

CSS Lecture Notes

1. Linking CSS to HTML

There are 3 main ways to use CSS:

a. Inline CSS

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

b. Internal CSS (in the <head>)

```
<style>
h1 {
  color: red;
}
</style>
```

c. External CSS (best practice) in a separate CSS file

- Create a file: styles.css

- Link it in HTML:

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

2. CSS Basics

CSS Syntax:

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

Example:

```
body {
  background-color: #f0f0f0;
  font-family: Arial, sans-serif;
}
```

Comments:

```
/* This is a comment */
```

Units:

- px (pixels): margin: 20px;
- % (percentage): width: 50%;
- em/rem (relative): font-size: 1.5em;
- vw/vh (viewport): width: 100vw;

3. CSS Selectors

a. Basic Selectors

```
h1 {} /* element selector */  
.class {} /* class selector */  
#id {} /* ID selector */
```

b. Grouping Selectors

```
h1, h2, h3 { color: navy; }
```

c. Descendant

```
div p { color: gray; }
```

d. Pseudo-classes

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

e. Attribute Selectors

```
input[type="text"] { border: 1px solid #ccc; }
```

4. Commonly used CSS properties

Category	Property	Example	Description
Text	color	color: red;	Text color
	font-size	font-size: 16px;	Size of text
	font-family	font-family: Arial, sans-serif;	Typeface
	text-align	text-align: center;	Horizontal text alignment
	line-height	line-height: 1.5;	Space between lines
Box Model	margin	margin: 10px;	Space outside the element
	padding	padding: 10px;	Space inside the element
	border	border: 1px solid black;	Border around the element
	width	width: 100px;	Element width
	height	height: 100px;	Element height
Background	background-color	background-color: #f0f0f0;	Background fill color

	background-image	background-image: url('bg.jpg');	Background image
	background-size	background-size: cover;	Resize background to cover/contain area
Positioning	position	position: absolute;	Positioning context (static, relative...)
	top, left, right, bottom	top: 10px;	Offset from container edge
	z-index	z-index: 1;	Stack order (higher is on top)
Display/ Layout	display	display: flex;	Box type: block, inline, flex, grid
	flex-direction	flex-direction: row;	Flex direction
	justify-content	justify-content: center;	Align flex items horizontally
	align-items	align-items: center;	Align flex items vertically
	grid-template-columns	grid-template-columns: 1fr 1fr;	Grid column layout
	gap	gap: 10px;	Space between grid/flex items
Other	overflow	overflow: auto;	What happens if content overflows
	opacity	opacity: 0.5;	Transparency level (0–1)
	cursor	cursor: pointer;	Mouse cursor style
	transition	transition: all 0.3s ease;	Smooth change of property values
	box-shadow	box-shadow: 0 4px 8px #ccc;	Shadow around box

4. Flexbox

Parent Properties:

```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
}
```

Child Properties:

```
.item {  
  flex: 1;  
  align-self: center;  
}
```

Example:

```
.container {  
  display: flex;  
  justify-content: space-between;  
}
```

5. CSS Grid

Basic Setup:

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 10px;  
}
```

Placing Items:

```
.item {  
  grid-column: 1 / 3;  
  grid-row: 2 / 4;  
}
```

Named Areas:

```
.container {  
  display: grid;  
  grid-template-areas:  
    "header header"  
    "sidebar main"  
    "footer footer";  
}
```

```
.header {  
  grid-area: header;  
}
```