# Solutions-Exercises: DOM Manipulation

## 1.Subtraction

function subtract(){

let firstNumberElement = document.getElementById('firstNumber');

let secondNumberElement = document.getElementById('secondNumber');

let resultElement = document.getElementById('result');

let firstNumber = Number(firstNumberElement.value);

let secondNumber = Number(secondNumberElement.value);

let subtractionResult = firstNumber - secondNumber;

resultElement.textContent = subtractionResult;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Subtraction</title>

</head>

<body>

<div id="wrapper">

<input type="text" id="firstNumber" value="13.33" disabled>

<input type="text" id="secondNumber" value="22.18" disabled>

<div id="result"></div>

</div>

<script src="01Subtraction.js"></script>

<script>

window.onload = function () {

subtract();

function subtract() {

let firstNumberElement = document.getElementById('firstNumber');

let secondNumberElement = document.getElementById('secondNumber');

let resultElement = document.getElementById('result');

let firstNumber = Number(firstNumberElement.value);

let secondNumber = Number(secondNumberElement.value);

let subtractionResult = firstNumber - secondNumber;

resultElement.textContent = subtractionResult;

}

}

</script>

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Subtraction</title> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div id="wrapper"> |
|  | <input type="text" id="firstNumber" value="13.33" disabled> |
|  | <input type="text" id="secondNumber" value="22.18" disabled> |
|  |  |
|  | <div id="result"></div> |
|  | </div> |
|  | <script src="subtract.js"></script> |
|  | <script> |
|  | window.onload = function () { |
|  | subtract(); |
|  |  |
|  | function subtract() { |
|  | document.getElementById("result").textContent = Number(document.getElementById("firstNumber").value) - |
|  | Number(document.getElementById("secondNumber").value); |
|  | } |
|  | } |
|  | </script> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function subtract() { |
|  | let num1 = Number(document.getElementById('firstNumber').value); |
|  | let num2 = Number(document.getElementById('secondNumber').value); |
|  | document.getElementById('result').textContent = num1 - num2; |
|  | } |

## 2.Fill Dropdown

function addItem(){

let newItemTextElement = document.getElementById('newItemText');

let newItemValueElement = document.getElementById('newItemValue');

let selectMenuElement = document.getElementById('menu');

let optionElement = document.createElement('option');

optionElement.textContent = newItemTextElement.value;

optionElement.value = newItemValueElement.value;

selectMenuElement.appendChild(optionElement);

newItemTextElement.value = '';

newItemValueElement.value = '';

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Fill Dropdown</title>

<script src="02FillDropdown.js"></script>

</head>

<body>

<h1>Dropdown Menu</h1>

<div>

<select id="menu"></select>

</div>

<label for="newItemText">Text: </label><input type="text" id="newItemText" />

<label for="newItemValue">Value: </label><input type="text" id="newItemValue" />

<input type="button" value="Add" onclick="addItem()">

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Fill Dropdown</title> |
|  | <script src="dropdown.js"></script> |
|  | </head> |
|  |  |
|  | <body> |
|  | <h1>Dropdown Menu</h1> |
|  | <div> |
|  | <select id="menu"></select> |
|  | </div> |
|  | <label for="newItemText">Text: </label> |
|  | <input type="text" id="newItemText" /> |
|  | <label for="newItemValue">Value: </label> |
|  | <input type="text" id="newItemValue" /> |
|  | <input type="button" value="Add" onclick="addItem()"> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function addItem() { |
|  | let opt = document.createElement("option"); |
|  | opt.textContent = document.getElementById("newItemText").value; |
|  | opt.value = document.getElementById("newItemValue").value; |
|  | document.getElementById("menu").appendChild(opt); |
|  | document.getElementById("newItemText").value = ""; |
|  | document.getElementById("newItemValue").value = ""; |
|  | } |

|  |
| --- |
| function addItem() { |
|  | let inputText = document.getElementById('newItemText'); |
|  | let inputValue = document.getElementById('newItemValue'); |
|  |  |
|  | let textContent = inputText.value; |
|  | inputText.value = ''; |
|  |  |
|  | let valueContent = inputValue.value; |
|  | inputValue.value = ''; |
|  |  |
|  | let option = document.createElement('option'); |
|  | option.value = valueContent; |
|  | option.textContent = textContent; |
|  |  |
|  | let menu = document.getElementById('menu'); |
|  | menu.appendChild(option); |
|  | } |

## 3.Accordion

function toggle() {

let spanButton = document.getElementsByClassName('button')[0];

let divExtra = document.getElementById('extra');

if (spanButton.textContent === 'More'){

//if (spanButton.textContent === 'More' || spanButton.style.display === 'none') {

spanButton.textContent = 'Less';

divExtra.style.display = 'block';

}

else {

spanButton.textContent = 'More';

divExtra.style.display = 'none';

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Accordion</title>

<style>

#accordion {

border: 1px solid black;

display: inline-block;

width: 400px;

}

#accordion p {

margin: 1em;

}

.button {

float: right;

background: #5555ff;

padding: 0.1em 1em 0.1em 1em;

color: white;

cursor: pointer;

}

#extra {

display: none;

}

.head {

background: #ccccff;

padding: 1em;

}

</style>

</head>

<body>

<div id="accordion">

<div class="head">DOM Manipulations Exercise <span class="button" onclick="toggle()">More</span></div>

<div id="extra">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et

dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip

ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu

fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia

deserunt mollit anim id est laborum.</p>

</div>

</div>

<script>

function toggle() {

let spanButton = document.getElementsByClassName('button')[0];

let divExtra = document.getElementById('extra');

if (spanButton.textContent === 'More') {

//if (spanButton.textContent === 'More' || spanButton.style.display === 'none') {

spanButton.textContent = 'Less';

divExtra.style.display = 'block';

}

else {

spanButton.textContent = 'More';

divExtra.style.display = 'none';

}

}

</script>

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Accordion</title> |
|  | <style> |
|  | #accordion { |
|  | border: 1px solid black; |
|  | display: inline-block; |
|  | width: 400px; |
|  | } |
|  |  |
|  | #accordion p { |
|  | margin: 1em; |
|  | } |
|  |  |
|  | .button { |
|  | float: right; |
|  | background: #5555ff; |
|  | padding: 0.1em 1em 0.1em 1em; |
|  | color: white; |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | #extra { |
|  | display: none; |
|  | } |
|  |  |
|  | .head { |
|  | background: #ccccff; |
|  | padding: 1em; |
|  | } |
|  | </style> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div id="accordion"> |
|  | <div class="head">DOM Manipulations Exercise |
|  | <span class="button" onclick="toggle()">More</span> |
|  | </div> |
|  | <div id="extra"> |
|  | <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore |
|  | magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo |
|  | consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. |
|  | Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> |
|  | </div> |
|  | </div> |
|  | <script> |
|  | function toggle() { |
|  | let btn = document.getElementsByClassName("button")[0]; |
|  | let extra = document.getElementById("extra"); |
|  | (extra.style.display === "none" || extra.style.display === "") ? extra.style.display = "block": extra.style |
|  | .display = "none"; |
|  | btn.textContent = (btn.textContent == "More" ? "Less" : "More"); |
|  | } |
|  | </script> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function toggle() { |
|  | let button = document.querySelector('span.button'); |
|  | let currentState = button.textContent; |
|  | let targetDiv = document.getElementById('extra'); |
|  |  |
|  | if (currentState === 'More') { |
|  | more(); |
|  | } else { |
|  | less(); |
|  | } |
|  |  |
|  | function more() { |
|  | button.textContent = 'Less'; |
|  | targetDiv.style.display = 'block'; |
|  | } |
|  |  |
|  | function less() { |
|  | button.textContent = 'More'; |
|  | targetDiv.style.display = 'none'; |
|  | } |
|  | } |

## 4.Sections

function create(sentences){

let divContent = document.getElementById('content');

for(let sentence of sentences){

let p = document.createElement('p');

p.textContent = sentence;

p.style.display = 'none';

let div = document.createElement('div');

div.appendChild(p);

div.addEventListener('click', showParagraph);

divContent.appendChild(div);

}

function showParagraph(event){

//let p = event.target.firstChild;

let p = this.firstChild;

//let p = this.children[0];

p.style.display = 'block';

//p.style.display = 'inline';

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Sections</title>

<style>

#content {

width: 1000px;

}

#content div {

float: left;

width: 300px;

height: 300px;

margin: 2em;

background: #5555ff;

text-align: center;

}

#content div p {

color: white;

margin: 6em 3em 6em 3em;

}

</style>

</head>

<div id="content">

</div>

<body onload="create(['Section 1', 'Section 2', 'Section 3', 'Section 4']);">

<script>

function create(sentences) {

let divContent = document.getElementById('content');

for (let sentence of sentences) {

let p = document.createElement('p');

p.textContent = sentence;

p.style.display = 'none';

let div = document.createElement('div');

div.appendChild(p);

div.addEventListener('click', showParagraph);

divContent.appendChild(div);

}

function showParagraph(event) {

//let p = event.target.firstChild;

let p = this.firstChild;

//let p = this.children[0];

p.style.display = 'block';

//p.style.display = 'inline';

}

}

</script>

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Sections</title> |
|  | <style> |
|  | #content { |
|  | width: 1000px; |
|  | } |
|  |  |
|  | #content div { |
|  | float: left; |
|  | width: 300px; |
|  | height: 300px; |
|  | margin: 2em; |
|  | background: #5555ff; |
|  | text-align: center; |
|  | } |
|  |  |
|  | #content div p { |
|  | color: white; |
|  | margin: 6em 3em 6em 3em; |
|  | } |
|  | </style> |
|  | </head> |
|  | <div id="content"> |
|  | </div> |
|  |  |
|  | <body onload="create(['Section 1', 'Section 2', 'Section 3', 'Section 4']);"> |
|  | <script> |
|  | function create(sentences) { |
|  | let container = document.getElementById("content"); |
|  | sentences.forEach(sen => { |
|  | let divEle = document.createElement("div"); |
|  | let para = document.createElement("p"); |
|  | para.textContent = sen; |
|  | para.style.display = "none" |
|  | divEle.appendChild(para); |
|  | divEle.addEventListener("click", show); |
|  | container.appendChild(divEle); |
|  | }); |
|  |  |
|  | function show() { |
|  | this.children[0].style.display = "block"; |
|  | console.log(this.children[0]); |
|  | } |
|  |  |
|  | } |
|  | </script> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function create(sentences) { |
|  | let container = document.getElementById('content'); |
|  |  |
|  | for (const line of sentences) { |
|  | let paragraph = document.createElement('p'); |
|  | paragraph.textContent = line; |
|  | paragraph.style.display = 'none'; |
|  |  |
|  | let currentDiv = document.createElement('div'); |
|  | currentDiv.appendChild(paragraph); |
|  | currentDiv.addEventListener('click', displayChildren); |
|  |  |
|  | container.appendChild(currentDiv); |
|  | } |
|  |  |
|  | function displayChildren() { |
|  | let currentParagraph = this.firstChild; |
|  | currentParagraph.style.display = 'inline'; |
|  | } |
|  | } |

## 5.Notification

function notify(message) {

let divNotification = document.getElementById('notification');

divNotification.textContent = message;

divNotification.style.display = 'block';

setTimeout(hideNotification, 2000);

function hideNotification() {

divNotification.style.display = 'none';

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Notification</title>

<style>

body {

width: 600px;

text-align: center;

}

.header {

background-color: #5555ff;

color: white;

position: relative;

left: 0;

top: 0;

padding: 0.5em;

}

#container {

position: relative;

}

.post {

margin: 48px;

text-align: left;

}

#notification {

float: right;

background: #119911;

color: #ffffff;

padding: 0.5em 2em 0.5em 2em;

margin: 1em;

display: none;

position: absolute;

top: 0;

right: 0;

}

</style>

</head>

<body>

<div id="container">

<header class="header">

<h1>Welcome to our site</h1>

</header>

<div id="content">

<article class="post">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore

et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

aliquip ex ea commodo consequat.</p>

<p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla

pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit

anim id est laborum.</p>

</article>

<button onclick="notify('Something happened!')">Get notified</button>

</div>

<div id="notification"></div>

</div>

<script>

function notify(message) {

let divNotification = document.getElementById('notification');

divNotification.textContent = message;

divNotification.style.display = 'block';

setTimeout(hideNotification, 2000);

function hideNotification() {

divNotification.style.display = 'none';

}

}

</script>

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Notification</title> |
|  | <style> |
|  | body { |
|  | width: 600px; |
|  | text-align: center; |
|  | } |
|  |  |
|  | .header { |
|  | background-color: #5555ff; |
|  | color: white; |
|  | position: relative; |
|  | left: 0; |
|  | top: 0; |
|  | padding: 0.5em; |
|  | } |
|  |  |
|  | #container { |
|  | position: relative; |
|  | } |
|  |  |
|  | .post { |
|  | margin: 48px; |
|  | text-align: left; |
|  | } |
|  |  |
|  | #notification { |
|  | float: right; |
|  | background: #119911; |
|  | color: #ffffff; |
|  | padding: 0.5em 2em 0.5em 2em; |
|  | margin: 1em; |
|  | display: none; |
|  | position: absolute; |
|  | top: 0; |
|  | right: 0; |
|  | } |
|  | </style> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div id="container"> |
|  | <header class="header"> |
|  | <h1>Welcome to our site</h1> |
|  | </header> |
|  | <div id="content"> |
|  | <article class="post"> |
|  | <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore |
|  | magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea |
|  | commodo consequat.</p> |
|  | <p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur |
|  | sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> |
|  | </article> |
|  | <button onclick="notify('Something happened!')">Get notified</button> |
|  | </div> |
|  | <div id="notification"></div> |
|  | </div> |
|  | <script> |
|  | function notify(message) { |
|  | let notifyEle = document.getElementById("notification"); |
|  | let textNode = document.createTextNode(message); |
|  | notifyEle.appendChild(textNode); |
|  | notifyEle.style.display = "block"; |
|  | setTimeout(() => { |
|  | notifyEle.children = []; |
|  | notifyEle.style.display = "none"; |
|  | }, (2000)); |
|  | } |
|  | </script> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function notify(message) { |
|  | let notificationBox = document.getElementById('notification'); |
|  | notificationBox.textContent = message; |
|  | notificationBox.style.display = 'block'; |
|  | setTimeout(hideNotificationBox, 2000); |
|  |  |
|  | function hideNotificationBox() { |
|  | notificationBox.style.display = 'none'; |
|  | } |
|  | } |

## 6.Time Converter

function attachEventsListeners(){

let daysBtn = document.getElementById('daysBtn');

let hoursBtn = document.getElementById('hoursBtn');

let minutesBtn = document.getElementById('minutesBtn');

let secondsBtn = document.getElementById('secondsBtn');

daysBtn.addEventListener('click', convertTime);

hoursBtn.addEventListener('click', convertTime);

minutesBtn.addEventListener('click', convertTime);

secondsBtn.addEventListener('click', convertTime);

function convertTime(event){

let daysElement = document.getElementById('days');

let hoursElement = document.getElementById('hours');

let minutesElement = document.getElementById('minutes');

let secondsElement = document.getElementById('seconds');

let timeInSeconds = 0;

switch(event.target.id){

//switch(this.id){

case 'daysBtn':

timeInSeconds = Number(daysElement.value) \* 86400;

break;

case 'hoursBtn':

timeInSeconds = Number(hoursElement.value) \* 3600;

break;

case 'minutesBtn':

timeInSeconds = Number(minutesElement.value) \* 60;

break;

case 'secondsBtn':

timeInSeconds = Number(secondsElement.value);

break;

}

daysElement.value = timeInSeconds / 86400;

hoursElement.value = timeInSeconds / 3600;

minutesElement.value = timeInSeconds / 60;

secondsElement.value = timeInSeconds

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Time Converter</title>

<script src="06TimeConverter.js"></script>

<style>

label,

input {

display: inline-block;

width: 5em;

}

label {

text-align: right;

}

</style>

</head>

<body onload="attachEventsListeners()">

<h1>Time Converter</h1>

<div>

<label for="days">Days: </label>

<input type="text" id="days">

<input id="daysBtn" type="button" value="Convert">

</div>

<div>

<label for="hours">Hours: </label>

<input type="text" id="hours">

<input id="hoursBtn" type="button" value="Convert">

</div>

<div>

<label for="minutes">Minutes: </label>

<input type="text" id="minutes">

<input id="minutesBtn" type="button" value="Convert">

</div>

<div>

<label for="seconds">Seconds: </label>

<input type="text" id="seconds">

<input id="secondsBtn" type="button" value="Convert">

</div>

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Time Converter</title> |
|  | <script src="timeConverter.js"></script> |
|  | <style> |
|  | label, |
|  | input { |
|  | display: inline-block; |
|  | width: 5em; |
|  | } |
|  |  |
|  | label { |
|  | text-align: right; |
|  | } |
|  | </style> |
|  | </head> |
|  |  |
|  | <body onload="attachEventsListeners()"> |
|  | <h1>Time Converter</h1> |
|  | <div> |
|  | <label for="days">Days: </label> |
|  | <input type="text" id="days"> |
|  | <input id="daysBtn" type="button" value="Convert"> |
|  | </div> |
|  | <div> |
|  | <label for="hours">Hours: </label> |
|  | <input type="text" id="hours"> |
|  | <input id="hoursBtn" type="button" value="Convert"> |
|  | </div> |
|  | <div> |
|  | <label for="minutes">Minutes: </label> |
|  | <input type="text" id="minutes"> |
|  | <input id="minutesBtn" type="button" value="Convert"> |
|  | </div> |
|  | <div> |
|  | <label for="seconds">Seconds: </label> |
|  | <input type="text" id="seconds"> |
|  | <input id="secondsBtn" type="button" value="Convert"> |
|  | </div> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function attachEventsListeners() { |
|  | let buttons = document.querySelectorAll("input[id$='Btn']"); |
|  | for (let i = 0; i < buttons.length; i++) { |
|  | const element = buttons[i]; |
|  | element.addEventListener("click", convertTime); |
|  | } |
|  |  |
|  | function convertTime() { |
|  | let secondsBase = 0; |
|  | switch (this.id) { |
|  | case "daysBtn": |
|  | secondsBase = Number(document.getElementById("days").value) \* 86400; |
|  | break; |
|  |  |
|  | case "hoursBtn": |
|  | secondsBase = Number(document.getElementById("hours").value) \* 3600; |
|  | break; |
|  |  |
|  | case "minutesBtn": |
|  | secondsBase = Number(document.getElementById("minutes").value) \* 60; |
|  | break; |
|  |  |
|  | case "secondsBtn": |
|  | secondsBase = Number(document.getElementById("seconds").value); |
|  | break; |
|  | } |
|  | document.getElementById("days").value = secondsBase / 86400; |
|  | document.getElementById("hours").value = secondsBase / 3600; |
|  | document.getElementById("minutes").value = secondsBase / 60; |
|  | document.getElementById("seconds").value = secondsBase; |
|  | } |
|  | } |

|  |
| --- |
| function attachEventsListeners() { |
|  | let divs = document.getElementsByTagName('div'); |
|  |  |
|  | // Define functions map |
|  | let functions = { |
|  | days: (nummericValue) => daysBtn(nummericValue), |
|  | hours: (nummericValue) => hoursBtn(nummericValue), |
|  | minutes: (nummericValue) => minutesBtn(nummericValue), |
|  | seconds: (nummericValue) => secondsBtn(nummericValue) |
|  | } |
|  |  |
|  | let getInputBox = parent => parent.getElementsByTagName('input')[0]; |
|  |  |
|  | // Input references |
|  | let inputBoxes = {}; |
|  |  |
|  | for (const element of divs) { |
|  | // Attach Events |
|  | element.lastElementChild.addEventListener('click', fillTheRest); |
|  |  |
|  | // Extract input references |
|  | let currentInput = getInputBox(element) |
|  | inputBoxes[currentInput.id] = currentInput; |
|  | } |
|  |  |
|  | function fillTheRest() { |
|  | let currentInput = getInputBox(this.parentElement); |
|  | let rawValue = currentInput.value.trim(); |
|  | if (rawValue === '') { |
|  | return; |
|  | } |
|  |  |
|  | let nummericValue = Number(rawValue); |
|  | if (isNaN(nummericValue)) { |
|  | return; |
|  | } |
|  |  |
|  | functions[currentInput.id](nummericValue); |
|  | } |
|  |  |
|  | function daysBtn(nummericValue) { |
|  | inputBoxes.hours.value = nummericValue \* 24; |
|  | inputBoxes.minutes.value = nummericValue \* 24 \* 60; |
|  | inputBoxes.seconds.value = nummericValue \* 24 \* 60 \* 60; |
|  | } |
|  |  |
|  | function hoursBtn(nummericValue) { |
|  | inputBoxes.days.value = nummericValue / 24; |
|  | inputBoxes.minutes.value = nummericValue \* 60; |
|  | inputBoxes.seconds.value = nummericValue \* 60 \* 60; |
|  | } |
|  |  |
|  | function minutesBtn(nummericValue) { |
|  | inputBoxes.days.value = nummericValue / 24 / 60; |
|  | inputBoxes.hours.value = nummericValue / 60; |
|  | inputBoxes.seconds.value = nummericValue \* 60; |
|  | } |
|  |  |
|  | function secondsBtn(nummericValue) { |
|  | inputBoxes.days.value = nummericValue / 24 / 60 / 60; |
|  | inputBoxes.hours.value = nummericValue / 60 / 60; |
|  | inputBoxes.minutes.value = nummericValue / 60; |
|  | } |
|  | } |

## 7.\* Distance Converter

function attachEventsListeners(){

let convertButton = document.getElementById('convert');

convertButton.addEventListener('click', convertDistance);

function convertDistance(){

let inputDistanceElement = document.getElementById('inputDistance');

let inputUnitsSelectElement = document.getElementById('inputUnits');

let outputDistanceElement = document.getElementById('outputDistance');

let outputUnitsSelectElement = document.getElementById('outputUnits');

let distanceInMeters = 0;

let inputDistance = Number(inputDistanceElement.value);

switch(inputUnitsSelectElement.value){

case 'km': distanceInMeters = inputDistance \* 1000; break;

case 'm': distanceInMeters = inputDistance; break;

case 'cm': distanceInMeters = inputDistance \* 0.01; break;

case 'mm': distanceInMeters = inputDistance \* 0.001; break;

case 'mi': distanceInMeters = inputDistance \* 1609.34; break;

case 'yrd': distanceInMeters = inputDistance \* 0.9144; break;

case 'ft': distanceInMeters = inputDistance \* 0.3048; break;

case 'in': distanceInMeters = inputDistance \* 0.0254; break;

}

let outputDistance = 0;

switch(outputUnitsSelectElement.value){

case 'km': outputDistance = distanceInMeters / 1000; break;

case 'm': outputDistance = distanceInMeters; break;

case 'cm': outputDistance = distanceInMeters / 0.01; break;

case 'mm': outputDistance = distanceInMeters / 0.001; break;

case 'mi': outputDistance = distanceInMeters / 1609.34; break;

case 'yrd': outputDistance = distanceInMeters / 0.9144; break;

case 'ft': outputDistance = distanceInMeters / 0.3048; break;

case 'in': outputDistance = distanceInMeters / 0.0254; break;

}

outputDistanceElement.value = outputDistance;

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Distance Converter</title>

<script src="07DistanceConverter.js"></script>

<style>

label, input {

display: inline-block;

width: 5em;

}

label {

text-align: right;

}

</style>

</head>

<body onload="attachEventsListeners()">

<h1>Distance Converter</h1>

<div>

<label for="inputDistance">From:</label>

<input type="text" id="inputDistance">

<select id="inputUnits">

<option value="km">Kilometers</option>

<option value="m">Meters</option>

<option value="cm">Centimeters</option>

<option value="mm">Millimeters</option>

<option value="mi">Miles</option>

<option value="yrd">Yards</option>

<option value="ft">Feet</option>

<option value="in">Inches</option>

</select>

<input type="button" id="convert" value="Convert">

</div>

<div>

<label for="outputDistance">To:</label>

<input type="text" id="outputDistance" disabled="disabled">

<select id="outputUnits">

<option value="km">Kilometers</option>

<option value="m">Meters</option>

<option value="cm">Centimeters</option>

<option value="mm">Millimeters</option>

<option value="mi">Miles</option>

<option value="yrd">Yards</option>

<option value="ft">Feet</option>

<option value="in">Inches</option>

</select>

</div>

</body>

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Distance Converter</title> |
|  | <script src="distanceConverter.js"></script> |
|  | <style> |
|  | label, |
|  | input { |
|  | display: inline-block; |
|  | width: 5em; |
|  | } |
|  |  |
|  | label { |
|  | text-align: right; |
|  | } |
|  | </style> |
|  | </head> |
|  |  |
|  | <body onload="attachEventsListeners()"> |
|  | <h1>Distance Converter</h1> |
|  | <div> |
|  | <label for="inputDistance">From:</label> |
|  | <input type="text" id="inputDistance"> |
|  | <select id="inputUnits"> |
|  | <option value="km">Kilometers</option> |
|  | <option value="m">Meters</option> |
|  | <option value="cm">Centimeters</option> |
|  | <option value="mm">Millimeters</option> |
|  | <option value="mi">Miles</option> |
|  | <option value="yrd">Yards</option> |
|  | <option value="ft">Feet</option> |
|  | <option value="in">Inches</option> |
|  | </select> |
|  | <input type="button" id="convert" value="Convert"> |
|  | </div> |
|  | <div> |
|  | <label for="outputDistance">To:</label> |
|  | <input type="text" id="outputDistance" disabled="disabled"> |
|  | <select id="outputUnits"> |
|  | <option value="km">Kilometers</option> |
|  | <option value="m">Meters</option> |
|  | <option value="cm">Centimeters</option> |
|  | <option value="mm">Millimeters</option> |
|  | <option value="mi">Miles</option> |
|  | <option value="yrd">Yards</option> |
|  | <option value="ft">Feet</option> |
|  | <option value="in">Inches</option> |
|  | </select> |
|  | </div> |
|  | </body> |
|  |  |
|  | </html> |

|  |
| --- |
| function attachEventsListeners() { |
|  | document.getElementById("convert").addEventListener("click", convertDistance); |
|  |  |
|  | function convertDistance() { |
|  | let baseValue = Number(document.getElementById("inputDistance").value); |
|  | let selectedFromVal = document.getElementById("inputUnits").value; |
|  | let selectedToVal = document.getElementById("outputUnits").value; |
|  | if (selectedFromVal === selectedToVal) { |
|  | document.getElementById("outputDistance").value = baseValue; |
|  | } else { |
|  | let baseInMeters = 0; |
|  | switch (selectedFromVal) { |
|  | case "km": |
|  | baseInMeters = baseValue \* 1000; |
|  | break; |
|  | case "m": |
|  | baseInMeters = baseValue \* 1; |
|  | break; |
|  | case "cm": |
|  | baseInMeters = baseValue \* 0.01; |
|  | break; |
|  | case "mm": |
|  | baseInMeters = baseValue \* 0.001; |
|  | break; |
|  | case "mi": |
|  | baseInMeters = baseValue \* 1609.34; |
|  | break; |
|  | case "yrd": |
|  | baseInMeters = baseValue \* 0.9144; |
|  | break; |
|  | case "ft": |
|  | baseInMeters = baseValue \* 0.3048; |
|  | break; |
|  | case "in": |
|  | baseInMeters = baseValue \* 0.0254; |
|  | break; |
|  | } |
|  | let output = document.getElementById("outputDistance"); |
|  | switch (selectedToVal) { |
|  | case "km": |
|  | output.value = baseInMeters / 1000; |
|  | break; |
|  | case "m": |
|  | output.value = baseInMeters / 1; |
|  | break; |
|  | case "cm": |
|  | output.value = baseInMeters / 0.01; |
|  | break; |
|  | case "mm": |
|  | output.value = baseInMeters / 0.001; |
|  | break; |
|  | case "mi": |
|  | output.value = baseInMeters / 1609.34; |
|  | break; |
|  | case "yrd": |
|  | output.value = baseInMeters / 0.9144; |
|  | break; |
|  | case "ft": |
|  | output.value = baseInMeters / 0.3048; |
|  | break; |
|  | case "in": |
|  | output.value = baseInMeters / 0.0254; |
|  | break; |
|  | } |
|  | } |
|  | } |
|  | } |

|  |
| --- |
| function attachEventsListeners() { |
|  | let button = document.getElementById('convert'); |
|  | button.addEventListener('click', convert); |
|  |  |
|  | // Get working references |
|  | let inputBox = document.getElementById('inputDistance'); |
|  | let outputBox = document.getElementById('outputDistance'); |
|  |  |
|  | let inputUnits = document.getElementById('inputUnits'); |
|  | let outputUnits = document.getElementById('outputUnits'); |
|  |  |
|  | // Define Conversions |
|  | let conversions = { |
|  | km: { |
|  | km: n => n, |
|  | m: n => n \* 1000, |
|  | cm: n => n \* 1000 \* 100, |
|  | mm: n => n \* 1000 \* 100 \* 10, |
|  | mi: n => n \* 1000 / 1609.34, |
|  | yrd: n => n \* 1000 / 0.9144, |
|  | ft: n => n \* 1000 / 0.3048, |
|  | in: n => n \* 1000 / 0.0254 |
|  | }, |
|  | m: { |
|  | km: n => n / 1000, |
|  | m: n => n, |
|  | cm: n => n \* 100, |
|  | mm: n => n \* 1000, |
|  | mi: n => n / 1609.34, |
|  | yrd: n => n / 0.9144, |
|  | ft: n => n / 0.3048, |
|  | in: n => n / 0.0254 |
|  | }, |
|  | cm: { |
|  | km: n => n / 100 / 1000, |
|  | m: n => n / 100, |
|  | cm: n => n, |
|  | mm: n => n \* 10, |
|  | mi: n => n / 100 / 1609.34, |
|  | yrd: n => n / 100 / 0.9144, |
|  | ft: n => n / 100 / 0.3048, |
|  | in: n => n / 100 / 0.0254 |
|  | }, |
|  | mm: { |
|  | km: n => n \* 0.001 / 1000, |
|  | m: n => n \* 0.001, |
|  | cm: n => n / 10, |
|  | mm: n => n, |
|  | mi: n => n \* 0.001 / 1609.34, |
|  | yrd: n => n \* 0.001 / 0.9144, |
|  | ft: n => n \* 0.001 / 0.3048, |
|  | in: n => n \* 0.001 / 0.0254 |
|  | }, |
|  | mi: { |
|  | km: n => n \* 1609.34 / 1000, |
|  | m: n => n \* 1609.34, |
|  | cm: n => n \* 1609.34 \* 100, |
|  | mm: n => n \* 1609.34 \* 1000, |
|  | mi: n => n, |
|  | yrd: n => n \* 1609.34 / 0.9144, |
|  | ft: n => n \* 1609.34 / 0.3048, |
|  | in: n => n \* 1609.34 / 0.0254 |
|  | }, |
|  | yrd: { |
|  | km: n => n \* 0.9144 / 1000, |
|  | m: n => n \* 0.9144, |
|  | cm: n => n \* 0.9144 \* 100, |
|  | mm: n => n \* 0.9144 \* 1000, |
|  | mi: n => n \* 0.9144 / 1609.34, |
|  | yrd: n => n, |
|  | ft: n => n \* 0.9144 / 0.3048, |
|  | in: n => n \* 0.9144 / 0.0254 |
|  | }, |
|  | ft: { |
|  | km: n => n \* 0.3048 / 1000, |
|  | m: n => n \* 0.3048, |
|  | cm: n => n \* 0.3048 \* 100, |
|  | mm: n => n \* 0.3048 \* 1000, |
|  | mi: n => n \* 0.3048 / 1609.34, |
|  | yrd: n => n \* 0.3048 / 0.9144, |
|  | ft: n => n, |
|  | in: n => n \* 0.3048 / 0.0254 |
|  | }, |
|  | in: { |
|  | km: n => n \* 0.0254 / 1000, |
|  | m: n => n \* 0.0254, |
|  | cm: n => n \* 0.0254 \* 100, |
|  | mm: n => n \* 0.0254 \* 1000, |
|  | mi: n => n \* 0.0254 / 1609.34, |
|  | yrd: n => n \* 0.0254 / 0.9144, |
|  | ft: n => n \* 0.0254 / 0.3048, |
|  | in: n => n |
|  | } |
|  | } |
|  |  |
|  | function convert() { |
|  | // Get input as number |
|  | let rawValue = inputBox.value.trim(); |
|  | if (rawValue === '') { |
|  | return; |
|  | } |
|  |  |
|  | let value = Number(rawValue); |
|  | if (isNaN(value)) { |
|  | return; |
|  | } |
|  |  |
|  | let fromUnit = conversions[inputUnits.value]; |
|  | if (!fromUnit) { |
|  | return; |
|  | } |
|  |  |
|  | let convertingFunc = fromUnit[outputUnits.value]; |
|  | if (!convertingFunc) { |
|  | return; |
|  | } |
|  |  |
|  | let result = convertingFunc(value); |
|  | outputBox.value = result; |
|  | } |
|  | } |

|  |
| --- |
| function attachEventsListeners() { |
|  | let button = document.getElementById('convert'); |
|  | button.addEventListener('click', convert); |
|  |  |
|  | // Get working references |
|  | let inputBox = document.getElementById('inputDistance'); |
|  | let outputBox = document.getElementById('outputDistance'); |
|  |  |
|  | let inputUnits = document.getElementById('inputUnits'); |
|  | let outputUnits = document.getElementById('outputUnits'); |
|  |  |
|  | // Define Conversions |
|  | let meterBasedRates = { |
|  | km: 1000, |
|  | m: 1, |
|  | cm: 0.01, |
|  | mm: 0.001, |
|  | mi: 1609.34, |
|  | yrd: 0.9144, |
|  | ft: 0.3048, |
|  | in: 0.0254 |
|  | } |
|  |  |
|  | function convert() { |
|  | // Get input as number |
|  | let rawValue = inputBox.value.trim(); |
|  | if (rawValue === '') { |
|  | return; |
|  | } |
|  |  |
|  | let value = Number(rawValue); |
|  | if (isNaN(value)) { |
|  | return; |
|  | } |
|  |  |
|  | let fromRate = meterBasedRates[inputUnits.value]; |
|  | let targetRate = meterBasedRates[outputUnits.value] |
|  | if (!fromRate || !targetRate) { |
|  | return; |
|  | } |
|  |  |
|  | let result = value \* fromRate / targetRate; |
|  | outputBox.value = result; |
|  | } |
|  | } |