

| Web Design Principles | | | HTML/CSS Example | | | | Concept | Explanation | HTML/CSS Example | |
|---|---|---|------------------|--|--|---|---------------------------|---|---|--|
| Principle | Explanation | Example | Image Formats | | | | Basic Table | Organize data into rows and columns. | <table>tr</tr><td>1</td></tr><td>2</td></tr></table> | |
| Repetition | Repeat visual elements (colors, fonts, styles) for consistency. | Use the same font and color scheme for all headings: h1, h2 (font-family: Arial; color: navy;). | Format | Full Name | Key Features | Best Use Cases | Table Caption | Add a title or description to a table. | <caption>Table Title</caption> | |
| Contrast | Use contrasting colors, sizes, or fonts to make elements stand out. | h1 { color: white; background: black; } | GIF | Graphics Interchange Format | • Supports 256 colors. • Supports transparency. • Supports animation. | • Simple animations. • Small icons or logos with solid colors. | Colspan | Merge cells horizontally. | <td colspan="2">Spanned Cell</td></tr> | |
| Proximity | Group related items together for better organization. | <div class="contact-info">Phone: 123-456-7890</div> | Inline CSS | Apply styles directly to an element using the style attribute. | <h1 style="color: red;">Heading</h1> | | Rowspan | Merge cells vertically. | <td rowspan="2">Spanned Cell</td></tr> | |
| Alignment | Align elements to create a clean, organized layout. | <div style="text-align: center;"> | Embedded CSS | Define styles in the <head> section for a single page. | <style> p { color: blue; } </style> | | Audio | Embed audio files with controls. | <audio controls>source src="audio.mp3" type="audio/mpeg"></audio> | |
| WCAG POUR Principles (web content accessibility principles) | | | External CSS | Link to an external .css file for site-wide styling. | <link rel="stylesheet" href="styles.css"> | | Video | Embed video files with controls. | <video controls>source src="video.mp4" type="video/mp4"></video> | |
| Principle | Explanation | Example | Class Selector | Apply styles to multiple elements with the same class. | <style> .red { color: red; } </style> <p class="red">Red Text</p> | PNG | Portable Network Graphics | Supports millions of colors. • High-quality graphics with sharp edges. | YouTube Embed | <iframe src="https://www.youtube.com/embed/VIDEO_ID"></iframe> |
| Perceivable | Content must be presented in ways users can perceive (see or hear). | Use alt text for images: | ID Selector | Apply styles to a unique element using its ID. | <style> #header { background: yellow; } </style> <div id="header">Header</div> | | | | | |
| Operable | Users must be able to navigate and interact with the content. | Ensure keyboard navigation: Link | Descendant | Apply styles to elements within a container. | <style> footer p { color: green; } | | | | | |

| | | | | | |
|-------------------------|--|---|---|---|---|
| Understandable | Interact with the interface. Content and navigation must be clear and easy to understand. | <code>href= "#main" #to click the button</code> Use simple language: <code><a href= "#main" #to click the button</code> to submit the form. | Selector Specific container. <code><div></code> Div Element Block-level container for grouping elements. <code><div class="container">Content</div></code> | | Lecture 1E: CSS Box Model |
| Robust | Content must work across devices, browsers, and assistive technologies. | Use valid HTML5: <code><!DOCTYPE html></code> | Span Element Inline container for styling specific text. <code>Inline Text</code> | | <p>Pseudo-classes are used to define a special state of an element. They are prefixed with a colon (:)</p> <p>Pseudo-Class Explanation</p> <ul style="list-style-type: none"> :hover Applies styles when the user hovers over an element. <code>:hover {color: red;}</code> :focus Applies styles when an element receives focus (e.g., input fields). <code>input:focus {background: #2px solid blue;}</code> :active Applies styles when an element is being activated (e.g., clicked). <code>button:active {background: green;}</code> :visited Applies styles to links that have been visited (e.g., color purple). :first-child Applies styles to the first child of a parent element. <code>:first-child {font-weight: bold;}</code> :last-child Applies styles to the last child of a parent element. <code>:last-child {color: blue;}</code> :nth-child (n) Applies styles to elements based on their position in a parent element. <code>:nth-child(2) {background: yellow;}</code> :not (selector) Applies styles to elements that do not match a selector. <code>:not (special) {color: gray;}</code> |
| Description List | Displays terms and their descriptions. | <code><dl><dt>Term</dt><dd>Definition</dd></dl></code> | Concept Explanation | HTML/CSS Example | |
| | | | Unordered List Displays a list of items without a specific order. | <code>Item 1Item 2</code> | |
| | | | Ordered List Displays a list of items with a specific order. | <code>Item 1Item 2</code> | |
| | | | Definition List Displays a list of terms and their definitions. | <code><dl><dt>Term</dt><dd>Definition</dd></dl></code> | |
| | | | Form Elements Displays form elements like text inputs, buttons, etc. | <code><input type="text"><input type="button" value="Submit"></code> | |
| | | | Table Displays data in a tabular format. | <code><table><tr><th>Header</th><td>Data</td></tr></table></code> | |

| | | | | | | | | | | | |
|--------------------|---|---|------------------|--|--|-----------------|--|--|-------------------|--|--------------------------------------|
| Special Characters | Use entities to display reserved characters. | © (©), < (<), > (>), & (&), (&#sp;), ‑ (‑) | Image Tag | Set an image as the background of an element. The following code sets the background image of a paragraph to a small graphic with optional dimensions and alt text. | <pre><p style="background-image: url('small.jpg'); width: 100px; height: 180px;"></pre> | Rounded Corners | Create rounded edges for elements. | <pre>div { border-radius: 18px; }</pre> | Topic | Description | Example |
| Anchor Tag | Creates hyperlinks. | <code></code> | Background Image | Set an image as the background of an element. | <pre>body { background-image: url('top.jpg'); }</pre> | Box Shadow | Add shadow effects to elements. | <pre>div { box-shadow: 5px 5px 5px gray; }</pre> | Normal Flow | Browser displays elements in the order they are coded. | box1.html, box2.html |
| Mailto Link | Opens the default email client. | <code>Email Me</code> | Image Link | Use an image as a hyperlink. | <pre></pre> | Text Shadow | Add shadow effects to text. | <pre>h1 { text-shadow: 3px 3px 3px gray; }</pre> | Floating Property | Makes elements "float" on the left/right side of the browser or another element. | <code>img { float: left; }</code> |
| Telephone | Initiates a phone call on supported devices. | <code>Call Us</code> | Favicon | Small icon displayed in the browser tab. | <pre><link rel="icon" href="favicon.ico" type="image/x-icon"></pre> | Opacity | Control the transparency of an element. | <pre>div { opacity: 0.5; }</pre> | Clear Property | Terminates a float. | <code>footer { clear: both; }</code> |
| Message Links | Links to initiate phone calls or text messages on mobile devices. | <code>Call <code>Text</code></code> | Image Map | Define clickable areas within an image. | <pre><map name="map"><area shape="rect" coords="0,0,180,180" href="link.html"></map></pre> | Gradient | Smooth blending of colors on a background. | <pre>body { background: linear-gradient(to bottom, white, blue); }</pre> | Overflow Property | Configures display of elements; clears floats before container ends. | <code>div { overflow: auto; }</code> |

| | | | | | | | | |
|------------------|---|--|---|----------------------|---|--|-------------|---------|
| Basic Layout | Divides content into two columns. | <pre><left { float: left; width: 50%; }> <right { float: right; width: 50%; }></pre> | <ul style="list-style-type: none">3. <code>focus</code>: Applies when the hyperlink is focused (e.g., via keyboard navigation).4. <code>hover</code>: Applies when the user hovers over the hyperlink with the mouse.5. <code>active</code>: Applies when the hyperlink is being clicked or activated. | Concept | Syntax/Example | // Example 3: Start/end indices (indices are swapped) <pre> console.log(str.substring(12, 7)); // Output: "World!" // Example 4: Negative start/end indices (e.g., from the end) console.log(str.substring(-5, 5)); // Output: "Hello!"</pre> | | |
| Display Property | Controls how elements are displayed. | <pre>a { display: block; }</pre> | JavaScript is an interpreted language, not a compiled language. JavaScript is interpreted when the code is run. This interpretation has to be repeated every time the code is run. | else if Statement | <pre>else if (condition) { // code }</pre> | // Example 1: Basic usage <pre> let result = 'test slice'; // Extracts from index 0 to 4 (5 is exclusive) console.log(result.slice(0, 5)); // Output: "test"</pre> | | |
| Position-class | Styles for anchor elements are displayed. | <pre>a.link { color: #ff0000; } a:hover { text-decoration: none; }</pre> | <ul style="list-style-type: none">1. <code>interactivity</code> (dynamic content, animations): Collect user data (forms, prompts). Validate user input.2. <code>script-type="text/javascript"</code>: <code><script type="text/javascript"></code>3. <code>script-type="text/javascript"</code>: <code><script type="text/javascript"></code> | switch Statement | <pre>switch (value) { case "Y": // code: break, default: // code; }</pre> | // Example 2: Negative start/end indices <pre> let result = 'test slice'; // Extracts from index 7 (length - 6) to the end console.log(result.slice(7)); // Output: "World!"</pre> | | |
| CSS Pseudo-class | Controls element positioning. | <pre>nav { position: fixed; }</pre> | JavaScript is a weakly typed language | Comparison Operators | <pre>==, !=, >, <, >=, <=</pre> | // Example 3: Negative start and end indices <pre> let result = 'test slice'; // Extracts from index 7 to index 11 console.log(result.slice(7, 11)); // Output: "World!"</pre> | | |
| CSS Property | Single image file containing multiple small graphics. | <pre>.icon { background: url('sprite.png') -10px -20px; }</pre> | <ul style="list-style-type: none">1. <code>name</code>: <code>let name = "John";</code> // String2. <code>age</code>: <code>let age = 30;</code> // Number3. <code>isStudent</code>: <code>let isStudent = true;</code> // Boolean4. <code>let x;</code> // Undefined5. <code>let y = null;</code> // Null6. <code>let bigNum = 12345678901234567890;</code> // BigInt | Logical Operators | <pre>&& (AND), (OR), ! (NOT)</pre> | // Example 4: Start/index > end index <pre> let result = 'test slice'; // Returns an empty string console.log(result.slice(10, 5)); // Output: ""</pre> | | |
| CSS Sprites | | | | for...in Loop | <pre>while (condition) { // code }</pre> | Topic | Description | Example |

[illegible][illegible]

| | | | |
|--|---|--|--|
| <p>Array Method: concat() Concatenates two arrays.</p> <p>Array Method: slice() Copies part of an array.</p> <p>Array Method: join() Converts an array to a string.</p> <p>Array Method: sort() Sorts an array in ascending order.</p> <p>Array Method: reverse() Reverses the order of elements in an array.</p> <p>What is the BOM?</p> <ul style="list-style-type: none"> BOM (Browser Object Model): A collection of objects provided by the browser for scripting. Purpose: Allow interaction with browser features like the window, history, location, and so on. 2 Common Browser Objects Window Object: Represents the browser window. <ul style="list-style-type: none"> Example: <code>window.alert("hello")</code> Properties: <code>browserName</code>, <code>version</code>, <code>screen</code> size, etc. History Object: Tracks visited pages (history stack). <ul style="list-style-type: none"> Properties: <code>length</code> (number of pages in history). Methods: <code>back()</code>, <code>forward()</code>, etc. Location Object: Contains current page's URL, server, port, and protocol. <ul style="list-style-type: none"> Method: <code>reload()</code> (replaces current page in history). Navigator Object: Provides browser and OS details. <ul style="list-style-type: none"> Properties: <code>appName</code>, <code>userAgent</code>. Screen Object: Contains display properties (e.g., screen size, color depth). Document Object: Represents the webpage. <ul style="list-style-type: none"> Properties: <code>querySelector()</code>, <code>image</code>, <code>links</code>, etc. Collections: <code>images</code>, <code>links</code>, <code>forms</code>. | <p>Elements</p> <ul style="list-style-type: none"> <code>getElementById()</code>: Submits the form. <code>getElementByName()</code>: Resets the form. <p>Text Elements</p> <ul style="list-style-type: none"> <code>getElementById()</code>: Single-line text input. <code>getElementByName()</code>: Password input. <code>getElementById()</code>: Hidden input. <p>Text Element Attributes</p> <ul style="list-style-type: none"> size: Width of the text box. maxlength: Maximum characters allowed. value: Default value. select(): Selects all text in the text box. <p>Text Element Methods</p> <ul style="list-style-type: none"> <code>onfocus</code>, <code>onblur</code>, <code>onchange</code>, <code>onselect</code>, <code>onkeydown</code>, <code>onkeypress</code>, <code>onkeyup</code>. <p>Text Element Event Handlers</p> <p>Textarea Element</p> <ul style="list-style-type: none"> Example: <code><textarea> Multi-line text input.</code> <p>Textarea Attributes</p> <ul style="list-style-type: none"> rows: Number of rows. cols: Number of columns. wrap: Controls line wrapping. <p>Textarea Event Handlers</p> <ul style="list-style-type: none"> <code>onchange</code>, <code>onkeyup</code>, <code>onkeypress</code>, <code>onkeydown</code>. <p>Checkboxes & Radio Buttons</p> <p>Checkbox/Label to Event Handlers</p> <ul style="list-style-type: none"> <code>onfocus</code>, <code>onblur</code>, <code>onchange</code>. | <p>1. Frames:</p> <ul style="list-style-type: none"> Frames allow multiple HTML documents to be displayed as panes within a single browser window. Example: <code><frameset rows="50%,*" ><frame name="UpperWindow" src="UpperWindow.html"></frameset></code> Access frames using: <ul style="list-style-type: none"> <code>parent</code>: parent to access the parent frame. <code>window.parent.frames[0]</code> or <code>window.parent.frames["UpperWindow"]</code> to access specific frames. <p>2. Opening and Managing New Windows:</p> <ul style="list-style-type: none"> Use <code>window.open()</code> to open a new window with parameters like URL, window name, and features (e.g., width, height). Example: <code>var newWindow = window.open("test2.html", "myWindow", "width=200, height=100")</code> Add HTML to a new window using <code>document.write()</code>. Communicate between windows using <code>window.open</code> to access the original window. <p>3. Window Manipulation:</p> <ul style="list-style-type: none"> Methods for resizing and moving windows: <ul style="list-style-type: none"> <code>resizeTo(width, height)</code>, <code>moveTo(x, y)</code>, <code>resizeBy(deltaWidth, deltaHeight)</code>. <p>4. String Manipulation:</p> <ul style="list-style-type: none"> Methods like <code>split()</code>, <code>replace()</code>, <code>search()</code>, and <code>match()</code> are used for string manipulation. Example: <code>var myString = "January - December / January - February";</code> <p>5. Regular Expressions (RegExp):</p> <ul style="list-style-type: none"> Used for pattern matching, splitting, and replacing strings. Can be created using literals (<code>/pattern/</code>) or the <code>RegExp</code> constructor. Attributes like <code>g</code> (global) and <code>i</code> (case-insensitive) can be applied. Special characters (e.g., <code>\d</code>, <code>\b</code>) and grouping (<code>()</code>) are used to define complex patterns. String methods like <code>split()</code>, <code>replace()</code>, <code>search()</code>, and <code>match()</code> can be used with <code>RegExp</code>. <p>6. Date and Time:</p> <ul style="list-style-type: none"> <code>new Date()</code>: Converts the date and time to UTC. <code>toLocaleDateString()</code>: Returns the local date and time as a string. <code>getTimezoneOffset()</code>: Returns the difference in minutes between local time and UTC. <code>toLocaleTimeString()</code>: Returns the local time as a string. <code>toLocaleTimeString()</code>: Returns the time portion of the date as a string. <code>toLocaleDateString()</code>: Returns the local date as a string. <code>toLocaleString()</code>: Returns the date portion of the date as a string. <p>One-Shot Timer:</p> <ul style="list-style-type: none"> Use <code>setTimeout()</code> to trigger a function once after a specified delay. Example: <code>var timerID = setTimeout(function() { alert("Time is up!"); }, 3000);</code> triggers an alert after 3 seconds. Use <code>clearTimeout(timerID)</code> to cancel the timer. <p>Repeating Timer:</p> <ul style="list-style-type: none"> Use <code>setInterval()</code> to trigger a function repeatedly at specified intervals. Example: <code>var myIntervalID = setInterval(function() { myFunction(); }, 5000);</code> calls <code>myFunction()</code> every 5 seconds. Use <code>clearInterval(myTimerID)</code> to stop the repeating timer. <p>Key Examples:</p> <ul style="list-style-type: none"> Date Object: <code>var localTime = new Date();</code> creates a Date object for the current time. UTC Conversion: <code>localTime.toUTCString()</code> converts the date to UTC. One-Shot Timer: <code>setTimeout(function() { alert("Time is up!"); }, 3000);</code> triggers an alert after 3 seconds. | <p>Topic</p> <p>What are Cookies?</p> <p>Description</p> <p>Text files stored on a user's computer to personalize web experiences (e.g., remembering user names).</p> <p>Example</p> |
|--|---|--|--|

| | | | | | | | | | | | | | | |
|----------------------|---|--|--|--|---|--|--|--|--|--|-----------------|--|---|--|
| Cookie String | A cookie string consists of six parts: <ul style="list-style-type: none">• Name: Identifier for the cookie.• Value: Data stored in the cookie.• Expires: Expiration date for the cookie.• Path: Specifies the path on the domain where the cookie is valid.• Domain: Specifies the domain and subdomains where the cookie is valid.• Secure: Indicates if the cookie should only be sent over secure (HTTPS) connections. | <pre>document.cookie = "UserName=Paul"; document.cookie = "UserName=Paul"; document.cookie = "UserName=Paul; expires=Tue, 28 Dec 2020 00:00:00 GMT"; document.cookie = "UserName=Paul;path=/"; document.cookie = "UserName=Paul;domain=MyDomain.com"; document.cookie = "UserName=Paul;secure";</pre> | <ul style="list-style-type: none">• Example: document.getElementById("Heading").element; Represents an HTML element.• Example: var HiElement = document.getElementById("Heading"); node; Represents any node in the DOM (element, text, etc).• Example: var HiElement = HiElement.nextSibling; nodeType; Returns the type of the node (e.g., for 3 text nodes). | <ul style="list-style-type: none">• Useful Methods:<ul style="list-style-type: none">• getElementById(): Retrieves an element by its ID.• getElementByIdTagName(): Retrieves elements by tag name.• Useful Properties:<ul style="list-style-type: none">• nextSibling: Returns the next sibling node.• nodeType: Returns the type of the node (e.g., for 3 text nodes). | DHTML (Dynamic HTML) <ul style="list-style-type: none">• Definition: Combines HTML, CSS, and JavaScript to dynamically change web pages after loading.• Cross-Browser Issues: Differences in event models between browsers.<ul style="list-style-type: none">• IE Event Model:<ul style="list-style-type: none">• Uses window.event object.• Example: if (window.event.type == 'click') { ... }• Non-IE Event Model:<ul style="list-style-type: none">• Passes an event object to handlers.• Example: if (evt.type == 'click') { ... } | Remote Scripting <ul style="list-style-type: none">• Definition: Allows client-side JavaScript to request and receive data from a server without refreshing the page.• XMLHttpRequest Object: Used for making HTTP requests. | Ajax (Asynchronous JavaScript and XML) <ul style="list-style-type: none">• Definition: Enables asynchronous client-server communication without page refreshes.• Requirements: Server-side technology to output data (e.g., XML, JSON, or plain text). | DOM Navigation <ul style="list-style-type: none">• var HiElement = document.getElementById("Heading");• var HiElement = HiElement.nextSibling; | DHTML Events <ul style="list-style-type: none">• IE: if (window.event.type == 'click') { ... }• Non-IE: if (evt.type == 'click') { ... } | Remote Scripting <ul style="list-style-type: none">• Modern browsers: var xmlhttp = new XMLHttpRequest();• IE 5/6: var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); | 4. Ajax: | 2. Grid System: Bootstrap's 12-column grid system for responsive layouts. Basic Grid Structure: <code><div class="container"><div class="row"><div class="col-md-4 col-6"><div class="col-md-4 col-6"></div></div></div></code> Push and Pull <code><div class="col-md-4 col-6"><div class="col-md-4 col-6"></div></div></code> 3. Responsive Utilities: <code><div class="col-md-4 col-6"><div class="col-md-4 col-6"></div></div></code> 4. Typography: Lead Paragraph: <code><p>Test This paragraph stands out</p></code> Text Alignment: <code><p>Test Left aligned text</p><p>Test Center aligned text</p><p>Test Right aligned text</p></code> 5. Buttons: <code><button class="btn btn-primary">Primary button</button><button class="btn btn-success">Success button</button><button class="btn btn-danger">Danger button</button></code> Button Sizes: <code><button class="btn btn-lg">Large button</button><button class="btn btn-sm">Small button</button><button class="btn btn-xs">Extra Small button</button></code> Buttons with Icons: <code><button class="btn btn-default"> Search</button></code> 6. Navigation: <code><nav>HomeAboutContact</nav></code> 7. Carousel: <code><div><div></div><div></div><div></div></div><div><div><div><div><div><div></div></div></div></code> | HTML5 and Figure Used to group images with captions. <code><figure><figcaption></figcaption></figure></code> Telephone Links Links to telephone phone calls or links to mobile phone calls on devices. <code>Call 888-555-5555Call 888-555-5555</code> HTML5 Semantic Elements Elements to structure content semantically. <code><section><article><aside></aside></section></code> Floating Images Floating images with captions using CSS. <code><caption>Golden Gate Bridge</caption></div><div><caption>Golden Gate Bridge</caption></div></div></code> HTML5 Compatibility Ensures HTML5 elements work in Internet Explorer 8 and HTML5 Shim. <code><script></script></code> CSS Styling for printing Configure separate styles for printing (page-break-before: always); <code><print></code> Mobile Web Design Best Practices Techniques for optimizing websites for mobile devices. <code><meta name="viewport" content="width=device-width, initial-scale=1"></code> Viewport Meta Tag Configures the width and scale of the viewport on mobile devices. <code><meta name="viewport" content="width=device-width, initial-scale=1"></code> Responsive Web Design Techniques for creating responsive layouts. <code><div><div><div><div><div><div></div></div></div></code> CSS Media Queries Directs the browser to apply styles based on device capabilities. <code>@media only screen and (max-width: 480px) { header { background-image: url(mobile.jpg); }</code> Flexible Images Ensures images scale properly on different devices. <code>CSS: img { max-width: 100%; height: auto; }</code> iPhone (Portrait) Ensures one screen and (max-width: 414px) body { font-size: 14px; padding: 10px; } | Used to group images with captions. <code><figure><figcaption></figcaption></figure></code> Links to telephone phone calls or links to mobile phone calls on devices. <code>Call 888-555-5555Call 888-555-5555</code> Elements to structure content semantically. <code><section><article><aside></aside></section></code> Floating images with captions using CSS. <code><caption>Golden Gate Bridge</caption></div><div><caption>Golden Gate Bridge</caption></div></div></code> Ensures HTML5 elements work in Internet Explorer 8 and HTML5 Shim. <code><script></script></code> Configure separate styles for printing (page-break-before: always); <code><print></code> Techniques for optimizing websites for mobile devices. <code><meta name="viewport" content="width=device-width, initial-scale=1"></code> Configures the width and scale of the viewport on mobile devices. <code><meta name="viewport" content="width=device-width, initial-scale=1"></code> Techniques for creating responsive layouts. <code><div><div><div><div><div><div></div></div></div></code> Directs the browser to apply styles based on device capabilities. <code>@media only screen and (max-width: 480px) { header { background-image: url(mobile.jpg); }</code> Ensures images scale properly on different devices. <code>CSS: img { max-width: 100%; height: auto; }</code> Ensures one screen and (max-width: 414px) body { font-size: 14px; padding: 10px; } |
|----------------------|---|--|--|--|---|--|--|--|--|--|-----------------|--|---|--|

| | | |
|--|--|---|
| <p>DOM Document Object Model</p> <ul style="list-style-type: none"> DOM Tree Structure: Represents an HTML document as a tree, with <code><html></code> as the root node and nested elements as child nodes. <ul style="list-style-type: none"> Example: <code><body></code> is a child of <code><html></code>. Navigating the DOM: Use DOM objects, properties, and methods to manipulate the tree. <ul style="list-style-type: none"> Base Objects: <ul style="list-style-type: none"> document: Represents the entire HTML document. | <ul style="list-style-type: none"> <code>o http.open("GET", "http://localhost/myTextFile.txt", false);</code> <code>o http.onreadystatechange = function() { if (http.readyState == 4) { ... } };</code> <p>1. Bootstrap Setup Include Bootstrap CSS and JS in your HTML file: <code><!-- Bootstrap CSS --><link href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"><!-- Bootstrap JS and dependencies</code></p> | <p>resources relative to the current <code></code> <code>index.html</code> or <code>Home.html</code>.</p> <p>display: none; Hide menu on small screens *)</p> <p>*) iPhone (landscape) *)</p> <pre> @media only screen and (max-width: 736px) { header { height: 50px; } menu { display: block; // Show menu in landscape *) } } </pre> |
|--|--|---|

