|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **HTML code** | | Java code | Console.log |
| getElementByID() | <div>  <p id=”message”>javaScriptDOM</p>  </div> | | <script>  let msg = document.getElementById(“message”);  console.log(msg)  </script> | <p id=”message”>javaScriptDOM</p> |
| getElementByID  *(se selecteaza doar 1 rind, 1 element)* | <div>  <p id=”message”>javaScriptDOM</p>  <p id=”message”>javaScript</p>  </div> | | <script>  let msg = document.getElementById(“message”);  console.log(msg)  </script> | <p id=”message”>javaScriptDOM</p> |
| getElementsByName()  screen: | <div>  <p type=”radio” name=”language” value=”JS”>javaScriptDOM</p>  <p type=”radio” name=”language” value=”Python”>javaScriptDOM</p>  </div> | | <script>  let btn = document.getElementByName(“language”);  console.log(btn)  </script> | NodeList(2) [input, input]  0: input  1: input  length: 2  [[prototype]]: NodeList |
| getElementsByTagName()  screen:  first heading  second heading  third heading  fourth heading | <h1>first heading</h1>  <h1>second heading</h1>  <h1>third heading</h1>  <h1>fourth heading</h1> | | <script>  let heading = document.getElementByTagName(“h1”);  console.log(heading)  </script> | HTMLCollection(4) [h1, h1, h1, h1] |
| getElementsByClassName()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1> | | <script>  let msg = document.getElementByClassName(“message”);  console.log(msg)  </script> | HTMLCollection(3) [h1.message, h1.message, h1.message]  0: h1.message  1: h1.message  2: h1.message  length: 3  [[prototype]]: HTMLCollection |
| getElementsByClassName()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let cont = document.getElementById(“container”);  let msg = cont.getElementByClassName(“message”);  console.log(msg)  </script> | HTMLCollection(2) [h1.message, h1.message] |
| querySelector()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelector(“.message”);  console.log(msg)  </script> | <h1 class=”message”>first heading</h1> |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelectorAll(“.message”);  console.log(msg)  </script> | NodeList(3) [h1.message, h1.message, h1.message]  0: h1.message  1: h1.message  2: h1.message  length: 3 |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let cont = document.getelementById(“container”)  let msg = cont.querySelectorAll(“.message”);  console.log(msg)  </script> | NodeList(2) [h1.message, h1.message]  0: h1.message  1: h1.message  (s-au selectat din container) |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelectorAll(“#container”);  console.log(msg)  </script> | NodeList(2) [h1.message, h1.message]  0: h1.message  1: h1.message  (s-au selectat din container) |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelectorAll(“div”);  console.log(msg)  </script> | NodeList(2) [h1.message, h1.message]  0: h1.message  1: h1.message  (s-au selectat din div) |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelectorAll(“h1”);  console.log(msg)  </script> | NodeList(3) [h1, h1, h1]  0: h1  1: h1  2: h1  length: 3 |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelectorAll(“div h1”);  console.log(msg)  </script> | NodeList(2) [h1.message, h1.message]  0: h1.message  1: h1.message |
| querySelectorAll()  screen:  first heading  second heading  third heading | <h1 class=”message”>first heading</h1>  <div id=”container”>  <h1 class=”message”>second heading</h1>  <h1 class=”message”>third heading</h1>  </div> | | <script>  let msg = document.querySelectorAll(“div, h1”);  console.log(msg)  </script> | NodeList(2) [h1.message, div#container, h1.message, h1.message]  0: h1.message  1: h1.message  2: h1.message |
| Traversing elements  screen:  Welcome to GreatStack | <div class=”title”>  <p class=”text”>Welcome to GreatStack</p>  </div> | | <script>  let txt = document.querySelector(“.text”);  console.log(txt.parentNode);  </script> | <div class=”title”>…</div> |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”>  <p>Welcome to GreatStack 1</p>  <p>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | | <script>  let parent = document.querySelector(“.title”);  console.log(parent.firstChild);  </script> | #text |
| vGetting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”><p>Welcome to GreatStack 1</p>  <p>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | | <script>  let parent = document.querySelector(“.title”);  console.log(parent.firstChild);  </script> | <p>Welcome to GreatStack 1</p> |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”>  <p>Welcome to GreatStack 1</p>  <p>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | | <script>  let parent = document.querySelector(“.title”);  console.log(parent.firstElementChild);  </script> | <p>Welcome to GreatStack 1</p> |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”>  <p>Welcome to GreatStack 1</p>  <p>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | | <script>  let parent = document.querySelector(“.title”);  console.log(parent.lastElementChild);  </script> | <p>Welcome to GreatStack 4</p> |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”>  <p>Welcome to GreatStack 1</p>  <p>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | <script>  let parent = document.querySelector(“.title”);  console.log(parent.childNodes);  </script> | | NodeList(9) [text, p, text, p, text, p, text, p, text ] |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”> <p>Welcome to GreatStack 1</p>  <p>Welcome to GreatStack 2</p><p>Welcome to GreatStack 3</p><p>Welcome to GreatStack 4</p></div> | <script>  let parent = document.querySelector(“.title”);  console.log(parent.childNodes);  </script> | | NodeList(4) [p, p, p, p, ] |

|  |  |  |  |
| --- | --- | --- | --- |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”>  <p >Welcome to GreatStack 1</p>    <p class=”second”>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | <script>  let second = document.querySelector(“.second”);  console.log(second.previousElementSibling);  </script> | <p>Welcome to GreatStack 1</p> |
| Getting Child Elements of a Node  screen:  Welcome to GreatStack 1  Welcome to GreatStack 2  Welcome to GreatStack 3  Welcome to GreatStack 4 | <div class=”title”>  <p >Welcome to GreatStack 1</p>  <p class=”second”>Welcome to GreatStack 2</p>  <p>Welcome to GreatStack 3</p>  <p>Welcome to GreatStack 4</p>  </div> | <script>  let second = document.querySelector(“.second”);  console.log(second.nextElementSibling);  </script> | <p>Welcome to GreatStack 3</p> |
| (Manipulating Elements)  Creat Elements  screen:  No text on the screen(nothing) | <body>  </body> | <script>  let div = document.createElement(“.div”);  console.log(div);  </script> | <div></div> |
| (Manipulating Elements)  Creat Elements  screen:  No text on the screen(nothing) | <body>    </body> | <script>  let div = document.createElement(“.div”);  div.innerHTML = “<p>Welcome to GreatStack</p>”;  console.log(div);  </script> | <div>  <p>Welcome to GreatStack </p>  </div> |
| (Manipulating Elements)  Creat Elements  screen:  Welcome to GreatStack | <body>    </body> | <script>  let div = document.createElement(“.div”);  div.innerHTML = “<p>Welcome to GreatStack</p>”;  document.body.appenChild(div);  console.log(div);  </script> | <div>  <p>Welcome to GreatStack </p>  </div> |
| (Manipulating Elements)  Creat Elements  screen:  Welcome to GreatStack 2 | <body>  </body> | <script>  let div = document.createElement(“.div”);  div.innerHTML = “<p>Welcome to GreatStack 2</p>”;  document.body.appenChild(div);  console.log(div);  </script> | <div>  <p>Welcome to GreatStack 2</p>  </div> |
| (Manipulating Elements)  Creat Elements  screen:  No text on the screen(nothing) | <body>  </body> | <script>  let div = document.createElement(“.div”);  div.innerHTML = “<p>Welcome to GreatStack 2</p>”;  div.id = “title” ;  console.log(div);  </script> | <div id=””title”>…  <p>Welcome to GreatStack 2</p>  </div> |
| (Manipulating Elements)  Creat Elements  screen:  No text on the screen(nothing) | <body>    </body> | <script>  let div = document.createElement(“.div”);  div.innerHTML = “<p>Welcome to GreatStack </p>”;  div.className = “title” ;  console.log(div);  </script> | ( is nothing) |
| appendChild() Method  screen:  Home  About  Blog  Project  Contact | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  let list = document.createElement(“li”);  list.innerHTML = “Contact”;  menu.appenchild(list);  </script> | (is nothing) |
| textContent  screen:  Home  About  Blog  Project | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  console.log(menu.textContent);  </script> | Home  About  Blog  Project |
| textContent  screen:  Home  About  Blog  Project | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  console.log(menu.innerText);  </script> | Home  About  Blog  Project |
| textContent  screen:  Home  About  Blog | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li style=”display: none;”>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  console.log(menu.innerText);  </script> | Home  About  Blog |
| textContent  screen:  Home  About  Blog | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li style=”display: none;”>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  console.log(menu.textContent);  </script> | Home  About  Blog  Project |
| textContent  screen:  Hello | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li style=”display: none;”>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  menu.textContent = “Hello”;  </script> | ( is nothing) |
| innerHTML  screen:  Hello | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li style=”display: none;”>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  menu.innerHTML = “Hello”;  </script> | (is nothing) |
| innerHTML  screen:  Hello | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li style=”display: none;”>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  menu.innerHTML = “<h1>Hello</h1>”;  </script> | (is nothing) |
| innerHTML  screen:  <h1>Hello</h1> | <body>  <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  <li style=”display: none;”>Project </li>  </ul>  </body> | <script>  let menu = document.getElementById(“menu”);  menu.textContent = “<h1>Hello</h1>”;  </script> | (is nothing) |
| after() Meth  screen:  ??? | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  Element.after(node);  menu.after(node1. Node2, … nodeN)  </script> | ??? |
| Append() Method  screen:  ??? | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  parentNode.append(newNodes);  </script> | ??? |
| prepend() Method  screen:  ??? | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  parentNode.prepend(newNodes); | ??? |
| insertAdjacentHTML() Method  Home  About  Contact  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  element.insertAdjacentHTML(positionName, text);  menu.insertAdjacentHTML(“beforeend”, “<li>contact</li>”);  </script> | <ul id=”menu”>  <li>Home</li>  <li>About</li>  <li>Contact</li>  <li>Blog</li>  </ul> |
| insertAdjacentHTML() Method  screen:  Contact  Home  About  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  element.insertAdjacentHTML(positionName, text);  menu.insertAdjacentHTML(“afterbegin”, “<li>contact</li>”);  </script> | <ul id=”menu”>  <li>Contact</li>  <li>Home</li>  <li>About</li>  <li>Blog</li>  </ul> |
| replaceChild() Method  screen:  Home  Contact  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  parentNode.replaceChild(newChild, oldChild);  menu.replaceChild(“contact”, “about”);  </script> | <ul id=”menu”>  <li>Home</li>  <li>Contact</li>  <li>Blog</li>  </ul> |
| cloneNode() Method  screen:  Home  About  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  let newNode = originalNode.cloneNode();  let newNode = .menu.cloneNode();  console.log(newNode);  </script> | <ul id=”menu”></ul> |
| cloneNode() Method  screen:  Home  About  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  let newNode = originalNode.cloneNode(true);  let newNode = .menu.cloneNode();  console.log(newNode); | <ul id=”menu”>  <li>Home</li>  <li>About</li>  <li>Blog</li>  </ul> |
| cloneNode() Method  screen:  Home  About  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  let newNode = originalNode.cloneNode(false);  let newNode = .menu.cloneNode();  console.log(newNode);  </script> | <ul id=”menu”></ul> |
| removeChild() Method  screen:  Home  About | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  parentNode.removeChild(childNode);  menu.removeChild(menu.lastElementChild);  </script> | <ul id=”menu”>  <li>Home</li>  <li>About</li>  </ul> |
| removeChild() Method  screen:  About  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  parentNode.removeChild(childNode);  menu.removeChild(menu.firstElementChild);  </script> | <ul id=”menu”>  <li>About</li>  <li>Blog</li>  </ul> |
| InsertBefore() Method  screen:  contact  Home  About  Blog | <ul id=”menu”>  <li>Home </li>  <li>About </li>  <li>Blog </li>  </ul> | <script>  let menu = document.getElementById(“menu”);  parentNode.insertbefore(newNode, existingNode);  menu.insertBefore(contact, menu.firstElemetChild);  </script> | <ul id=”menu”>  <li>Contact</li>  <li>Home</li>  <li>About</li>  <li>Blog</li>  </ul> |
| Attribute() methods  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.attributes)  </script> | NameNodeMap {  0: type,  1: id,  2: placeholder,  type: type, id: id, placeholder: placeholder,  length: 3} |
| getAtribute() Method  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.getAttribute(“type”))  </script> | text |
| getAtribute() Method  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.getAttribute(“placeholder”))  </script> | username |
| setAttribute() Method  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  inputBox.setAttribute(“name”, “value”));  inputBox.setAttribute(“class”, “user”));  console.log(inputBox);  </script> | <input type=”text” id=”username” placeholder=”username”    class=”user”> |
| hasAttribute() Method  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.hasAttribute(“class”));  </script> | False  (fiindca nu avem in codul html clasă) |
| hasAttribute() Method  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.hasAttribute(“id”));  </script> | True  (fiindca avem in codul html id) |
| removeAttribute() Method  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  inputBox.removeAttribute(“placeholder”);  console.log(inputBox); | <input type=”text” id=”username” |
| Manipulating Element’s Style: Style property  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.style);  </script> | cssStyleDeclaration { 0:  ‘background-color’, 1: ‘font-size’, accentColor: ‘ ‘ , additiveSymbols: ‘ ‘, alignContent: ‘ ‘, alignItems: ‘ ‘, alignSelf: ‘ ‘ ,…}  } |
| Manipulating Element’s Style: Style property  screen:   |  | | --- | | username | | username | | <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px”>  </body>  <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px”>  </body> | <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.style.backgroundColor);  </script>  <script>  let inputBox = document.getElementById(“username”);  console.log(inputBox.style.fontSize);  </script> | Red  20px |
| Manipulating Element’s Style: Style property  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px”>  </body> | <script>  let inputBox = document.getElementById(“username”);  inputBox.style.padding = “10px”;  console.log(inputBox);  </script> | <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font-size: 20px; padding: 10px;” > |
| CSS text  screen:   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px”>  </body> | <script>  let inputBox = document.getElementById(“username”);  inputBox.style.cssText = “width:200px”;  console.log(inputBox);  </script> | <input type=”text” id=”username” placeholder=”username” style=”width: 200px;” > |
| CSS text  Screen   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px”>  </body> | <script>  let inputBox = document.getElementById(“username”);  inputBox.style.cssText += “width:200px”;  console.log(inputBox);  </script> | <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font-size: 20px; width: 200px;> |
| CSS text  Screen   |  | | --- | | username | | <body>  <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font\*-size:20px” >;  </body> | <script>  let inputBox = document.getElementById(“username”);  inputBox.style.cssText += “width:200px height:100px”;  console.log(inputBox);  </script> | <input type=”text” id=”username” placeholder=”username” style=”background-color: red; font-size: 20px; width: 200px; height: 100px> |
| getComputedStyle() Method  Screen   |  | | --- | | username | | <title>GreatStack </title>  <style>  Input {  background-color: red;  font-size: 12px  }  </style>  <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  windows.getComputedStyle(element, pseudoElement)  console.log(windows.getComputedStyle(inputBox).fontSize);  </script> | 12px |
| getComputedStyle() Method  Screen   |  | | --- | | username | | <title>GreatStack </title>  <style>  Input {  background-color: red;  font-size: 12px  }  </style>  <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  windows.getComputedStyle(element, pseudoElement)  console.log(windows.getComputedStyle(inputBox).backgroundColor);  </script> | rgb(255, 0, 0) |
| getComputedStyle() Method  Screen   |  | | --- | | username | | <title>GreatStack </title>  <style>  Input {  background-color: red;  font-size: 12px  width: 300px;  }  </style>  <body>  <input type=”text” id=”username” placeholder=”username”>  </body> | <script>  let inputBox = document.getElementById(“username”);  windows.getComputedStyle(element, pseudoElement)  console.log(windows.getComputedStyle(inputBox).width);  </script> | 300px |
| className property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  console.log( title.className);  </script> | main message |
| className property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.className=”new”;  console.log(title)  </script> | <h1 id=”title” class=”new”> GreatStack</h1> |
| className property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.className += ” new”;  console.log(title)  </script> | <h1 id=”title” class=”man message new”> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  console.log(title.classList);  </script> | DOMToken(2) [‘main’, ‘message’, value: ‘main message’]  0: “main”  1: ”message”  length: 2  value: “main message”  [[Prototype]]: DOMTokenList |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.add(“new”);  console.log(title);  </script> | <h1 id=”title” class=”man message new”> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.add(“new”, “new2”);  console.log(title);  </script> | <h1 id=”title” class=”man message new new2”> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.remove(“message”);  console.log(title);  </script> | <h1 id=”title” class=”main”> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.remove(“message”, “main”);  console.log(title);  </script> | <h1 id=”title” class> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.replace(“message”, “msg”);  console.log(title);  </script> | <h1 id=”title” class=”main msg”> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  console.log(title.classList.contains(“message”));  </script> | true |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  console.log(title.classList.contains(“msg”));  </script> | false |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.toggle(“message”)  console.log(title);  </script> | <h1 id=”title” class=”main”> GreatStack</h1> |
| classList property  screen:  GreatStack | <body>  <h1 id=”title” class=”main message”> GreatStack </h1>  </body> | <script>  let title = document.getElementById(“title”);  title.classList.toggle(“new”);  console.log(title);  </script> | <h1 id=”title” class=”main message new”> GreatStack</h1> |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button”>Click Here</button>  </div>  </body> | (is nothing) | ( is nothing) |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button” onClick=”console.log(‘Button Clicked!’)”>Click Here</button>  </div>  </body> | (is nothing) | (on click button)  Button  Clicked! |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button” onClick=”displayMsg ”>Click Here</button>  </div>  </body> | <script>  Function displayMsg() {  console.log(“Button Clicked From Fn”);  }  </script> | (on click button)  Button  Clicked! |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button” onClick=”console.log(event);”>Click Here</button>  </div>  </body> | (is nothing) | (on click button)  PointerEvent {isTruste  d:true, pointerId: 1,  width: 1, height: 1, pressure: 0, …  } |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button” onClick=”console.log(event.type);”>Click Here</button>  </div>  </body> | (is nothing) | (on click button)  click |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button” onClick=”console.log(event.target);”>Click Here</button>  </div>  </body> | (is nothing) | <button tupe=”button”  onclick=”console.log(event,target);”>Click Here</button> |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button type=”button” onClick=”console.log(this);”>Click Here</button>  </div>  </body> | (is nothing) | <button tupe=”button”  onclick=”console.log(this);”>Click Here</button> |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn” type=”button” onClick=”console.log(this.id);”>Click Here</button>  </div>  </body> | (is nothing) | (on click button)  btn |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn” type=”button” onClick=”console.log(id);”>Click Here</button>  </div>  </body> | (is nothing) | (on click button)  btn |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn” type=”button” onClick=”console.log(type);”>Click Here</button>  </div>  </body> | (is nothing) | (on click button)  button |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body> | <script>  let btn = document.getElementById(“btn”)  btn.onclick = function() {  console.log(“button clicked”)  }  </script> | (on click button)  button  clicked |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body> | <script>  let btn = document.getElementById(“btn”)  btn.onclick = function() {  console.log(this.id);  }  </script> | (on click button)  btn |
| JavaScript Events  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body> | <script>  let btn = document.getElementById(“btn”)  btn.onclick = null;  </script> | (is nothing) |
| addEventListener() method  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body> | <script>  let btn = document.getElementById(“btn”)  btn.addEventListener( ‘click’, function(), {  console.log(“Button Clicked!”)  })  </script> | (on click button)  button  clicked! |
| addEventListener() method  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body> | <script>  let btn = document.getElementById(“btn”)  btn.addEventListener( ‘click’, function(event), {  console.log(event.type);  })  </script> | (on click button)  click |
| addEventListener() method  screen:   |  | | --- | | Click Here |   addEventListener() method  screen:   |  | | --- | | Click Here | | <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body>  <body>  <div id=”container”>  <button id=”btn”>Click Here</button>  </div>  </body> | <script>  let btn = document.getElementById(“btn”)  function displayMsg(){  console.log(“Button Clicked”);  }  btn.addEventListener( ‘click’, displayMsg);  </script>  <script>  let btn = document.getElementById(“btn”)  function displayMsg(){  console.log(“Button Clicked”);  }  btn.addEventListener( ‘click’, displayMsg);  btn.removeEventListener(‘click’, displayMsg);  </script> | (on click button)  Button  Clicked  (on click button) |
|  |  |  |  |
|  |  |  |  |