

# **AAQ IT UNIT 3: WEBSITE DEVELOPMENT CHEAT SHEET**

*A student-friendly code bank for building modern, responsive, and interactive features*

## 1. Full Document Structures (The Boilerplates)

---

### A. The Clean HTML Blueprint (`index.html`)

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Rons Tech Hub Home</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div id="main-container">
    <div id="header">
      <div id="logo-box">
        
      </div>
      <div id="carousel-box">
        <button class="carousel-btn" onclick="manualMove(-1)">◀</button>
        <div id="carousel-content">Loading Latest Tech News...</div>
        <button class="carousel-btn" onclick="manualMove(1)">▶</button>
      </div>
      <div id="search-box">
        <input type="text" placeholder="Search...">
      </div>
    </div>

    <div id="nav">
      <a href="#movies" class="btn-pill">Movies</a>
      <a href="#tv-shows" class="btn-pill">TV Shows</a>
      <a href="#anime" class="btn-pill">Anime</a>
      <a href="#games" class="btn-pill">Games</a>
    </div>

    <div id="left-column">
      <div id="video-container">
        <iframe src="https://www.youtube.com/embed/TdvVj1XNBg4" referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>
      </div>
      <div id="video-text">Video description text goes here.</div>
    </div>

    <div id="right-column">
      <div class="content-card">
        <div class="card-header">
          <h3>Top Movie</h3>
          <div class="card-preview"></div>
        </div>
        <p>Interstellar trending details go here.</p>
      </div>
    </div>

    <div id="footer">
      <div class="social-links">
        <a href="https://facebook.com" target="_blank">Facebook</a>
      </div>
    </div>
  </div>

```

```

        <a href="https://youtube.com" target="_blank">YouTube</a>
        <a href="https://tiktok.com" target="_blank">TikTok</a>
    </div>
</div>
</div>

<script>
    const headlines = [
        "Interstellar: Ultimate Sci-Fi Movie Choice",
        "The Flash: Action-Packed TV Show Streaming Now",
        "Bleach: Must-Watch Anime Return",
        "GTA VI: Most Anticipated Game of the Year"
    ];
    let index = 0;
    let timer = setInterval(autoMove, 3000);

    function updateDisplay() {
        document.getElementById("carousel-content").textContent = headlines[index];
    }

    function autoMove() {
        index = (index + 1) % headlines.length;
        updateDisplay();
    }

    function manualMove(direction) {
        clearInterval(timer);
        index = (index + direction + headlines.length) % headlines.length;
        updateDisplay();
        timer = setInterval(autoMove, 3000);
    }
    updateDisplay();
</script>
</body>
</html>

```

## B. The Smart Responsive Stylesheet (style.css)

```

@charset "utf-8";

body {
  font-family: 'Space Mono', monospace;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

img, video {
  max-width: 100%;
  height: auto;
  display: block;
}

#main-container {
  display: flex;
  flex-wrap: wrap;
  flex-direction: column;
  padding: 10px;
  gap: 15px;
}

#header, #nav, #left-column, #right-column, #footer {
  width: 100%;
}

#logo {
  max-width: 180px;
  display: block;
  margin: 0 auto 10px auto;
}

#carousel-box {
  display: flex;
  justify-content: space-between;
  align-items: center;
  border: 2px solid #333;
  padding: 15px;
  background-color: #fff;
  gap: 10px;
}

.carousel-btn {
  background: #e0e0e0;
  border: 1px solid #333;
  font-size: 16px;
  padding: 5px 10px;
  cursor: pointer;
  border-radius: 4px;
}

```

```
#carousel-content {
  flex: 1;
  text-align: center;
  font-weight: bold;
}

#search-box input {
  width: 100%;
  padding: 8px;
  border: 2px solid #333;
  box-sizing: border-box;
}

#nav {
  display: flex;
  gap: 10px;
  padding: 10px 0;
  overflow-x: auto;
}

.btn-pill {
  padding: 6px 14px;
  border: 2px solid #333;
  border-radius: 15px;
  text-decoration: none;
  color: #333;
  white-space: nowrap;
  text-align: center;
  background-color: #fff;
}

#video-container {
  position: relative;
  padding-bottom: 56.25%;
  height: 0;
  overflow: hidden;
  background-color: #000;
  border: 2px solid #333;
}

#video-container iframe {
  position: absolute;
  top: 0; left: 0;
  width: 100%; height: 100%;
  border: none;
}

#video-text {
  border: 2px solid #333;
  padding: 10px;
  margin-top: 15px;
  background-color: #fff;
}
```

```

}

#right-column {
  display: grid;
  grid-template-columns: 1fr;
  gap: 15px;
}

.content-card {
  border: 2px solid #333;
  padding: 15px;
  background: #fff;
}

.card-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 10px;
}

.card-preview {
  width: 70px;
  height: 30px;
  background-color: #eee;
  border: 1px solid #aaa;
}

#footer {
  border-top: 2px solid #333;
  padding: 20px 0;
  margin-top: 20px;
}

.social-links {
  display: flex;
  justify-content: center;
  gap: 20px;
}

.social-links a {
  text-decoration: none;
  color: #333;
  font-weight: bold;
}

@media (min-width: 1024px) {
  #main-container {
    flex-direction: row;
    gap: 20px;
    padding: 20px;
  }
}

```

```

#header {
  display: grid;
  grid-template-columns: 180px 1fr 300px;
  gap: 20px;
  align-items: center;
}
#nav {
  width: 15%;
  flex-direction: column;
  gap: 12px;
  overflow-x: visible;
}
#left-column {
  flex: 1;
}
#right-column {
  width: 28%;
}
}

```

## 2. HTML Implementation Breakdown (Simple Language Guide)

---

### A. The Main Layout Frame (<div id="main-container">)

**What it is:** A giant parent container block that acts as a cardboard box holding every single section of your website together.

**How it works:** By grouping everything inside this single tag, you can easily control the dimensions and alignment of your whole website page using CSS layout systems.

### B. Image Branding Blocks (<img id="logo">)

**What it is:** An image tag used to load external picture asset files onto your webpage, like your channel or business logo graphics.

**How it works:** The src attribute tells the code exactly where your image file lives inside your project directory folders. The alt attribute provides description backup text if the picture fails to link up properly.

### C. Interactive Ticker Buttons (<button class="carousel-btn">)

**What it is:** Custom click blocks positioned on the outer edges of your sliding announcement area.

**How it works:** The onclick="" attribute listens for human interactions. When tapped on a phone or clicked with a mouse track, it runs a background JavaScript action code to swap the news slide.

#### D. Navigation Link Bars (<a class="btn-pill">)

**What it is:** Custom hyperlink routing tracks styled into modern rounded menu buttons.  
**How it works:** The href="" tracks path file tags (like #movies) or alternative page files. This structure uses an overall class that transforms normal underline text links into neat oval capsule blocks using style parameters.

#### E. Media Embedding Players (<iframe>)

**What it is:** A secure video streaming viewport cut into your left page layout block.  
**How it works:** The src rule points to a secure embedded server path link. The allowfullscreen toggle ensures users can blow up the frame to monitor sizes when playing media files.

#### F. External Social Links (target="\_blank")

**What it is:** Regular anchor text links sitting inside your page footer to connect to outside channels.  
**How it works:** Adding the critical attribute target="\_blank" forces outside sites to load inside a separate new viewing window tab. This stops users from closing out your website portfolio.

### 3. CSS Implementation Breakdown (Simple Language Guide)

---

#### A. Global Page Styling (body)

**What it does:** Sets up the basic font choices for the entire page and strips away default browser gaps.  
**Why we use it:** It sets the typography layout look and stops accidental blank row tracks from misaligning your structural borders.

#### B. Image Constraint Boundaries (img, video)

**What it does:** Standardises media behavior so that massive source picture assets don't overflow the screen bounds.  
**Why we use it:** `max-width: 100%` dynamically downscales layouts on small devices while `height: auto` preserves proper object dimensions.

#### C. Mobile Stack Engine (#main-container)

**What it does:** Sets up the vertical element order system before desktop laptop sizes are computed.  
**Why we use it:** Using `flex-direction: column` commands components to stack cleanly downward in a single file track, making reading simple on smartphone touchscreens.

## D. Phone Navigation Overflows (#nav)

**What it does:** Creates an horizontal swipe container panel if link pill elements trail off-screen on phones.

**Why we use it:** `overflow-x: auto` allows long menu button chains to safely scroll horizontally via touch gestures instead of crashing out side layout boxes.

## E. Capsule Button Radii (.btn-pill)

**What it does:** Smooths out link borders and styles individual links into touch-friendly oval buttons.

**Why we use it:** `border-radius: 15px` generates the signature curved aesthetic, while `white-space: nowrap` halts text wrapping if text length expands.

## F. Widescreen Ratio Lock (#video-container)

**What it does:** Locks embedded player components into an absolute 16:9 box scaling rule ( $16 \div 9 = 0.5625$ ).

**Why we use it:** Using a `padding-bottom: 56.25%` technique builds a responsive container window that shrinks fluidly without displaying ugly black horizontal crop bars.

## G. Desktop Layout Trigger (@media (min-width: 1024px))

**What it does:** Detects if the page viewport is opened on big screens and instantly builds a premium side-by-side framework.

**Why we use it:** Overrides basic vertical mobile stacks by flipping layout properties to rows, snapping elements into structural multi-column grids.

# 4. JavaScript Implementation Breakdown (Simple Language Guide)

---

## A. Data Storage Arrays (const headlines = [...])

**What it is:** A standard code inventory bank used to hold a series of text strings together in one group.

**How it works:** It acts as a digital index card deck. Each text announcement row inside the brackets represents a numbered item position starting from zero.

## B. Automated Interval Engines (setInterval)

**What it is:** A built-in loop trigger command that manages automated page updates behind the scenes.

**How it works:** It counts down time parameters using milliseconds ( $3000\text{ms} = 3\text{s}$ ).

Every time the clock reaches zero, it calls up your slider function code blocks to swap headings smoothly.

### C. Content Index Wrappings (%)

**What it is:** A remainder calculation trick used to recycle slide queues back to entry one automatically.

**How it works:** Using the modulo mathematical sign `%` checks your total array size. When your slide tracker reaches item four, it prevents a boundary fault error by flipping your tracking index back to zero.

### D. Manual Control Resets (`clearInterval`)

**What it is:** A critical execution block that blocks automated intervals if a user overrides system timing.

**How it works:** If a human clicks a manual left/right indicator button, it resets the active countdown immediately. This blocks glitchy animation overlap loops and starts a fresh, clean 3-second cycle.