Yahan CSS ki notes ka roman English mein version diya gaya hai:

---

1. CSS Basics

CSS Kya Hai?

- **Cascading Style Sheets (CSS)** ek style sheet language hai jo HTML ya XML document ki presentation ko define karti hai.
- Ye layout, colors, fonts, aur web pages ke overall appearance ko control karta hai.

CSS Syntax

- **Selector:** HTML element ko target karta hai jisme style apply karna hai.
- **Property:** Wo style feature jo aap change karna chahte hain.
- **Value:** Property ko diya gaya value.

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

Example:

```css
h1 { color: blue; font-size: 2em; }
```

CSS Ki Types

1. **Inline CSS:** Directly HTML element ke andar `style` attribute ka use karke.

```html
<h1 style="color: blue;">Hello World</h1>
```

2. **Internal CSS:** `<style>` tag ke andar `<head>` section mein define hota hai.

```html
<style> h1 { color: blue; } </style>
```

3. **External CSS:** Ek external `.css` file se link kiya jata hai.

```html
<link rel="stylesheet" href="styles.css">
```

---

2. Selectors

Basic Selectors

- **Universal Selector:** `*`

```css
* { margin: 0; padding: 0; }
```

- **Type Selector:** Specific HTML element ko target karta hai.

```css
```

```
p { color: red; }
```

- **Class Selector:** Specific class wale elements ko target karta hai.

```css
.example { font-size: 20px; }
```

- **ID Selector:** Specific ID wale element ko target karta hai (page par unique hona chahiye).

```css
#header { background-color: #333; }
```

Combinators

- **Descendant Selector (space):** Nested elements ko target karta hai.

```css
div p { color: green; }
```

- **Child Selector (>):** Direct children ko target karta hai.

```css
div > p { color: blue; }
```

Pseudo-classes

- `:hover` : Element pe hover hone par style apply hota hai.

```
css
```

```
a:hover { color: red; }
```

Copy code

- `:first-child` : Parent ke first child ko target karta hai.

```
p:first-child { font-weight: bold; }
```

Copy code

Pseudo-elements

- `::before` : Element ke pehle content add karta hai.

```
p::before { content: "Note: "; font-weight: bold; }
```

Copy code

---

3. Box Model

- **Content Box:** Element ka asli content.
- **Padding:** Content ke aas-paas ka space.
- **Border:** Padding ke aas-paas (optional).
- **Margin:** Element ke bahar ka space jo doosre elements se alag karta hai.

Box Model Properties

```
```

Copy code

```
div { width: 300px; height: 200px; padding: 20px; border: 5px solid black; margin: 15px; }
```

---

4. Positioning

Position Property

- `static` : Default position (normal flow).
- `relative` : Apne normal position ke relative positioned hota hai.
- `absolute` : Nearest positioned ancestor ke relative positioned hota hai.
- `fixed` : Viewport ke relative positioned hota hai, scroll nahi hota.
- `sticky` : Ek point ke baad sticky ho jata hai jab page scroll hota hai.

css Copy code

```
div { position: relative; top: 20px; left: 50px; }
```

Z-index

- Stacking order ko control karta hai positioned elements ka.
- Z-index ka higher value ka element lower value wale elements ke upar hota hai.

css Copy code

```
div { position: absolute; z-index: 10; }
```

5. Flexbox

Flexbox Basics

- `display: flex;` Element ko flex container bana deta hai.
- Flex items (flex container ke direct children) ko rows ya columns mein arrange kiya ja sakta hai.

Key Flex Properties

- `justify-content` : Main axis par items ko align karta hai (default horizontal).
  - `flex-start` , `flex-end` , `center` , `space-between` , `space-around`
- `align-items` : Cross axis par items ko align karta hai (default vertical).
  - `flex-start` , `flex-end` , `center` , `baseline` , `stretch`
- `flex-direction` : Flex container ki direction ko specify karta hai.
  - `row` , `column` , `row-reverse` , `column-reverse`
- `flex-wrap` : Flex items ko multiple lines mein wrap karne ki permission deta hai.
  - `nowrap` , `wrap` , `wrap-reverse`

Example: Flexbox Container

css Copy code

```css
.container { display: flex; justify-content: space-between; align-items: center; }
```

6. Grid Layout

Grid Basics

- `display: grid;` Element ko grid container bana deta hai.
- Container ko rows aur columns mein divide kiya jata hai, complex layouts ke liye.

Key Grid Properties

- `grid-template-rows` / `grid-template-columns` : Rows/columns ki size ko define karta hai.
- `grid-gap` : Rows aur columns ke beech ka gap define karta hai.
- `grid-column` / `grid-row` : Item ko grid par place karne ke liye specify karta hai.

Example: Grid Layout

css Copy code

```css
.container { display: grid; grid-template-columns: 1fr 1fr 1fr; gap: 20px; }
```

---

7. Transitions and Animations

CSS Transitions

- States ke beech smooth changes allow karte hain.

css Copy code

```
button { background-color: blue; transition: background-color 0.3s ease; } button:hover { background-color: green; }
```

CSS Animations

- `@keyframes` : Animation sequence ko define karta hai.
- `animation` : Animation ko element par apply karta hai.

```css
@keyframes fadeIn { from { opacity: 0; } to { opacity: 1; } } div { animation: fadeIn 2s ease-in-out; }
```

---

8. Media Queries

- **Media queries** allow karti hain ki styles ko conditions jaise screen width, height, orientation, etc. ke basis par apply kiya jaye.
- Ye **responsive designs** banane mein madad karti hain.

```css
@media (max-width: 768px) { .container { flex-direction: column; } }
```

---

9. CSS Variables

- CSS Variables allow karti hain values ko store karna aur unhe styles mein reuse karna.

```css
:root { --main-color: #3498db; } button { background-color: var(--main-color); }
```

---

10. Best Practices

1. **External Stylesheets Ka Use Karo:** Code ko organized aur reusable rakhta hai.
2. **Flexbox aur Grid Ka Use Karo Layouts Ke Liye:** Complex layouts ko simplify karta hai.
3. **Semantic HTML Use Karo:** Accessibility aur SEO optimization ke liye zaroori hai.
4. `!important` **Ka Kam Use Karo:** Styles ko unnecessary overwrite karne se bachna chahiye.
5. **Performance Ke Liye Optimize Karo:** Heavy CSS properties ka use kam karo.

Ye notes CSS seekhne aur web pages design karte waqt aapke liye ek achhi foundation hain. In concepts ko practice karke aap flexibly aur dynamically websites bana sakte hain.