**Въведение в програмен език JavaScript**

Програмният език JavaScript е интерпретаторен език от високо ниво. Както повечето скрипт езици и JavaScript поддържа динамично типизиране – типа на данните се определя в момента на инициализацията им (задава им се стойност). JavaScript е хибриден език, прототипно-базиран, обектно ориентиран и поддържа функционално програмиране. Има вградени приложни програмни интерфейси (API) за работа с текст, масиви, списъци, регулярни изрази и др.

Програмният код на JavaScript се интегрира директно в HTML страниците чрез етикет s𝑐𝑟𝑖𝑝𝑡 или чрез текстов файл с разширение 𝑗𝑠. Следователно, за въвеждането на изходния JavaScript код може да използвате кой да е текстов редактор. На практика се работи с on-line или off-line развойни среди (IDE). Основният недостатък на on-line развойните среди е невъзможността да ги използвате, ако нямате достъп до Интернет или мрежовата връзка в даден момент е много бавна. Алтернативата е да използвате off-line развойна среда, която трябва да инсталирате на вашия компютър.

Използваме Node.js -> съдържа браузърно ядро – IDE

Типизация на данните в JavaScript В JavaScript всички данни се декларират чрез ключови думи 𝒗𝒂𝒓 или 𝒍𝒆𝒕 (ES6+), без да се задава техния тип явно. Вътрешно, след инициализация на данните, те се причисляват към един от следните типове с които JavaScript работи:

• Числа;

• низове (текст);

• булеви данни;

• функции;

• обекти.

Програмно можем да определим типа на данните чрез използване на ключова дума (оператор) 𝑡𝑦𝑝𝑒𝑜𝑓. Синтаксисът е „𝑡𝑦𝑝𝑒𝑜𝑓 данна”. Този израз връща вътрешния тип на данна.

Примери:

**Number->** 10, 3.15, -1.5

**String -> “**Hello” , …

**Undefined** -> променлива на която не присвоена стойност

**Bigint -**>let x = BigInt("123456789012345678901234567890");

**Boolean**

let x = 5;

let y = 5;

let z = 6;

(x == y) // Returns true

(x == z) // Returns false

**Null**  
**Symbol   
Object** ->може да съдържа както вградени обекти, така и дефинирани от потребителя обекти

Built-in object types can be: objects, arrays, dates, maps, sets, intarrays, floatarrays, promises, and more

// Numbers:

let length = 16;

let weight = 7.5;

// Strings:

let color = "Yellow";

let lastName = "Johnson";

// Booleans

let x = true;

let y = false;

// Object:

const person = {firstName:"John", lastName:"Doe"};

// Array object:

const cars = ["Saab", "Volvo", "BMW"];

// Date object:

const date = new Date("2022-03-25");

**Пример за употребата на typeof .**

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML =

typeof 0 + "<br>" +

typeof 314 + "<br>" +

typeof 3.14 + "<br>" +

typeof (3) + "<br>" +

typeof (3 + 4);

</script>

**Пример за създаване на обект и дефиниране на метод**

const person = {

firstName: "John",

lastName: "Doe",

id: 5566,

fullName: function() {

return this.firstName + " " + this.lastName;

}

};

function solve(input){

let numAsString=input.toString();//"123654"

let sum=0;

for (let index = 0; index < numAsString.length; index++) {

let currNum=Number(numAsString[index]);

sum=sum+currNum;

}console.log(`${sum}`);

}

solve(123654)

Основни методи за обработка на масиви

Картина, която съдържа текст, екранна снимка, софтуер, Компютърна икона

Описанието е генерирано автоматично

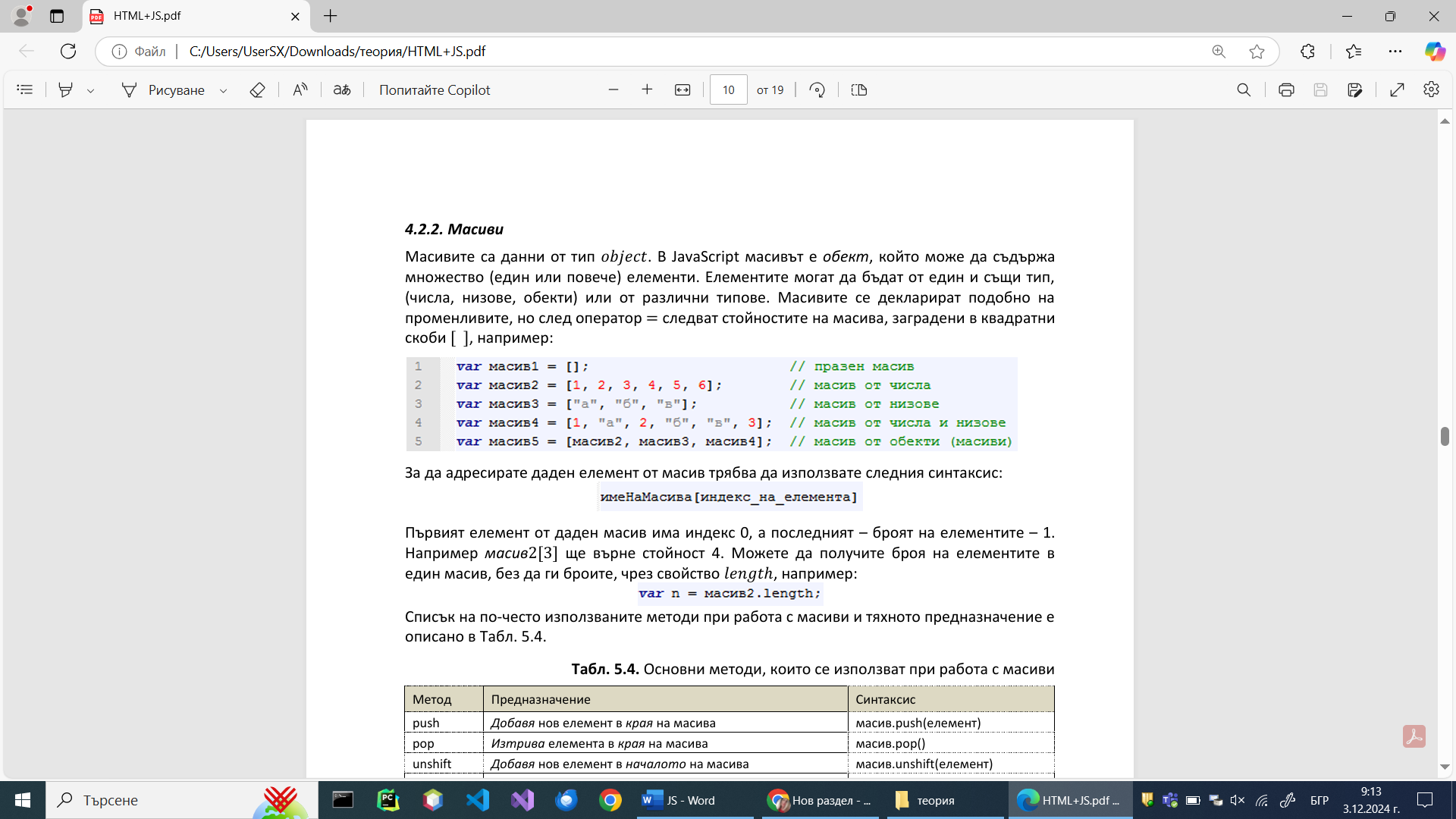
Основни методи за обработка на низове

Картина, която съдържа текст, електроника, екранна снимка, софтуер

Описанието е генерирано автоматично

Масиви

Масивите са данни от тип 𝑜𝑏𝑗𝑒𝑐𝑡. В JavaScript масивът е обект, който може да съдържа множество (един или повече) елементи. Елементите могат да бъдат от един и същи тип, (числа, низове, обекти) или от различни типове. Масивите се декларират подобно на променливите, но след оператор = следват стойностите на масива, заградени в квадратни скоби [ ], например:



Задачи с управляващи конструкции, цикли, масиви, създаване на обекти.

1. Напишете JS функция, която получава масив от числа и отпечатва сумата на първия и последния елемент в този масив.

|  |  |
| --- | --- |
| **Input** | **Output** |
| [20, 30, 40] | 60 |
| [10, 17, 22, 33, 40] | 43 |
| [11, 58, 69] | 80 |

* Да се редактира програмния код така, че функцията да приеме като аргументи 3 масива и да премахне елементи с стойност 20, 22, 69 и да се изведат останалите елементи след премахването.

1. Напишете JS функция, която получава число n и масив от елементи. Вашата задача е да създадете нов масив с n числа от оригиналния масив, да го обърнете и да отпечатате елементите му на един ред, разделени с интервал.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3, [10, 20, 30, 40, 50] | 30 20 10 |
| 4, [-1, 20, 99, 5] | 5 99 20 -1 |
| 2, [66, 43, 75, 89, 47] | 43 66 |

1. Напишете JS функция, която получава текст като първи параметър и една дума като втори. Намерете всички срещания на тази дума в текста и ги заменете със съответния брой '\*'.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'A small sentence with some words', 'small' | A \*\*\*\*\* sentence with some words |
| 'Find the hidden word', 'hidden' | Find the \*\*\*\*\*\* word |