**ДЗ – закічнити приклади в кінці документу!**

================ ТЕОРІЯ ====================

**Вибірка елементів сторінки**

|  |  |
| --- | --- |
|  |  |
| [getElementById()](https://puzzleweb.ru/javascript/document_getelementbyid.php) | Повертає елемент з вказаним ідентифікатором |
| [getElementsByClassName()](https://puzzleweb.ru/javascript/document_getelementsbyclassname.php) | Повертає колекцію елементів, які відповідають вказаному класу |
| getElementsByName() | Повертає колекцію елементів, які мають атрибут name з вказаним значенням.  Метод document.getElementsByName() относится к объекту HTMLDocument, а не к объекту Document, поэтому он доступен только в HTML-документах и не доступен в XML-документах. |
| [getElementsByTagName()](https://puzzleweb.ru/javascript/document_getelementsbytagname.php) | Повертає колекцію елементів вказаного типу |
| [querySelector()](https://puzzleweb.ru/javascript/document_queryselector.php) | Повертає перший елемент, який відповідає вказаному селектору |
| [querySelectorAll()](https://puzzleweb.ru/javascript/document_queryselectorall.php) | Повертає масив елементів, які відповідають вказаному селектору |

Доступ до вмісту

**Об’єкт Element**

Об’єкт Element є інтерфейсом, який надає доступ до властивостей і методів елементав

### Методы

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| [addEventListener()](https://puzzleweb.ru/javascript/element_addeventlistener.php) | Позволяет установить обработчик события для элемента. |
| [appendChild()](https://puzzleweb.ru/javascript/element_appendchild.php) | Добавляет дочерний узел в конец родительского элемента. |
| getAttribute() | Извлекает значение указанного атрибута. |
| [getElementsByClassName()](https://puzzleweb.ru/javascript/document_getelementsbyclassname.php) | Возвращает коллекцию элементов, соответствующих указанному классу. |
| [getElementsByTagName()](https://puzzleweb.ru/javascript/document_getelementsbytagname.php) | Возвращает массив элементов, указанного типа. |
| hasAttribute() | Возвращает значение true, если элемент имеет указанный атрибут, или false в обратном случае. |
| [insertBefore()](https://puzzleweb.ru/javascript/element_insertbefore.php) | Вставляет элемент перед указанным элементом. |
| removeAttribute() | Удаляет указанный атрибут из текущего элемента. |
| [removeChild()](https://puzzleweb.ru/javascript/element_removechild.php) | Удаляет указанный дочерний узел из текущего родительского элемента. |
| [removeEventListener()](https://puzzleweb.ru/javascript/element_removeeventlistener.php) | Удаляет обработчик события, который был установлен с помощью метода addEventListener(). |
| [setAttribute()](https://puzzleweb.ru/javascript/element_setattribute.php) | Добавляет к элементу атрибут и устанавливает для него значение. |

### Свойства

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| accessKey | Позволяет получить или задать клавишу доступа для элемента. |
| *атрибут*.value | Позволяет получить или задать значение для атрибута. |
| clientHeight | Возвращает высоту элемента в пикселях, включая внутренние отступы ([padding](https://puzzleweb.ru/css/pr_padding.php)). Внешние отступы ([margin](https://puzzleweb.ru/css/pr_margin.php)), рамки ([border](https://puzzleweb.ru/css/pr_border.php)) и горизонтальная полоса прокрутки (если она есть) не учитываются. |
| clientWidth | Возвращает ширину элемента в пикселях, включая внутренние отступы ([padding](https://puzzleweb.ru/css/pr_padding.php)). Внешние отступы ([margin](https://puzzleweb.ru/css/pr_margin.php)), рамки ([border](https://puzzleweb.ru/css/pr_border.php)) и вертикальная полоса прокрутки (если она есть) не учитываются. |
| [firstChild](https://puzzleweb.ru/javascript/element_firstchild.php) | Возвращает ссылку на первый дочерний узел внутри указанного узла. |
| firstElementChild | Возвращает первый дочерний элемент, если дочерних элементов нет, то возвращает null. |
| lastElementChild | Возвращает последний дочерний элемент, если дочерних элементов нет, то возвращает null. |
| [innerHTML](https://puzzleweb.ru/javascript/element_innerhtml.php) | Представляет содержимое, которое находится между открывающим и закрывающим тегами текущего элемента. |
| textContent | містить тільки текст всередині елемента |
| [parentNode](https://puzzleweb.ru/javascript/element_parentnode.php) | Возвращает родительский элемент указанного узла. |
| [style](https://puzzleweb.ru/javascript/element_style.php) | Объект, представляющий атрибут style элемента, он позволяет получить или установить встроенный стиль элемента. |
| textContent | Содержит текстовое содержимое узла и его потомков.   |  |  | | --- | --- | | 1  2  3  4  5  6  7 | <p>Внутри элемента &lt;p&gt;. <em>Внутри элемента &lt;em&gt;.</em></p>    <script>  var p = document.querySelector("p");    alert(p.textContent);   // Внутри элемента <p>. Внутри элемента <em>.  </script> | |

Приклад. Зчитування значення з текстового поля

----------- у тексті сторінки --------------

<input type="text" id="MyText" value="" /><br>

----------- звертання до елемента -------

var name = document.getElementById("MyText").**value**;

alert("Hello " + name);

Приклад. Перевірка стану чекбокса

----------- у тексті сторінки --------------

<input type="checkbox" id="MyCheckbox" value="10" /><br>

----------- звертання до елемента -------

if (document.getElementById("MyCheckbox").**checked**) {

alert("Is checked")

}

else {

alert("Is not checked");

}

Приклад. Аналіз значення з випадаючого списку

----------- у тексті сторінки --------------

<select id="MySelect">

<option value="1">text1</option>

<option value="2">text2</option>

<option value="3">text3</option>

</select>

----------- звертання до елемента -------

var value = document.getElementById("MySelect").**value**;

# Створення та видалення вузлів

Для створення елементів використовуються такі методи:

|  |  |
| --- | --- |
| **document.createElement (tag)** | Створює новий елемент із зазначеним тегом:  var div = **document.createElement** ('div'); |
| **document.createTextNode (text)** | Створює новий текстовий вузол з даним текстом:  var textElem = **document.createTextNode** ('Тут був я "); |
| **elem.cloneNode (true**) | створить «глибоку» копію елемента - разом з атрибутами, включаючи піделементи. Якщо ж викликати з аргументом false, то копія буде зроблена без дочірніх елементів |

# Додавання елемента: appendChild, insertBefore

|  |  |
| --- | --- |
| parentElem.**appendChild** (elem) | Додає elem в кінець дочірніх елементів parentElem |
| parentElem.insertBefore (elem, nextSibling) | Вставляє elem в колекцію дітей parentElem, перед елементом nextSibling |

**Видалення вузлів**

Для видалення вузла є два методи:

|  |  |
| --- | --- |
| **parentElem.removeChild (elem)** | Видаляє elem зі списку дітей parentElem. |
| **parentElem.replaceChild (newElem, elem)** | Серед дітей parentElem видаляє elem і вставляє на його місце newElem. |

===================================================

Обробка події натиснення на кнопку

<input type="button" value=" НАПИС НА КНОПЦІ " onclick=" ФУНКЦІЯ ()" />

Приклад

<head>

<meta charset="utf-8" />

<title></title>

<script>

function **getSumm**() {

var firstNumber = document.getElementById("first").value;

var secondNumber = document.getElementById("second").value;

var s = +firstNumber + +secondNumber;

document.getElementById("Summ").value=s;

}

</script>

</head>

<body>

First number <input type="text" id="first" value="0"/><br>

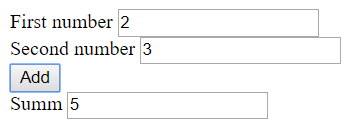
Second number <input type="text" id="second" value="0" /><br>

<input type="button" name="name" value="Add" onclick="getSumm()" /><br>

Summ <input type="text" id="Summ" value="0" /><br>

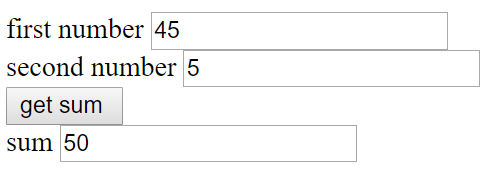
</body>

</html>



=========================== приклади =================

Інший підхід з вибіркою елементів



<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

function getSum() {

let inputs = document.getElementsByTagName("input");

let f = +inputs[0].value;

let s = +inputs[1].value;

inputs[3].value=f+s;

}

</script>

</head>

<body>

<form action="/www.gh.com" method="get">

first number

<input type="text" name="first" value="3" /><br>

second number

<input type="text" name="sec" value="5" /><br>

<input type="button" value="get sum " onclick="getSum()" /> <br>

sum

<input type="text" name="sum" value="" />

</form>

</body>

</html>

//===================================================

Задача 0. Знайти суму двох чисел (викристати різні підходи для вибірки елементів)

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

//----- Використовуємо доступ до елемента за id

//function getSum() {

// var number1 = parseFloat(document.getElementById('firstNumber').value);

// var number2 = parseFloat(document.getElementById('secondNumber').value);

// var s = number1 + number2;

// document.getElementById('summ').value = s;

//}

//------ Використаємо querySelector

//function getSum() {

// var number1 = parseFloat(document.querySelector("#firstNumber").value);

// var number2 = parseFloat(document.querySelector('#secondNumber').value);

// var s = number1 + number2;

// document.getElementById('summ').value = s;

//}

//------ Використаємо getElementsByTagName

//function getSum() {

// var inputElements = document.getElementsByTagName("input");

// var number1 = parseFloat(inputElements[0].value);

// var number2 = parseFloat(inputElements[1].value);

// var s = number1 + number2;

// inputElements[3].value = s;

//}

//-------- Використаємо querySelectorAll

function getSum() {

var inputElements = document.querySelectorAll("input[type='text']");

var number1 = parseFloat(inputElements[0].value);

var number2 = parseFloat(inputElements[1].value);

var s = number1 + number2;

inputElements[2].value = s;

}

</script>

</head>

<body>

<label for="firstNumber">First number : </label>

<input type="text" id="firstNumber" value="0" /> <br>

<label for="secondNumber">Second number : </label>

<input type="text" id="secondNumber" value="0" /><br>

<input type="button" name="name" value="Find sum" onclick="getSum()" /><br>

<label for="summ">Summ : </label>

<input type="text" id="summ" value="" /><br>

</body>

</html>

Задача 1. Користувач задає рік народження. Визначити кількість років користувача

Задача 2. Конвертер валют.

Задача 3. Калькулятор з вибором операції (додавання, віднімання, множення, ділення).

Задача 4. Визначення вартості замовлення (за допомогою чекбоксів вибирати першу страву, другу і десерт).

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

function getSumOfGroup(checkboxArr) {

Sum = 0;

for (var i = 0; i < checkboxArr.length; i++) {

if (checkboxArr[i].checked)

{

Sum += +checkboxArr[i].value;

}

}

return Sum;

}

function getTotalSum() {

var firstDishesGroup = document.querySelectorAll("[name='firstdish']");

var totalPrice1 = getSumOfGroup(firstDishesGroup);

var secondDishesGroup = document.querySelectorAll("[name='seconddish']");

var totalPrice2 = getSumOfGroup(secondDishesGroup);

var liquidsGroup = document.querySelectorAll("[name='liquid']");

var totalPrice3 = getSumOfGroup(liquidsGroup);

var total = totalPrice1 + totalPrice2 + totalPrice3;

document.getElementById("summ").value = total + "(Перша страва-" + totalPrice1 +

", Друга страва- " + totalPrice2 + ", Третя страва- " + totalPrice3 + ")";

}

</script>

</head>

<body>

Sup:

<input name="firstdish" type="checkbox" id="Sup" value="20" /> <br>

Borsch:

<input name="firstdish" type="checkbox" id="Borsch" value="25" /> <br><hr>

Rus:

<input name="seconddish" type="checkbox" id="Rus" value="15" /> <br>

Potato:

<input name="seconddish" type="checkbox" id="Potato" value="23" /> <br><hr>

Tea:

<input name="liquid" type="checkbox" id="Tea" value="10" /> <br>\

Kompot:

<input name="liquid" type="checkbox" id="kompot" value="8" /> <br>

<input type="button" name="name" value="Find sum" onclick="getTotalSum()" /><br>

<label for="summ">Summ : </label>

<input type="text" id="summ" value="" /><br>

</body>

</html>

Задача. Дано списки продуктів. Знайти сумарну вартість

================= СТВОРЕННЯ ЕЛЕМЕНТІВ =============

Задача. Зробити кнопку, при натисненні на яку з’являється додатковий input

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

function addF() {

//1.Створюємо новий інпут

var m = document.createElement('input');

//2.Задаємо значення його властивостей

m.type = 'text';

m.value = '0';

//3.Додаємо елемент у потрібний батьківський елемент

var d = document.getElementById('MyDiv');

d.appendChild(m);

//----------------------

d.innerHTML += "<br>";

}

</script>

</head>

<body>

<input type="button" value="add" onclick="addF()" />

<div id="MyDiv">

</div>

</body>

</html>

Задача. Відобразити одновимірний масив за допомогою таблиці

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

var a = [111,21,54,34];

function showTable() {

------- ДОРОБИТИ -----------------------

}

</script>

</head>

<body>

<input type="button" value="add" onclick="showTable()" />

<div id="MyDiv">

<table id="MyTable">

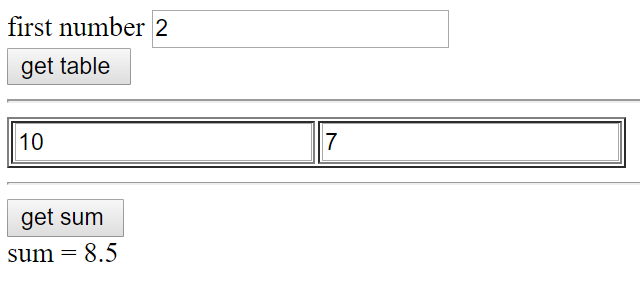
</table>

</div>

</body>

</html>

Приклад. Користувач задає кількість оцінок і натискає на кнопку «get table». В результаті формується таблиця з input, куди користувач вводить оцінки. Після цього натискає на кнопку “get sum” і знаходить середнє значення.



<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

function drawTable() {

var marks\_table = document.createElement("table");

marks\_table.id = "marks";

var row = document.createElement("tr");

var marks\_count = parseInt(document.getElementById("marks\_count").value);

for (var i = 0; i < marks\_count; i++) {

let td = document.createElement("td"); //Створюємо td

let inp = document.createElement("input"); //Свторюємо input

inp.type = "text"; //Налаштовуємо input

inp.value = "10";

td.appendChild(inp); //Вставляємо input в td

row.appendChild(td); //Всталяємо td у tr

}

marks\_table.appendChild(row);

marks\_table.border = " solid black 2px";

var div = document.getElementById("myTable");

div.innerHTML = "";

div.appendChild(marks\_table);

}

function getSum() {

//var table = document.getElementById("marks");

//let inputs = table.getElementsByTagName("input");

let inputs = document.querySelectorAll("table input");

var s = 0;

for (var i = 0; i < inputs.length; i++) {

s += +inputs[i].value;

}

let average = s / inputs.length;

document.getElementById("result").innerText = average;

}

</script>

</head>

<body>

<div>

first number

<input type="text" id="marks\_count" value="" /><br>

<input type="button" value="get table " onclick="drawTable()" /> <br>

<hr>

</div>

<div id="myTable">

</div>

<div>

<hr>

<input type="button" value="get sum " onclick="getSum()" /> <br>

sum = <span id="result"> </span>

</div>

</body>

</html>

Задача. Знайти середню оцінку учня. Кількість та оцінки вводяться користувачем (використати таблицю з компонентами input).

Задача. Сформувати список завдань ( з можливістю подальшого вилучення).

------------------------------------------------

Задача. Сформувати список працівників (ПІБ, посада, заробітна плата). По одному додається інформація про працівників у вигляді таблиці (як у попередньому завданні). Потім знайти сумарну заробітну плату працівників, у яких заробітна плата більша за 3000.

----------------------------------------------------

Задача. Знайти сумарну виручку від діяльності магазину протягом вказаної кількості тижнів (кожного тижня працює 5 днів).

-------------------------------------

Задача 5. Визначення вартості квитків (кількість – число, тип квитка – купе, загальний, плацкарт). На сторінці розмістити відповідні елементи для вводу потрібної інформації