ლაბორატორიული 3

**მონაცემთა სტრქუტურები**

1. **რიცხვები,** **სტრიქონები**

**სტრიქონის რიცხვად გარდაქმნის მეთოდები: უნარული +, parseInt, parseFloat**

უნარული პლუსსს გადაყავს სტრიქონი რიცხვში თუ სტრიქონი არის კორექტული ან აბრუნებს NaN-ს საწინააღმდეგო შემთხვევაში. ახდენს whitespace სიმბოლოების იგნორირებას სტრიქონის დასაწყისში და ბოლოში.

var s = "12.53";

alert( +s ); // 12.53

alert( +"12test" ); // NaN

alert( +" -10" ); // -10

alert( +" \n20 \n" ); // 20, \n არის whitespace სიმბოლო

alert( +"" ); // 0

alert( +"2 4" ); // NaN

parseInt და parseFloat ცდილობს გარდაქმნას სტრიქონი რიცხვად სანამ შესაძლებელია, როგორც კი არაკორექტული სიმბოლო შეხვდება, ჩერდება და იმ მომენტისთვის მიღებულ შედეგს აბრუნებს.

alert( parseInt('12px') ) // 12, გაჩერდება'p'-ზე

alert( parseFloat('12.3.4') ) // 12.3, გაჩერდება მეორე წერტილზე

**რიცხვის სტრიქონად გარდაქმნა: toString**

სინტაქსი: toString(ათვლითი\_სისტემის\_ფუძე);

var n = 255;

alert( n.toString(16) ); // ff

var m = 4;

alert( m.toString(2) ); // 100

**რიცხვების დამრგვალება: floor, ceil, round**

alert( Math.floor(3.1) ); // 3

alert( Math.ceil(3.1) ); // 4

alert( Math.round(3.1) ); // 3  
alert( Math.round(4.6) ); // 5

1. **სტრიქონები**

**სიგრძის გაგება:**

var str = "test";

alert( str.length ); // 4

**ცალკეული სიმბოლოების ამოღება:**

var str = "javascript";

alert( str.charAt(0) ); // "j"

var str = "angular";

alert( str[0] ); // "a"

**რეგისტრის შეცვლა:**

alert( "test".toUpperCase() ); // TEST

alert( "Test" [0].toLowerCase() ); // 'test'

**სტრიქონში ძებნა:**

var str = "Javascript is cool";

alert( str.indexOf("Javascript") ); // 0

alert( str.indexOf("is") ); // 11

alert( str.indexOf("javascript") ); // -1

**სტრიქონიდან ნაწილის ამოჭრა: substr, substring, slice.**

სინტაქსი:

**substring(****start [, end])**

var str = "javascript";

alert(str.substring(0,1)); // "j" იღებს start-დან end-მდე.

სინტაქსი:

**substr(start [, length])**

var str = "javascript";

str = str.substr(2,4); // vasc

alert(str)

სინტაქსი:

**slice(start [, end])**

**დავალებები:**

1. **L301** დაწერეთ სკრიპტი რომელიც შეამოწმებს შეიცავს თუ არა შეყვანილი სტრიქონი აკრძალულ სიტყვებს. აკრძალული სიტყვებია: რეკლამა, მარკეტინგი, ვირუსი
2. **L302** დაწერეთ სკრიპტი რომელიც შეყვანილ სტრიქონს, თუ მისი სიგრძე აჭარბებს 15 სიმბოლოს, მოაჭრის ზედმეტ ნაწილს და მის ნაცვლად მიადგავს ... მაგალითად: “some very long text” გადაიქცევა “some very ...”
3. **L303** დაწერეთ სკრიპტი რომელიც ფასს, შეყვანილს სტრიქონით სახით ფორმატით {currency}price, გადაიყვანს საპირისიპირო ვალუტაში (თუ მოცემულია ლარებში გადაიყვანს დოლარებში და პირიქით) კურსთი 1დოლარი = 2.6 ლარს. მაგალითი: $10 პასუხი 26, L26 პასუხი $10
4. **მასივები**

მასივის გამოცხადება / შექმნა

**ცარიელი:**

var arr = [];

**საწყისი ელემენტებით:**

var skills = ["HTML5", "CSS3", "Javascript"];

**ცალკეული ელემენტების ამოღება:**

alert( skills [0] ); // HTML5

alert( skills [1] ); // CSS3

alert( skills [2] ); // Javascript

**ელემენტის შეცვლა:**

skills [2] = 'PHP'; // ["HTML5", "CSS3", "PHP"]

**ელემენტის დამატება:**

skills [3] = 'Javascript'; // ["HTML5", "CSS3", "PHP", "Javascript"]

**ელემენტების რაოდენობის გაგება:**

var skills = ["HTML5", "CSS3", "Javascript"];

alert(skills.length ); // 3

**მეთოდები pop/push, shift/unshift**

მასივში არსებული მეთოდებით შესაძლებელია ისეთი მონაცემთა სტრუქტურების მიღება როგორიცაა რიგი და სტეკი.

**pop** - შლის ბოლო ელემენტს და აბრუნებს მას.

var skills = ["HTML5", "CSS3", "Javascript"];

alert(skills.pop() ); // წავშალეთ Javascript

alert(skills); // დარჩა HTML5, CSS3

**push** - ამატებს ელემენტს მასივის ბოლოში

var skills = ["HTML5", "CSS3"];

skills.push("Javascript");

alert(skills); // HTML5, CSS3, Javascript

**shift** - შლის პირველ ელემენტს და აბრუნებს მას.

var skills = ["HTML5", "CSS3", "Javascript"];

alert( skills.shift() ); // წავშალეთ HTML5

alert( skills ); // დარჩა CSS3, Javascript

**unshift** - ამატებს ელემენტს მასივის დასაწყისში

var skills = ["HTML5", "CSS3"];

skills. unshift ("Javascript");

alert(skills); // Javascript, HTML5, CSS3

**დავალებები:**

1. **L304** დაწერეთ სკრიპტი, რომელიც ჯერ კითხავს მომხმარებელის რამდენი სტუდენტის შეყვანა სურს, ხოლო შემდეგ შეატანინებს იმდენ სახელს რამდენი სტუდენტის შეყვანაც მოისურვა მომხმარებელმა
2. **L305** დაწერეთ სკრიპტი, რომელიც მომხმარებლის მიერ შეტანილი სტუდენტების სიიდან(წინა დავალების ანალოგიურად) გამოიტანს სტუდენტების სიას დალაგებულს ანბანის მიხედვით
3. **L306** დაწერეთ სკრიპტი რომელიც მოცემული სტუდენტების სიიდან(შეაქვს მომხმარებელს) გამოიტანს მხოლოდ ისეთებს, ვისი სახელიც სრულად ან ნაწილობრივ ემთხვევა მომხმარებლის მიერ შეტანილ სტრიქონს.
4. **L307** დაწერეთ სკრიპტი რომელიც სახელების მოცემული 2 სიიდან (შეაქვს მომხმარებელს) შეადგენს გაერთიანებულ სიას ისე, რომ სახელები არ მეორდებოდეს
5. **L308** დაწერეთ სკრიპტი რომელიც მომხმარებლის მიერ შეტანილ სტუდენტების სიას გამოიტანს შებრუნებული თანმიმდევრობით