Introduction To HTML/CSS
Exercise
Name: Mahesh Inder

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

A. 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 element. It is usually used within other HTML elements.
e.g. <img>, , <code>, <b> etc.

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.
e.g. <h1>, , <p>, <ul> etc.

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

A. ***display:none*** means that the element will not appear on the page at all. There will be no space allocated for it between the other elements.

visibility:hidden means the element is not visible, but space is allocated for it on the page. The element is rendered, it just isn't seen on the page.

3. Explain the clear and float properties.

A. ***Clear:*** 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.

Float: The float property is used for positioning and formatting content.
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.

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

A. ***position: absolute:*** An element with *position: absolute* is positioned relative to the nearest positioned ancestor. However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

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.

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.

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.

6. Why do we use meta tags?

A. The meta tag provides metadata about the HTML document. Metadata is not displayed on the page. 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.

7. Explain box model.

A. 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*.

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.

8. What are the different types of CSS Selectors?

A. There are following types of CSS selectors:

- ***Universal selecto***r: The universal selector works like a wildcard character, selecting all elements on a page.
  e.g. **{ color:white; }*
- ***Element selector:*** It is 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 elements.
  e.g. ul{ }
- ***Id Selector:*** 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.
- ***Class Selector:*** 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.
  e.g. .className{ padding:10px; }

9. **Define Doctype.**

A. Doctype is a declaration which is an instruction to the web browser about what version of HTML the page is written in. The declaration must be the very first thing in your HTML document, before the tag. In HTML 4.01, the 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.

10. Explain 5 HTML5 semantic tags.

A. A semantic element clearly describes its meaning to both the browser and the developer.
   Examples of non-semantic elements: , (tells nothing about its content.)
   Examples of semantic elements: <form>, <article> , <table> (clearly defines its content.)

1. **<section>:** The element defines a section in a document or it is grouping of content, typically with a heading.
2. **<article>**: The 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 element can be used: Forum post,Blog post and Newspaper article
3. **<header>:** The element specifies a header for a document or section.The element should be used as a container for introductory content. You can have several elements in one document.
4. **<footer>:** The element specifies a footer for a document or section.A 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.
5. **<nav>:** The element defines a set of navigation links.

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.

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

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

Answers to question 5, 11 and 12 are in different directories, i.e. table, form and webpage.