

# Web Development Quick Reference

## HTML Essentials

### Basic Structure

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Page Title</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <script src="script.js"></script>
</body>
</html>
```

### Common Tags

- `<div>` - Block container
- `<span>` - Inline container
- `<h1>` to `<h6>` - Headings
- `<p>` - Paragraph
- `<a href="url">` - Link
- `` - Image
- `<ul><li></li></ul>` - Unordered list
- `<ol><li></li></ol>` - Ordered list
- `<button>` - Button
- `<input type="text">` - Input field
- `<form action="" method="POST">` - Form

### Form Elements

html

```
<input type="text" name="username" id="user" placeholder="Enter name">
<input type="email" required>
<input type="password">
<input type="checkbox" checked>
<input type="radio" name="group">
<textarea rows="4"></textarea>
<select><option value="1">One</option></select>
<button type="submit">Submit</button>
```

## Semantic HTML

- `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<aside>`, `<footer>`
- 

# CSS Essentials

## Selectors

css

```
element { }      /* Tag */
.class { }       /* Class */
#id { }          /* ID */
element.class { } /* Combined */
parent > child { } /* Direct child */
element:hover { } /* Pseudo-class */
element::before { } /* Pseudo-element */
[attribute] { }   /* Attribute */
```

## Box Model

css

```
.box {
  width: 200px;
  height: 100px;
  padding: 10px;    /* Inside spacing */
  margin: 20px;    /* Outside spacing */
  border: 2px solid #000;
  box-sizing: border-box; /* Includes padding/border in width */
}
```

## Display & Position

css

```
display: block | inline | inline-block | none | flex | grid;  
position: static | relative | absolute | fixed | sticky;  
/* Positioning properties: top, right, bottom, left */
```

## Flexbox

css

```
.container {  
  display: flex;  
  flex-direction: row | column;  
  justify-content: flex-start | center | space-between | space-around;  
  align-items: flex-start | center | stretch;  
  flex-wrap: wrap;  
  gap: 10px;  
}  
.item { flex: 1; } /* Grow to fill space */
```

## Grid

css

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr; /* or repeat(3, 1fr) */  
  grid-template-rows: auto;  
  gap: 20px;  
}
```

## Common Properties

css

```
color: #333 | rgb(0,0,0) | rgba(0,0,0,0.5);  
background-color: blue;  
background-image: url('image.jpg');  
font-size: 16px | 1rem | 1em;  
font-weight: bold | 700;  
text-align: center | left | right;  
border-radius: 5px;  
opacity: 0.5;  
z-index: 10;
```

## Media Queries

css

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

## JavaScript Essentials

### Variables

js

```
let x = 5;      // Block-scoped, reassignable  
const y = 10;    // Block-scoped, constant  
var z = 15;     // Function-scoped (avoid)
```

### Data Types

js

```
let str = "text";  
let num = 42;  
let bool = true;  
let arr = [1, 2, 3];  
let obj = { key: "value", age: 25 };  
let nothing = null;  
let undef = undefined;
```

### Functions

js

```
function greet(name) { return "Hello " + name; }  
const greet = (name) => "Hello " + name; // Arrow function
```

### DOM Manipulation

js

```
document.getElementById("id");
document.querySelector(".class");
document.querySelectorAll("div");
element.textContent = "New text";
element.innerHTML = "<b>HTML</b>";
element.classList.add("active");
element.classList.remove("active");
element.classList.toggle("active");
element.style.color = "red";
element.setAttribute("data-id", "5");
element.getAttribute("data-id");
```

## Events

js

```
element.addEventListener("click", function(e) {
  console.log(e.target);
});
// Common events: click, submit, change, input, keydown, load
```

## Creating/Removing Elements

js

```
let div = document.createElement("div");
div.textContent = "Hello";
parent.appendChild(div);
parent.removeChild(element);
element.remove();
```

## Arrays

js

```
arr.push(4);      // Add to end
arr.pop();        // Remove from end
arr.shift();      // Remove from start
arr.unshift(0);   // Add to start
arr.length;
arr[0];          // Access by index
arr.forEach(item => console.log(item));
arr.map(x => x * 2); // Returns new array
arr.filter(x => x > 5);
arr.find(x => x === 3);
```

## Conditionals & Loops

js

```
if (x > 5) { } else if (x === 5) { } else { }
for (let i = 0; i < 10; i++) { }
while (condition) { }
arr.forEach(item => { });
```

## String Methods

js

```
str.length;
str.toUpperCase();
str.toLowerCase();
str.includes("text");
str.split(" ");
str.trim();
str.slice(0, 5);
```

## Local Storage (Browser)

js

```
localStorage.setItem("key", "value");
localStorage.getItem("key");
localStorage.removeItem("key");
localStorage.clear();
// Store objects: localStorage.setItem("obj", JSON.stringify(obj));
// Retrieve: JSON.parse(localStorage.getItem("obj"));
```

## Form Handling

js

```
form.addEventListener("submit", (e) => {
  e.preventDefault(); // Stop page reload
  let value = input.value;
});
```

## Bootstrap 5 Essentials

### CDN Links

html

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

### Grid System

html

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Half width on medium+</div>
    <div class="col-md-6">Half width on medium+</div>
  </div>
</div>
<!-- col-12 col-sm-6 col-md-4 col-lg-3 --&gt;</pre>
```

### Common Classes

Spacing: m-3 (margin), p-3 (padding), mt-2 (margin-top), px-4 (padding x-axis)

Numbers: 0-5 (0, 0.25rem, 0.5rem, 1rem, 1.5rem, 3rem)

Text: text-center, text-start, text-end, text-primary, fs-1 to fs-6

Display: d-none, d-block, d-flex, d-grid, d-inline

Flex: justify-content-center, align-items-center, flex-column

Background: bg-primary, bg-success, bg-danger, bg-light, bg-dark

Border: border, rounded, rounded-circle

Width: w-25, w-50, w-75, w-100

## Components

```
html
```

```
<!-- Button -->
```

```
<button class="btn btn-primary">Click</button>
```

```
<!-- Card -->
```

```
<div class="card">
```

```
  <div class="card-body">
```

```
    <h5 class="card-title">Title</h5>
```

```
    <p class="card-text">Content</p>
```

```
  </div>
```

```
</div>
```

```
<!-- Navbar -->
```

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
```

```
  <div class="container-fluid">
```

```
    <a class="navbar-brand" href="#">Brand</a>
```

```
  </div>
```

```
</nav>
```

```
<!-- Modal -->
```

```
<button data-bs-toggle="modal" data-bs-target="#myModal">Open</button>
```

```
<div class="modal fade" id="myModal">
```

```
  <div class="modal-dialog">
```

```
    <div class="modal-content">
```

```
      <div class="modal-header">
```

```
        <h5 class="modal-title">Title</h5>
```

```
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
```

```
      </div>
```

```
      <div class="modal-body">Content</div>
```

```
    </div>
```

```
  </div>
```

```
</div>
```

## Quick Tips

- IDs are unique, use classes for multiple elements
- Use `====` not `==` in JS for strict equality
- `querySelector` returns first match, `querySelectorAll` returns all
- `let` and `const` over `var`
- Always use `e.preventDefault()` on form submit to stop page reload
- Box-sizing: border-box makes sizing easier
- Mobile-first: style mobile, then add media queries for larger screens