HTML & CSS Cheat Sheet for Beginners

CHEAT SHEET

To use HTML

To use Css you have to have file with HTML at first. To have HTML file you should write base structure.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Page</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

This is base structure for HTML

Sheet 1: What does it mean?

- <head> - we use this to make website understand language on what website will be. Also it is used to add Css or to make tittle for your webpage
- <body> - we use this to write all the content we see on website

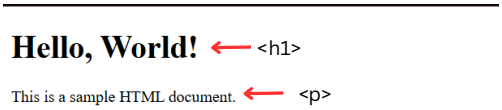
Common HTML Tags

Headings: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>.

```
<body>
    <h1>Hello, World!</h1>
</body>
```

We use this as title for something

Paragraph: <p>,



Links: <a>

We use this to put links on website

```
<body>
    <a href="https://...">Click me</a>
</body>
```

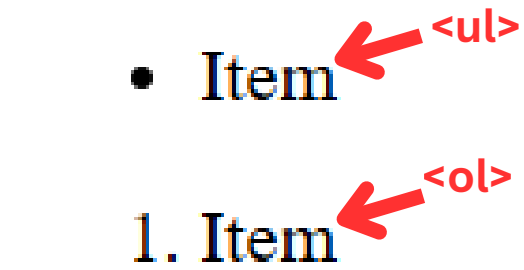
Images: <img>

We use this to put images on our website

```
<body>
    <img src="image.jpg" alt="description">
</body>
```

Lists:

- Unordered: Item
- Ordered: Item



Tables:

Tables are used to organize data in a structured way

```
<body>
    <table>
        <tr>
            <th>Header</th>
            <td>Data</td>
        </tr>
    </table>
</body>
```

Forms:

A <form> is used to create interactive user interfaces on a website to collect, enter, and submit user data, like text, checkboxes, and radio buttons, to a server

```
<body>
    <form>
        <input type="text" placeholder="Name">
        <input type="email" placeholder="Email">
        <button type="submit">Send</button>
    </form>
</body>
```

Semantic HTML5 Elements

<header> → top section of the page

<nav> → navigation menu

<main> → main content area

<section> → thematic section

<article> → independent content block

<aside> → sidebar content

<footer> → bottom section of the page

Adding CSS to HTML

Standart metod to add stylesheet:

<link rel="stylesheet" href="">

inside (href="") we should write css file, usually we call it "style.css"

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    <title>Document</title>
    <link rel="stylesheet" href=""> <!-- Missing href
</head>
```

we should write it to <head>

Other methods:

Internal style:

<style>

p { color: blue; }

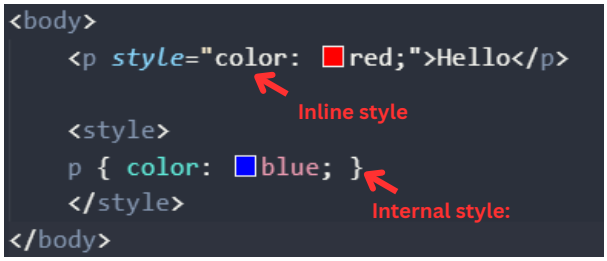
</style>

This way we add css in html file. We can add it both to the <head> and to the <body>

Inline style:

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

This way we use css just in the line or block we need, but we can't make there any global changes. Also we can write it only in <body>



Basic CSS Properties

CSS ***property*** is a parameter used in a CSS declaration that lets you style certain aspects of selected elements. For example, the opacity property is used to set the opacity of a selected element, allowing you to control if content behind that element is visible

```
/* Set 0.8 opacity on <img> elements */
img {
  opacity: 0.8;
}
```

Each property has a name (e.g., opacity), a value (e.g., 0.8), and a defined behavior on the rendering of the document. CSS also defines shorthand properties, so you can specify multiple related properties in a single declaration. For example, the margin property is a shorthand for margin-top, margin-right, margin-bottom, and margin-left, setting the margin of all four sides of an element

Syntax:

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

Here is how it is:

```css
/* Text */
h1 { color: red; font-size: 32px; text-align: center; }

/* Background & Colors */
body { background-color: #f0f0f0; color: #333; }

/* Margins & Padding */
div { margin: 20px; padding: 10px; }

/* Borders */
p { border: 1px solid black; border-radius: 5px; }

/* Size */
img { width: 200px; height: auto; }
```

CSS Box Model

Every element consists of:

- **Content** → text, image, etc.
- **Padding** → space inside the box
- **Border** → line around the element
- **Margin** → space outside the box

```css
div {
  margin: 10px;
  border: 2px solid black;
  padding: 5px;
  width: 100px;
}
```

CSS Selectors

- p → selects all <p>
- .class → selects elements with that class
- #id → selects element with that ID
- div p → selects all <p> inside
- div > p → selects only direct children
-

Useful Text & Font Styles

```css
p {
  font-family: Arial, sans-serif;
  font-size: 16px;
  font-weight: bold;
  text-transform: uppercase;
  text-decoration: underline;
  line-height: 1.5;
  letter-spacing: 1px;
}
```

Flexbox Essentials

```css
.container {
  display: flex;
  justify-content: space-between; /* horizontal alignment */
  align-items: center; /* vertical alignment */
}
```

justify-content: flex-start | center | space-between | space-around

align-items: flex-start | center | flex-end | stretch

CSS Grid Basics

```css
.container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr; /* 3 equal columns */
  gap: 10px;
}
```

- grid-template-columns: set columns (px, %, fr)
- grid-template-rows: set rows
- gap: space between grid items

Positioning Elements

```css
div {
  position: relative; /* default */
  top: 10px;
  left: 20px;
}
```

- static (default)
- relative (relative to normal position)
- absolute (relative to nearest positioned parent)
- fixed (relative to viewport)
- sticky (switches between relative/fixed depending on scroll)

Quick Tips

✅ Use semantic HTML for better SEO.
✅ Always add alt text for images.
✅ Use classes for reusable styles.
✅ Keep CSS organized with comments.
✅ Test responsiveness with media queries.