**Sol 1.**

HTML is made up of various elements that act as the building blocks of web pages. For the purpose of styling, elements are divided into two categories: *block-level* elements and *inline* elements.

In summary, 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.

Other examples of inline elements are:

--anchor <a> tag.

--image <img> 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.

**Sol 2.**

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.

display:none removes the element from the normal flow of the page, allowing other elements to fill in.

visibility:hidden leaves the element in the normal flow of the page such that is still occupies space.

**Sol 3.**

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.

**Sol 4.**

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.

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.

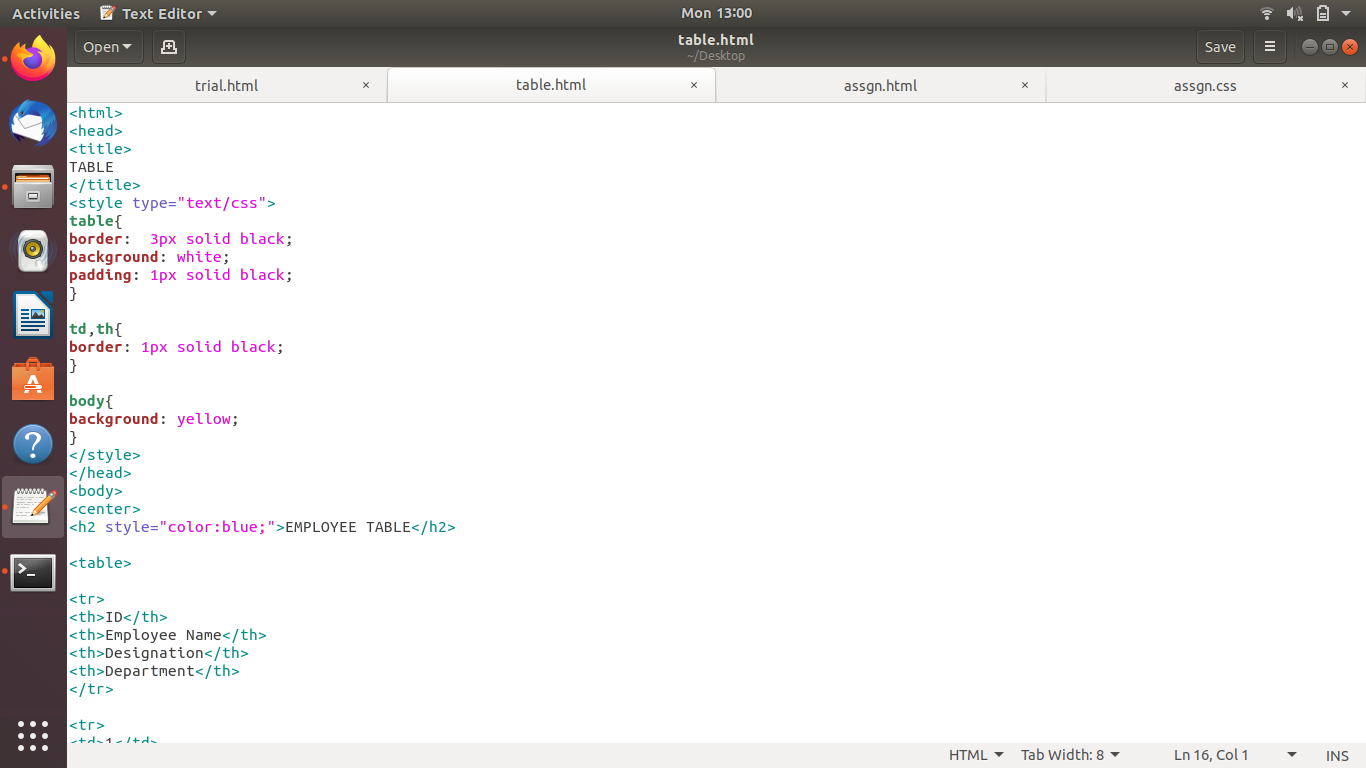
Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

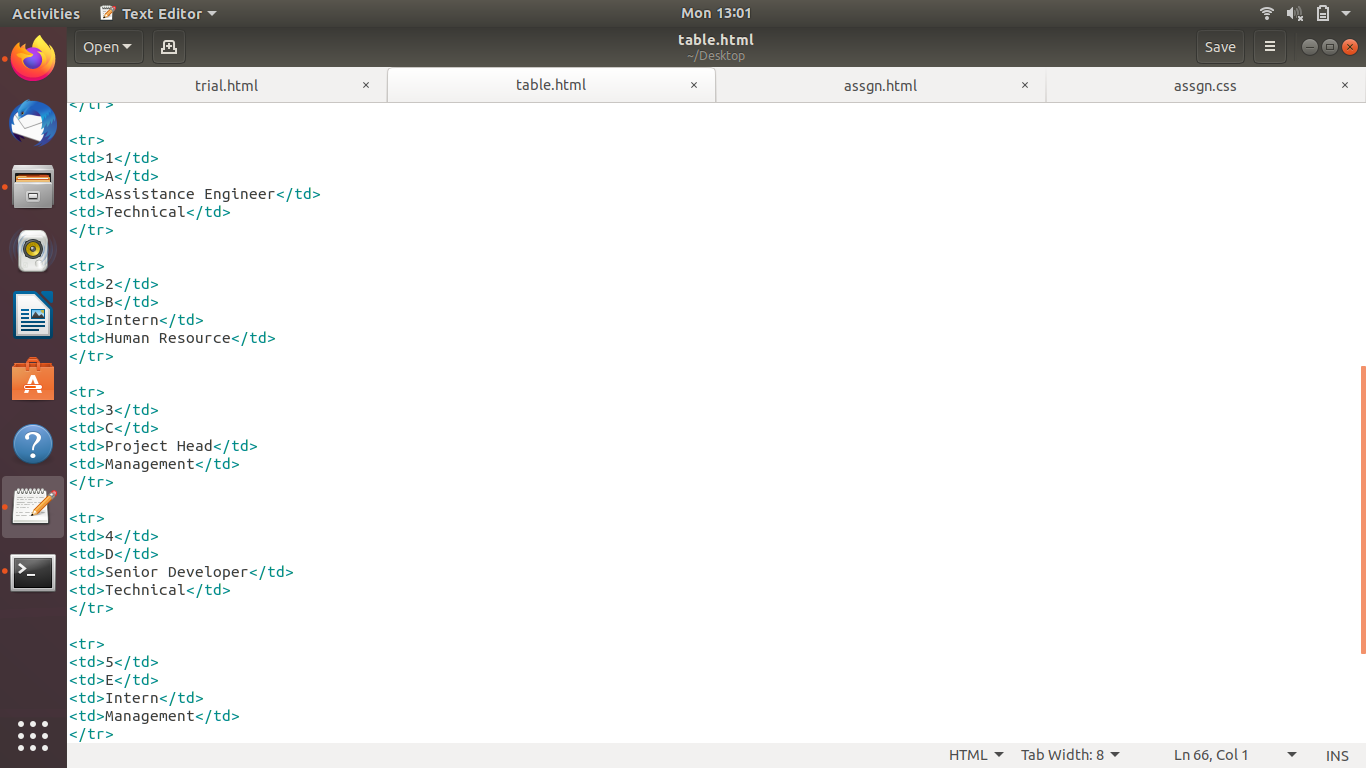
An element with **position: fixed**; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

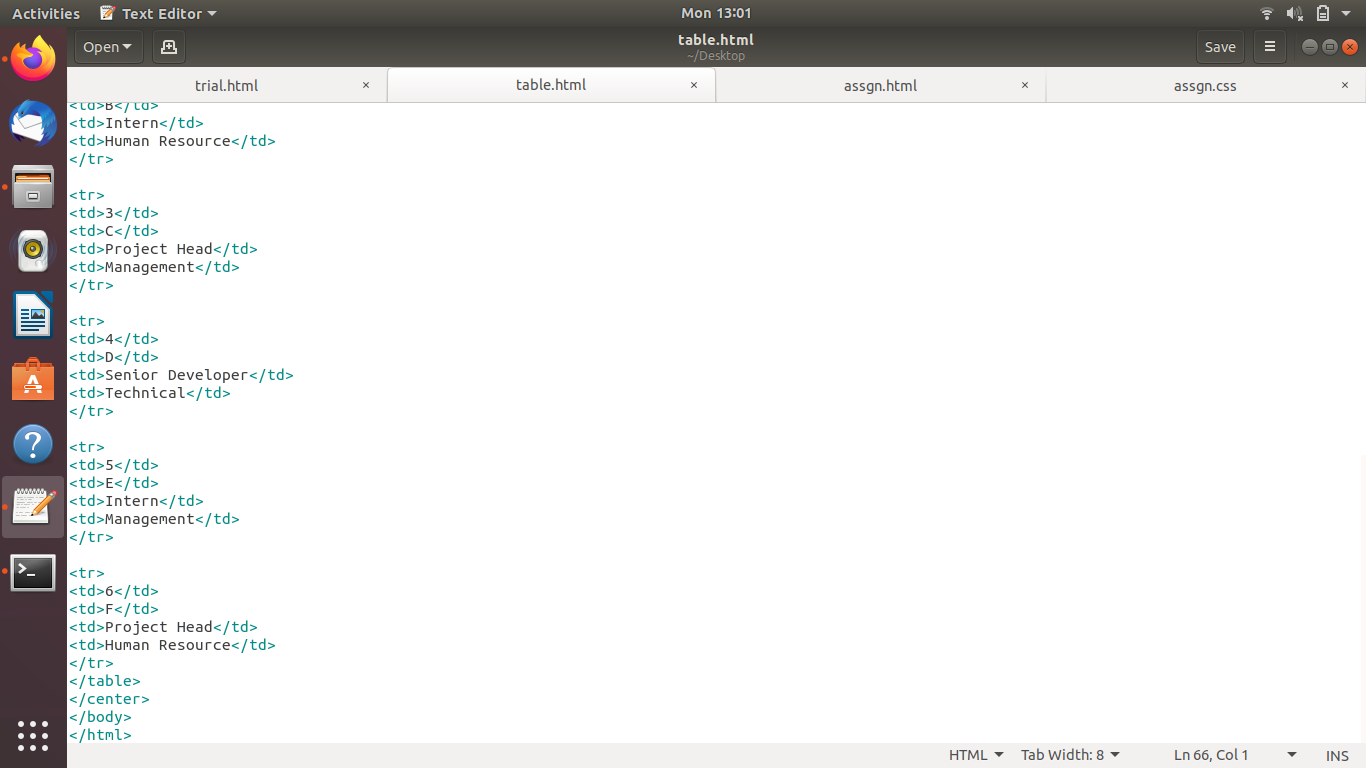
An element with **position: absolute**; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

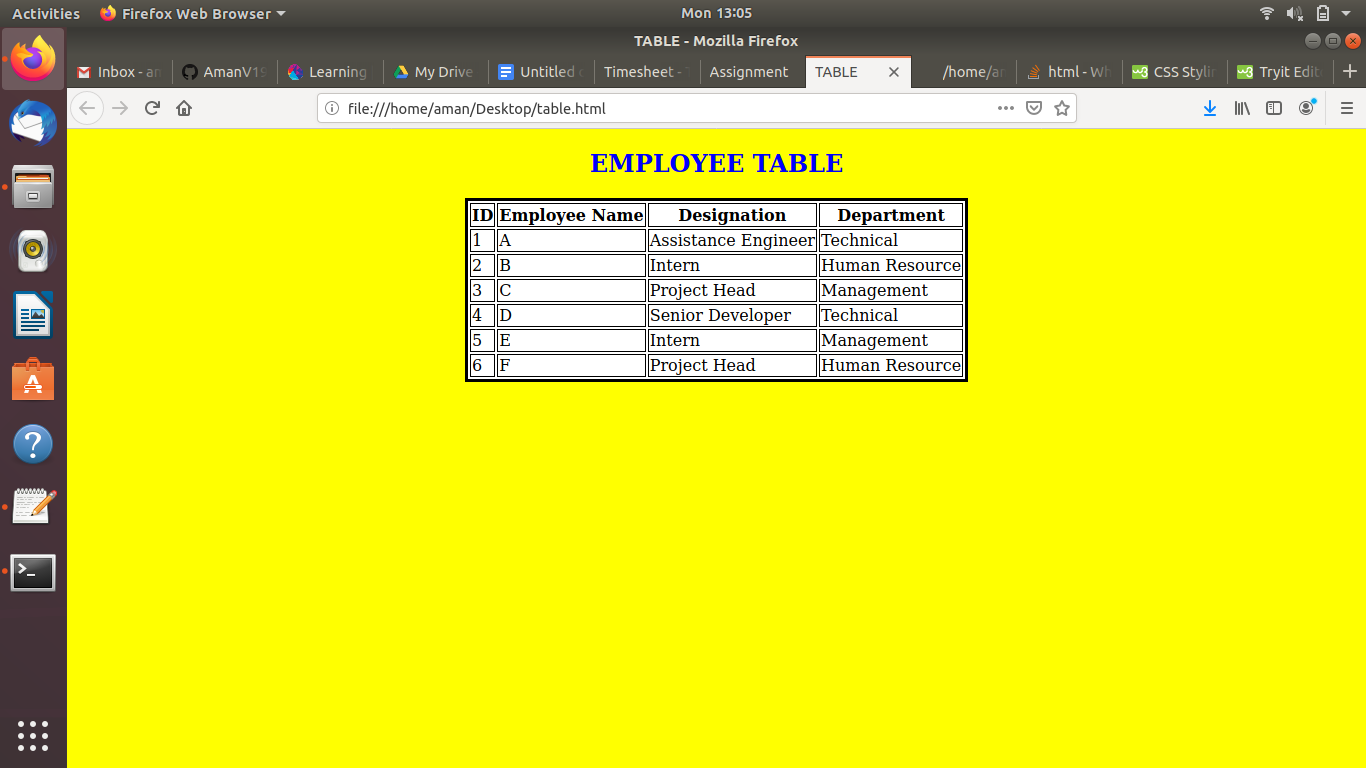
An element with **position: sticky**; is positioned based on the user's scroll position.

**Sol 5.**

****

****

****

****

**Sol 6.**

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.

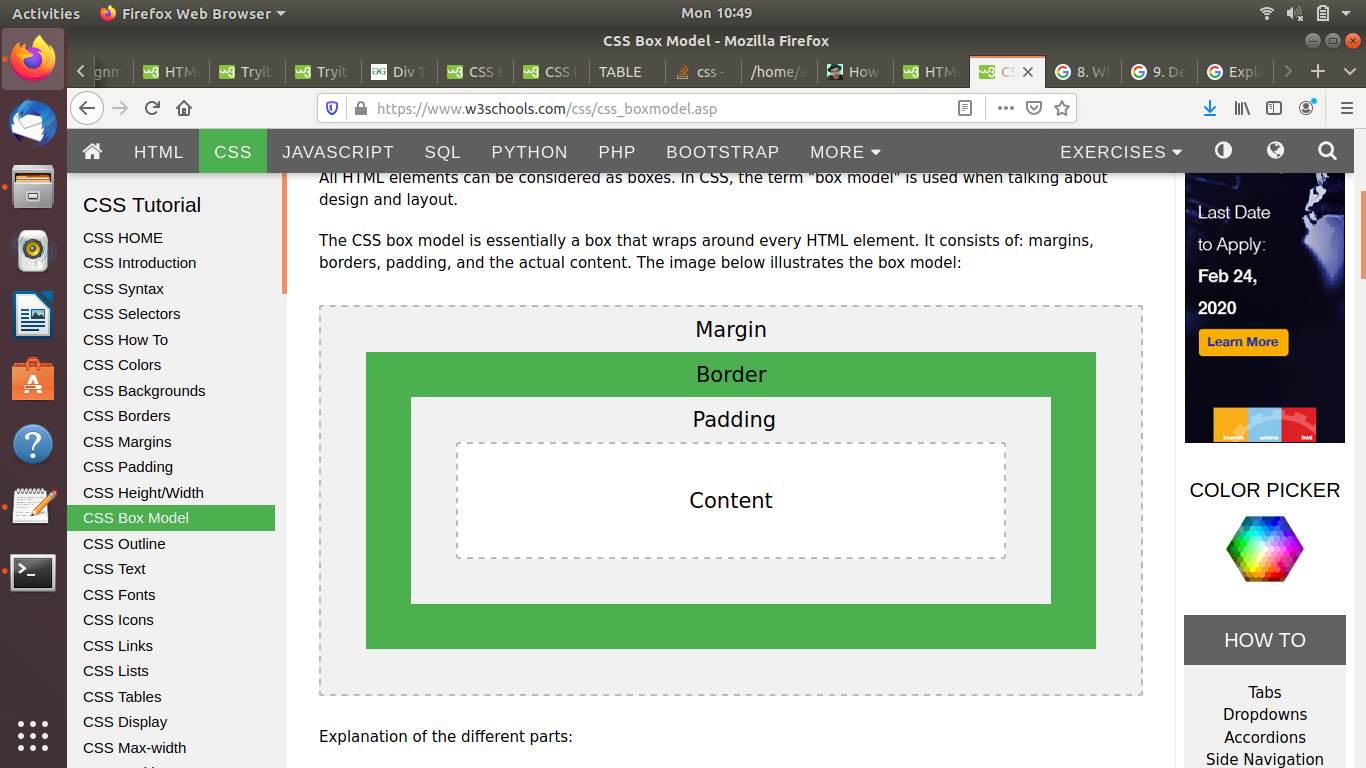
The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

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.

**Sol 7.**

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.

**Sol 8.**

## 1. Universal Selector

Selects all child elements under the parent element. Here style is applied to every element under the parent element. Its weight is more and to be used with care.

div \* {

font-size:14px;

}

## 2. Class Selector

Selects specified CSS class applied elements on the page. CSS class selector name starts with “.” followed by name.

.sec-important {

font-weight:bold;

}

## 3. ID Selector

Selects element which has a specified ID name. CSS ID selector name starts with “#” followed by name.

Note: ID name to be unique in a web page.

#p1 {

border:groove;

}

## 4. Element Selector

Selects elements based on element type.

div {

border:1px solid red;

}

## 5. Descendant Selector

Selects all specified descendant child elements under the parent element.

div p {

text-decoration:underline;

}

## 6. Child Selector

Selects all specified immediate child elements under the parent element.

div > p {

color:red;

}

## 

## 

## 7. Adjacent Sibling Selector

Selects specified elements which are immediate to an adjacent element.

div + p {

font-size:25px;

}

## 8. General Sibling Selector

Selects all specified elements which are siblings to an adjacent element.

div ~ p {

border:1px solid green;

}

**Sol 9.**

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.

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.

**Sol 10.**

The semantic elements added in HTML5 are:

* <article>
* <aside>
* <details>
* <figcaption>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>

Elements such as <header>, <nav>, <section>, <article>, <aside>, and <footer> act more or less like <div> elements. They group other elements together into page sections. However where a <div> tag could contain any type of information, it is easy to identify what sort of information would go in a semantic <header> region.

(A) The **<section>** and **<article>** elements are conceptually similar and interchangeable. To decide which of these you should choose, take note of the following:

1. An article is intended to be independently distributable or reusable.
2. A section is a thematic grouping of content.

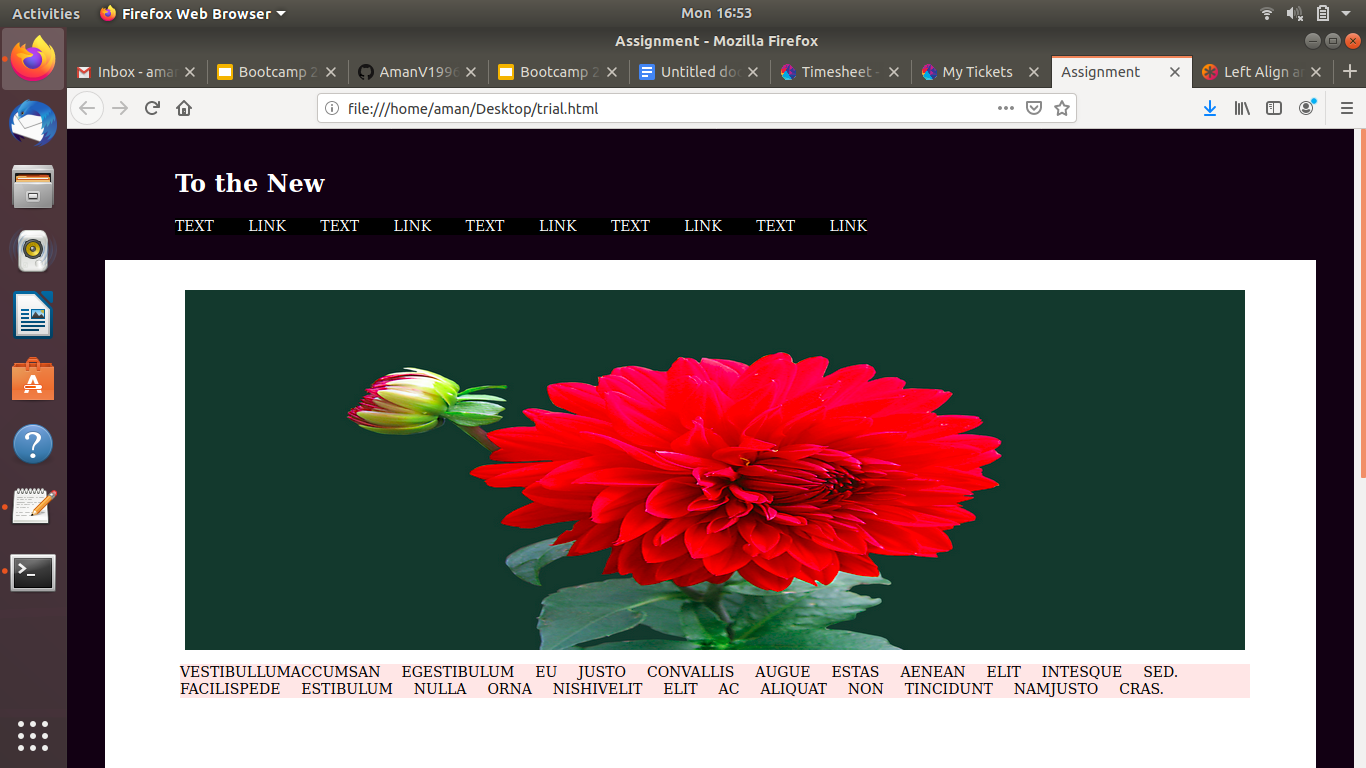
(B) The **<header>** element is generally found at the top of a document, a section, or an article and usually contains the main heading and some navigation and search tools.

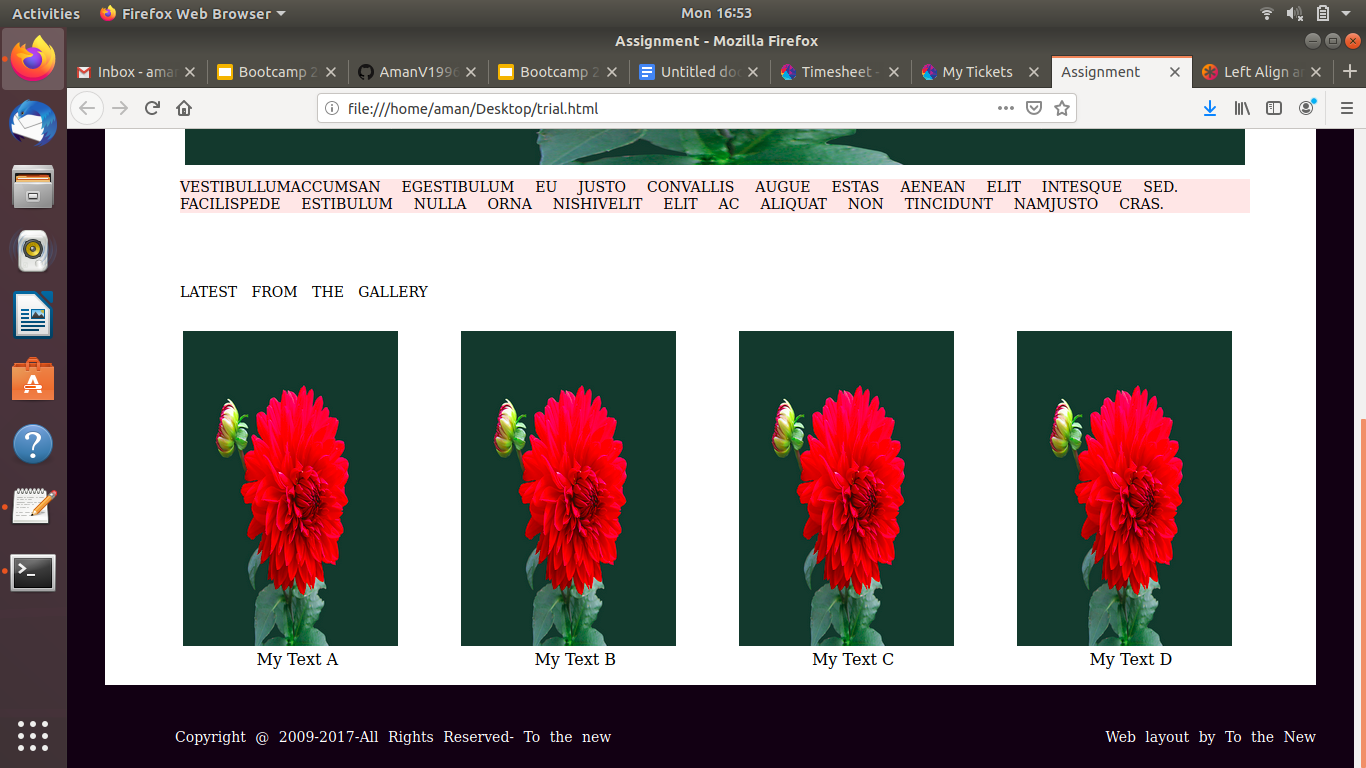
(C) The **<aside>** element is intended for content that is not part of the flow of the text in which it appears, however still related in some way. This of <aside> as a sidebar to your main content.

(D) A **<footer>** is generally found at the bottom of a document, a section, or an article. Just like the <header> the content is generally meta information, such as author details, legal information, and/or links to related information.

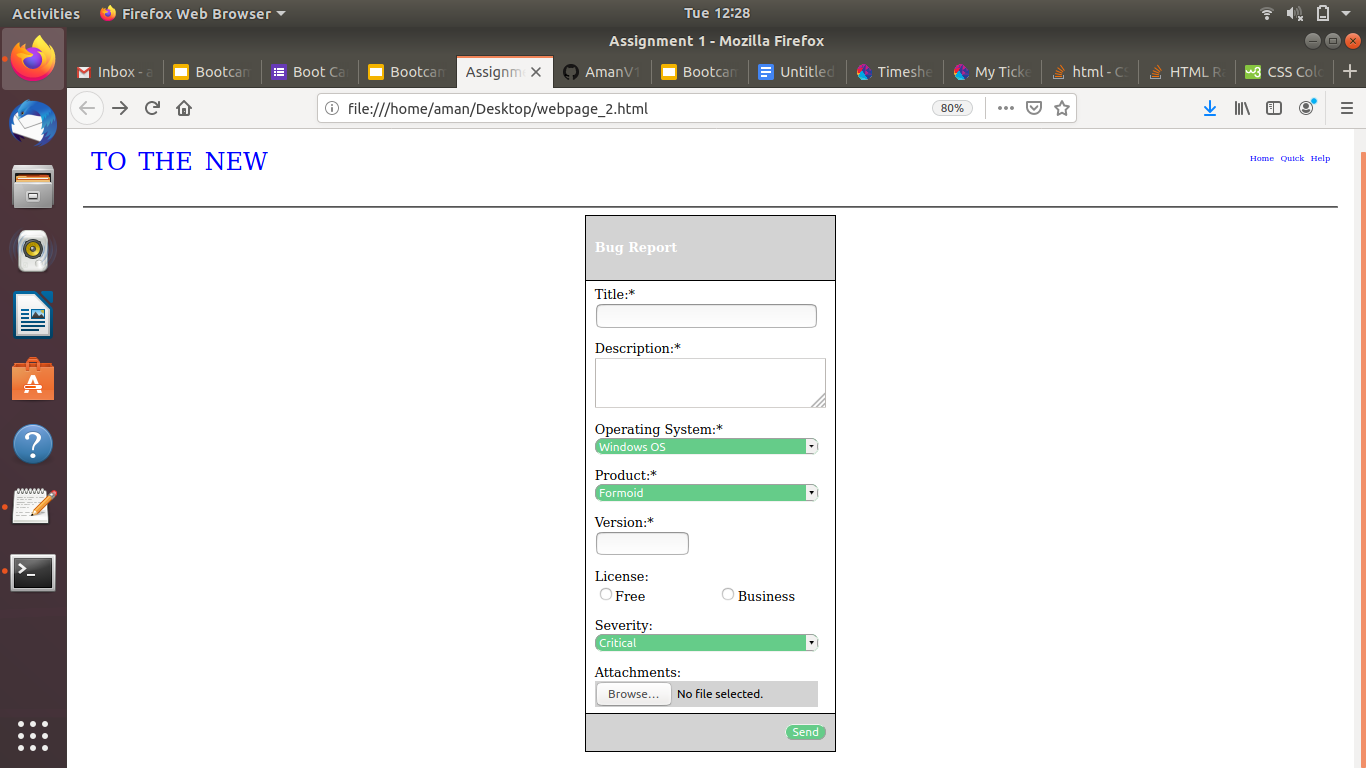
(E) The **<small>** element often appears within a <footer> or <aside> element which would usually contain copyright information or legal disclaimers, and other such fine print. However, this is not intended to make the text smaller. It is just describing its content, not prescribing a presentation.

**Sol 11.**

****

****

**Sol 12.**

****