🎨CSS (Cascading Style Sheets) – Complete Detailed Notes

1. What is CSS?

CSS stands for **Cascading Style Sheets**. It describes how HTML elements should be displayed on screen, paper, or in other media. It saves time by controlling the layout of multiple web pages all at once.

- **Cascading** means that the style applied is hierarchical and can be overridden by more specific rules.
- CSS separates content (HTML) from design (CSS).

🏐2. Types of CSS

1. **Inline CSS**: Applied directly within an HTML tag using the `style` attribute.

```
<h1 style="color: red;">Hello</h1>
```

- Used for quick styling.

- Not recommended for maintainability.

- **Internal CSS**: Written within a `<style>` tag in the `<head>` section.

```
<style>
  p {
    color: blue;
  }
</style>
```

- Useful for single-page websites.

- **External CSS**: Written in a separate `.css` file and linked via `<link>`.

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

- Best for large and maintainable projects.

🏐3. CSS Syntax

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

Example:

```
h1 {
  color: blue;
  font-size: 24px;
}
```

- **Selector**: The HTML element to style.
- **Property**: The style attribute to change.
- **Value**: The setting for that property.

🏐4. CSS Selectors

- **Universal Selector**: `* {}` – Applies to all elements.
- **Element Selector**: `p {}` – Applies to all `<p>` tags.
- **Class Selector**: `.className {}` – Targets all elements with a class.
- **ID Selector**: `#idName {}` – Targets a specific element with an ID.
- **Group Selector**: `h1, h2, p {}` – Applies the same styles to multiple elements.
- **Descendant Selector**: `div p {}` – Targets `<p>` inside `<div>`.
- **Child Selector**: `div > p {}` – Targets immediate child `<p>` of `<div>`.
- **Pseudo-class**: `a:hover {}` – Styles element when user interacts with it.
- **Pseudo-element**: `p::first-line {}` – Styles part of the element.

🏐5. Box Model

All HTML elements can be considered as boxes. The box model includes:

- **Content**: The text or image.
- **Padding**: Clears an area around the content.
- **Border**: A border around the padding.
- **Margin**: Clears an area outside the border.

```
div {
  margin: 10px;
  padding: 20px;
  border: 1px solid black;
}
```

🏐6. Colors in CSS

- **Named Colors**: `red`, `blue`, `green`
- **Hexadecimal**: `#ff0000` (Red)
- **RGB**: `rgb(255, 0, 0)`
- **HSL**: `hsl(0, 100%, 50%)`

🏐7. Fonts and Text

```
font-family: Arial, sans-serif;
font-size: 16px;
font-weight: bold;
text-align: center;
text-decoration: underline;
text-transform: uppercase;
letter-spacing: 2px;
line-height: 1.5;
```

🏐8. Backgrounds

```
background-color: #f0f0f0;
background-image: url('image.jpg');
background-repeat: no-repeat;
background-position: center;
background-size: cover;
```

🏐9. Borders and Outline

```
border: 2px solid red;
border-radius: 5px;
outline: 1px dashed black;
```

🏐10. Margin and Padding

```
margin: 10px;
padding: 15px;
```

Or shorthand:

```
margin: 10px 20px 30px 40px; /* top right bottom left */
```

🏐11. Display and Visibility

```
display: block | inline | inline-block | none | flex | grid;
visibility: visible | hidden;
```

- `display: none` removes from layout.

- `visibility: hidden` hides but keeps space.

🏐12. Positioning

```
position: static | relative | absolute | fixed | sticky;
top: 10px; left: 10px;
```

- `static` : Default
- `relative` : Offset from its normal position
- `absolute` : Relative to nearest positioned ancestor
- `fixed` : Stays in place on scroll
- `sticky` : Toggles between relative and fixed

🏐13. Flexbox

```
display: flex;
flex-direction: row | column;
justify-content: center | space-between;
align-items: center;
```

Used for 1D layouts (horizontal or vertical).

🏐14. Grid

```
display: grid;
grid-template-columns: 1fr 1fr;
grid-gap: 10px;
```

Used for 2D layouts (rows and columns).

🏐15. Media Queries (Responsive Design)

```css
@media screen and (max-width: 768px) {
  body {
    background-color: lightgray;
  }
}
```

Helps design adapt to different screen sizes.

🏐16. Pseudo-classes & Elements

```css
a:hover { color: red; }
input:focus { border-color: green; }
p::first-line { font-weight: bold; }
```

🏐17. Transitions and Animations

```css
.box {
  transition: all 0.3s ease;
}

@keyframes slide {
  0% { transform: translateX(0); }
  100% { transform: translateX(100px); }
}

.box {
  animation: slide 2s infinite;
}
```

🏐18. Z-index

Controls which element appears in front:

```css
z-index: 1000;
```

Higher value = more on top.

🏐19. Units in CSS

- `px` – pixels (fixed size)
- `%` – percentage of parent
- `em` – relative to font-size of element
- `rem` – relative to root font-size
- `vh`, `vw` – viewport height/width

🏐20. Best Practices

- Use external CSS for large projects
- Use semantic class names
- Avoid inline styles
- Keep layout separate from content
- Use shorthand properties

- Test on multiple devices

Let me know if you'd like to convert this into a downloadable Word or PDF file.