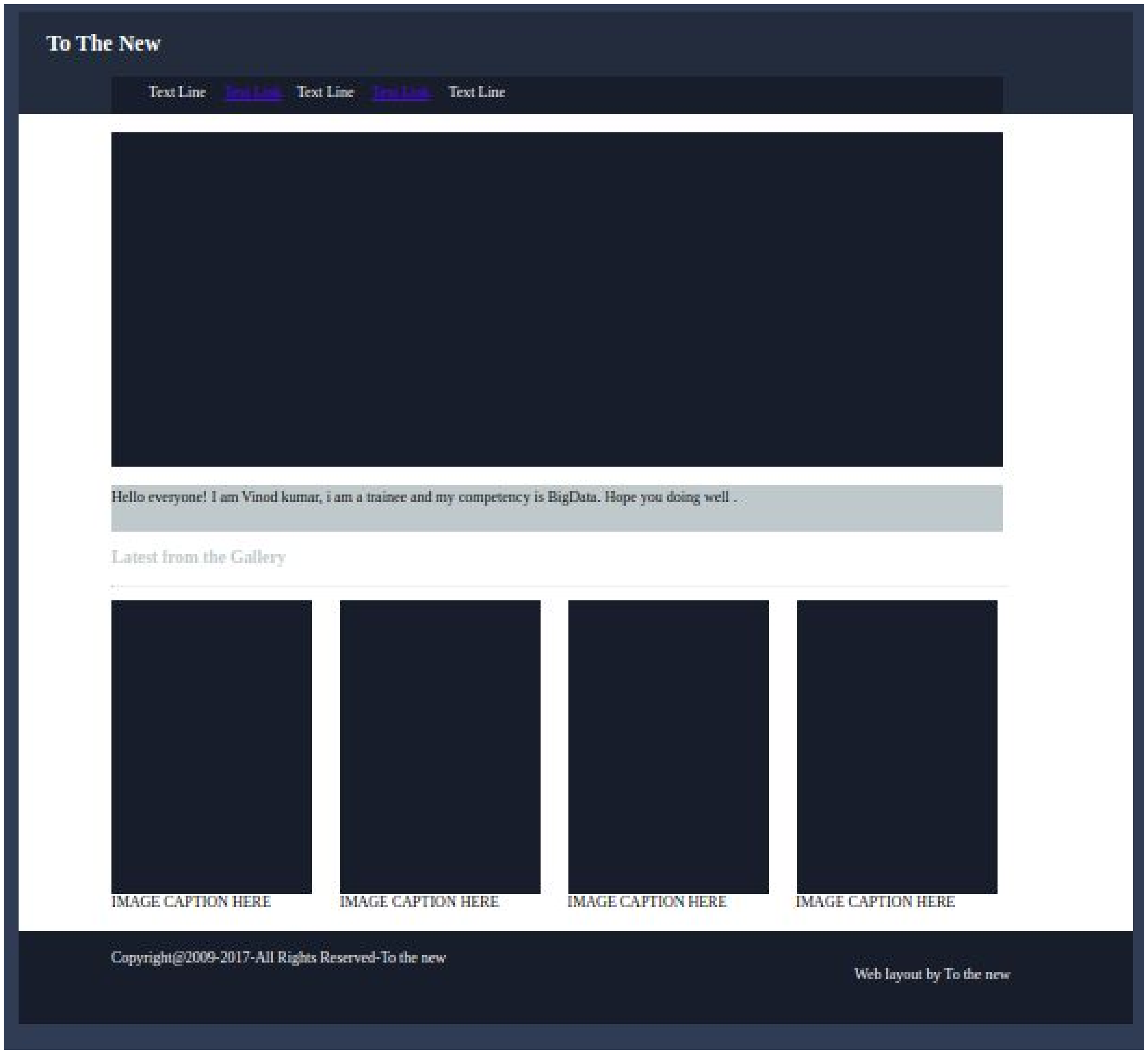
5. HTML5 <nav> Element

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

NOTICE: Not all links of a document should be inside a <nav> element. The <nav> element is intended only for major blocks of navigation links.

Q11. Create HTML for web-page.jpg (check resources, highest weightage for answers)

Answer: code availabe on git



To The New

Text Line Text Link Text Line Text Link Text Line

Hello everyone! I am Vinod kumar, i am a trainee and my competency is BigData. Hope you doing well .

Latest from the Gallery

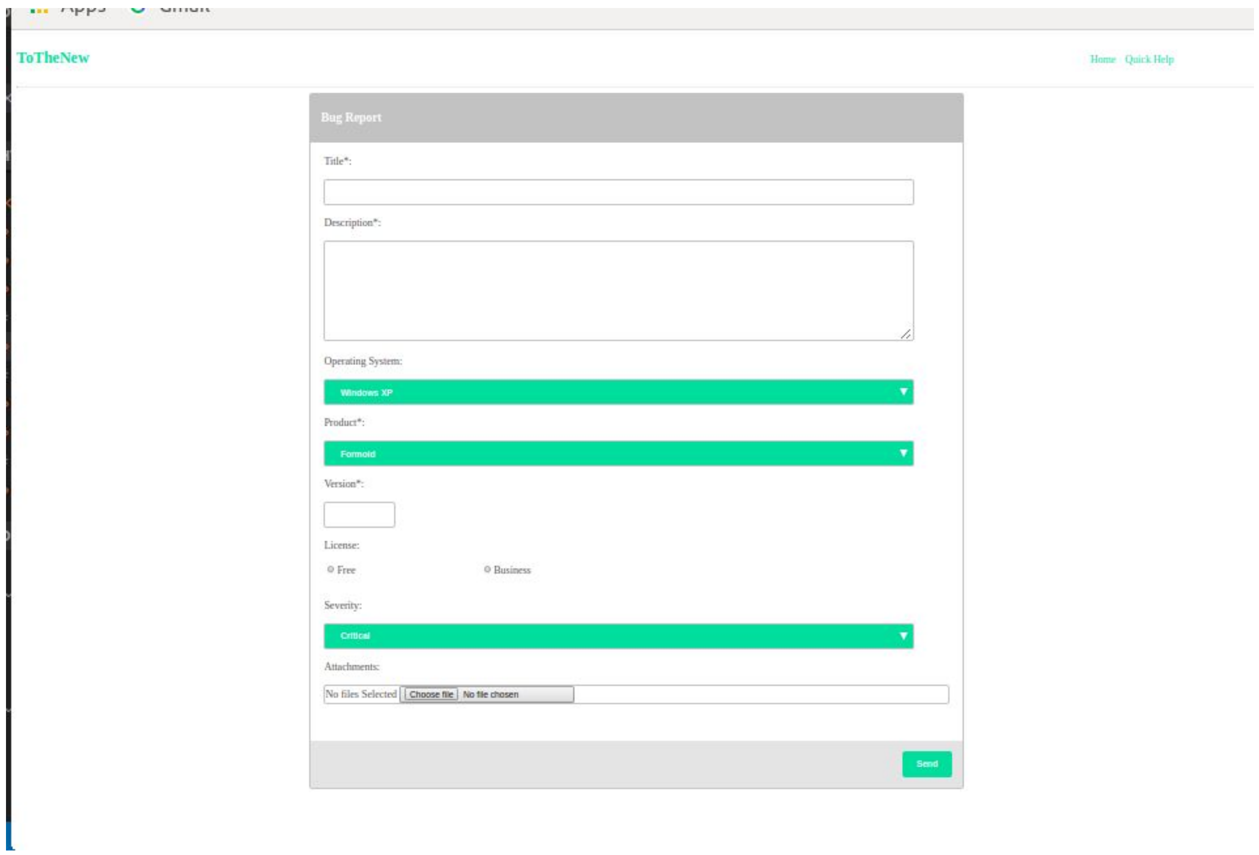
IMAGE CAPTION HERE IMAGE CAPTION HERE IMAGE CAPTION HERE IMAGE CAPTION HERE

Copyright@2009-2017-All Rights Reserved-To the new

Web layout by To the new

Q12. Create HTML for form.png (check resources, highest weightage for answers)

Answer: code availabe on git



ToTheNew

Home Quick Help

Bug Report

Title*:

Description*:

Operating System: Windows XP

Product*: Formoid

Version*:

License: Free Business

Severity: Critical

Attachments: No files Selected Choose file No file chosen

Send