Practical No 1

CSS Syntax:

CSS follows a specific format that consists of three main components:

1. **Selector:** Specifies which HTML element(s) to style.
2. **Property:** Defines the aspect of the element to modify (e.g., color, font-size).
3. **Value:** Assigns a specific value to the property.

General Syntax Structure:

```
selector {
    property: value;
}
```

```
e.g.
p {
    background-color: blue;
    color: white;
}
```

Practical No 2

Types of CSS:

1. **Inline CSS:** Applied directly within an HTML tag using the `style` attribute. It affects only that specific element.
2. **Internal CSS:** Defined within a `<style>` tag inside the `<head>` section of the HTML document. It affects elements within the same page.
3. **External CSS:** Written in a separate `.css` file and linked to the HTML file using a `<link>` tag. This method is used for maintaining styles across multiple pages.

CSS Selectors:

1. **Element Selector:** Targets all elements of a specific type (e.g., all `<h1>` tags).
2. **Class Selector (`.`):** Styles elements with a specific class name, allowing multiple elements to share the same style.
3. **ID Selector (#):** Targets a unique element with a specific ID. IDs should be unique within a page.
4. **Universal Selector (`*`):** Applies styles to all elements on a page.
5. **Grouping Selector (`,`):** Styles multiple elements at once, reducing repetition.

Practical No 3

1. **`background-color`** – This property sets the background color of an element. It can be defined using color names (e.g., red), HEX codes (e.g., `#ff0000`), RGB values (e.g., `rgb(255, 0, 0)`), or HSL values.
2. **`background-image`** – It allows you to set an image as the background of an element. The image can be a URL of an external or local file. If not properly sized, it may repeat across the page by default.
3. **`background-size`** – Defines the size of the background image. It determines how the image fits within the element, allowing values like `cover` (scales to fit the entire element) or `contain` (scales to fit within the element while maintaining aspect ratio).

Practical No 4

1. **`font-family`** – Specifies the type of font used for text. You can use predefined system fonts or custom fonts. It is recommended to provide multiple font names as fallback options.
2. **`font-size`** – Controls the size of the text. It can be set in various units such as pixels (`px`), ems (`em`), percentages (`%`), or relative units like `rem`.
3. **`font-weight`** – Defines the thickness of the text. Common values include `normal`, `bold`, `lighter`, or numeric values like `100` to `900`, where `900` is the boldest.
4. **`text-align`** – This property controls the alignment of text within an element. Possible values include `left`, `right`, `center`, and `justify`, where `justify` ensures even spacing between words.

Practical No 5

1. **`position`** – Specifies the method used for positioning an element on the page. It can be `static` (default, follows normal document flow), `relative` (positioned relative to itself), `absolute` (positioned relative to the nearest positioned ancestor), `fixed` (fixed relative to the viewport), or `sticky` (switches between relative and fixed based on scrolling).
2. **`top`** – Determines the distance of an element from the top edge of its containing element. It works when the `position` is set to `relative`, `absolute`, or `fixed`.
3. **`left`** – Similar to `top`, this property defines the distance of an element from the left edge of its container. It also works with `relative`, `absolute`, and `fixed` positioning.
4. **`right`** – Specifies the distance from the right edge of the containing element. It moves the element leftward if positioned absolutely or relatively.
5. **`bottom`** – Defines the distance from the bottom edge of the container. When applied, it pushes the element upward in an absolute or relative position.