

# Cheat Sheet for comprehensive CIW Advanced HTML5 & CSS3 Specialist

## HTML5 Structure and Semantics

### Basic Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>
  <!-- Content goes here -->
</body>
</html>
```

### Semantic Elements

- **<header>**: Represents introductory content or a set of navigational links.
- **<nav>**: Defines a section with navigation links.
- **<main>**: Specifies the main content of a document.
- **<article>**: Represents a self-contained composition in a document.
- **<section>**: Defines a section in a document.
- **<aside>**: Represents content aside from the content it is placed in.
- **<footer>**: Defines a footer for a document or section.

## Forms and Input Elements

### Form Attributes

- **action**: URL where form data is sent.
- **method**: HTTP method (`GET` or `POST`).
- **enctype**: Encoding type for form data.

### Input Types

- **<input type="text">**: Single-line text input.

- `<input type="password">`: Password input.
- `<input type="checkbox">`: Checkbox.
- `<input type="radio">`: Radio button.
- `<input type="submit">`: Submit button.
- `<input type="date">`: Date input.
- `<input type="email">`: Email input.
- `<input type="number">`: Number input.
- `<input type="file">`: File upload.

#### Form Validation

- `required`: Makes the input mandatory.
- `pattern`: Specifies a regular expression for validation.
- `min` and `max`: Sets minimum and maximum values for number inputs.

#### Multimedia Elements

##### Audio and Video

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

```
<video controls width="320" height="240">
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

##### Attributes

- `controls`: Adds play, pause, and volume controls.
- `autoplay`: Starts playing the media automatically.
- `loop`: Loops the media.
- `muted`: Mutes the media.

## CSS3 Basics

### Selectors

- **Element Selector:** `p { color: red; }`
- **Class Selector:** `.classname { font-size: 16px; }`
- **ID Selector:** `#idname { background-color: yellow; }`
- **Attribute Selector:** `input[type="text"] { border: 1px solid black; }`

### Box Model

- **Content:** Actual content area.
- **Padding:** Space inside the content.
- **Border:** Surrounds the padding and content.
- **Margin:** Space outside the border.

### Units

- **px:** Pixels.
- **em:** Relative to the font-size of the element.
- **rem:** Relative to the font-size of the root element.
- **%:** Percentage relative to the parent element.

## CSS3 Advanced Features

### Flexbox

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

### Grid Layout

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

## Transitions and Animations

```
.element {
  transition: all 0.5s ease;
}

.element:hover {
  transform: scale(1.1);
}

@keyframes example {
  0% { opacity: 0; }
  100% { opacity: 1; }
}
```

## Responsive Design

### Media Queries

```
@media (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

### Viewport Meta Tag

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

## Accessibility

### ARIA Roles and Attributes

- `role="navigation"`: Indicates the element is a navigation landmark.
- `aria-label="Close"`: Provides a label for an element.
- `aria-hidden="true"`: Hides an element from screen readers.

### Accessible Forms

- Use `<label>` elements with `for` attribute.
- Ensure form elements have descriptive `alt` text.

## JavaScript and DOM Manipulation

### Basic DOM Manipulation

```
document.getElementById('myId').innerHTML = 'New Text';  
document.querySelector('.myClass').style.color = 'blue';
```

### Event Handling

```
document.getElementById('myButton').addEventListener('click',  
function() {  
    alert('Button clicked!');  
});
```

## Best Practices

### HTML

- Use semantic elements for better structure and SEO.
- Validate your HTML to ensure compatibility.

### CSS

- Organize CSS with comments and logical sections.
- Use shorthand properties where possible.

### JavaScript

- Minimize DOM manipulation for performance.
- Use strict mode for better error handling.

## Tools and Resources

### Development Tools

- **Chrome DevTools:** For debugging and inspecting elements.
- **VS Code:** Popular code editor with extensions for HTML, CSS, and JavaScript.

### Online Resources

- **MDN Web Docs:** Comprehensive documentation for web technologies.
- **W3Schools:** Tutorials and references for HTML, CSS, and JavaScript.

## Examples

### Responsive Navigation Menu

```
<nav class="navbar">  
  <ul>  
    <li><a href="#">Home</a></li>
```

```

        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
    </ul>
</nav>

<style>
.navbar ul {
    display: flex;
    list-style: none;
}

.navbar li {
    margin: 0 10px;
}

@media (max-width: 600px) {
    .navbar ul {
        flex-direction: column;
    }
}
</style>

```

### ***Accessible Form***

```

<form>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <button type="submit">Submit</button>
</form>

```

This cheat sheet provides a comprehensive overview of essential HTML5 and CSS3 concepts, best practices, and tools for a CIW Advanced HTML5 & CSS3 Specialist.

By Ahmed Baheeg Khorshid

ver 1.0