1. შექმენით პროგრამა, რომელიც ნებისმიერი 10 ელემენტისგან შემდგარი მასივიდან გამოიტანს ყოველ მესამე ელემენტს ეკრანზე. შეასრულეთ ეს დავალება ციკლების მეშვეობით.
2. შექმენით, პროგრამა, რომელიც 10-დან 1-მდე რიცხვებს გამოიტანს. მაგ. 10,9,8,7,6,5,4,3,2,1, შეასრულეთ ეს დავალება ციკლების მეშვეობით.
3. მოცემული მასივიდან  
   let names = ['John', 'Nick', 'Bob', 'Mary', 'Bob', 'Sue', 'Ann', 'Bob', 'Bob']

ამოიღეთ 'Bob' სახელი ყველგან სადაც ის შეგხვდებათ და დაბრუნეთ მასივი 'Bob'-სახელის გარეშე. შედეგი ესეთი უნდა იყოს: ['John', 'Nick', 'Mary', 'Sue', 'Ann'] შეასრულეთ ეს დავალება ციკლების მეშვეობით, ასევე კარგი იქნება თუ იგივე დავალებას ციკლური ოპერაციების გამოყენების გარეშეც შეასრულებთ.

1. ციკლური ოპერაციების მეშვეობით შექმენით ფუნქცია, რომელიც დაიანგარიშებს კონკრეტული რიცხვის ხარისხს, მაგ. შექმენით ფუნქცია სახელით: fun და პარამეტრებით: number, power, მაგ. თუ რიცხვია 5 და ხარისხი 3, მაშინ თქვენმა ფუნქციამ უნდა დააბრუნოს 5\*5\*5 ანუ 125.
2. შექმენით ფუნქცია, რომელიც 1-დან 100-მდე რიცხვებში დაიანგარიშებს თუ რომელი რიცხვები იყოფა 3-ზე უნაშთოდ, და იმ რიცხვების ადგილას რომლებიც იყოფა 3-ზე უნაშთოდ გამოიტანს, Fizz, ასევე იგივე ფუნქციის მიხედვით გამოიანგარიშეთ თუ რიცხვი 5-ზე იყოფა უნაშთოდ და გამოიტანეთ Buzz, ხოლო თუ რიცხვი იყოფა 3-ზეც და 5-ზეც უნაშთოდ გამოიტანეთ FizzBuzz.

მაგ.   
1

2

3 Fizz

4

5 Buzz

6 Fizz

7

8

9 Fizz

10 Buzz

11

12 Fizz

13

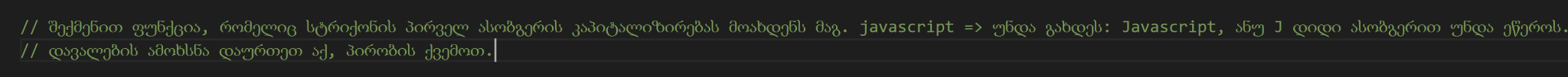
14

15 FizzBuzz

… 100-მდე.

1. შექმენით ფუნქცია,მაგ. სახელით: factorialCalculator(number), რომელიც დაიანგარიშებს და ეკრანზე გამოიტანს მისთვის პარამეტრის სახით მიწოდებული რიცხვის ფაქტორიალს, მაგ. 5! => 2\*3\*4\*5 = 120,
2. შექმენით ფუნქცია, რომელიც სტრიქონის პირველ ასობგერის კაპიტალიზირებას მოახდენს მაგ. javascript => უნდა გახდეს: Javascript, ანუ J დიდი ასობგერით უნდა ეწეროს.
3. შექმენით ფუნქცია, რომელიც შეამოწმებს არის თუ არა მასზე მიწოდებული სტრიქონი ცარიელი. მაგ. blankStringChecker(testString) => რომელიც დააბრუნებს this string is blank-ს თუ ცარიელია/ this string is not blank-ს თუ ცარიელი არაა.
4. შექმენით ფუნქცია, რომელიც მასზე მიწოდებულ სტრიქონს გადააქცევს მასივად. მაგ. stringToArray(testString) => თუ მივაწვდით სტრიქონს “Hello, World” => გამოიტანს მის ვერსიას მასივში. [“Hello”, “World”]
5. შექმენით ერთგვარი ემაილის დამცველი ფუნქცია, რომლის ფუნქციონალი მდგომარეობს შემდგომში: მომხმარებელი გაწვდის ემაილს წერტილით გამოყოფილ თავისი სახელით და გვარით, [beqa.beqauri@gmail.com](mailto:beqa.beqauri@gmail.com), თქვენი ფუნქცია კი გამოიტანს მის დაცულ ვერსიას ამ ფორმით: [beqa……..@gmail.com](mailto:beqa.beqauri@gmail.com).

წერტილების ოდენობა უნდა ემთხვეოდეს გვარის სიგრძეს.

შენიშვნა: 1-დან მე-6 დავალების ჩათვლით დაგჭირდებათ ციკლური ოპერაციების განხორციელება და შეეცადეთ იგივე დავალებები დაწეროთ, როგორც ორივე ციკლური ოპერანდის გამოყენებით ცალ-ცალკე, როგორც For loop-ის ისე While loop-ის.  
  
ასევე Github-ზე დავალების ატვირთვისას დავალების ზემოთ კომენტარის სახით აუცილებლად დაურთეთ იმ ამოცანის პირობა რასაც ასრულებთ. მაგ. ესე როგორც ფოტოზეა.  


დამატებითი მასალა:  
სტრიქონებზე სამუშაო მეთოდები:   
<https://www.w3schools.com/js/js_string_methods.asp>  
For ციკლები:  
<https://www.w3schools.com/js/js_loop_for.asp>  
While ციკლები:  
<https://www.w3schools.com/js/js_loop_while.asp>  
For of ციკლური ოპერაციები:  
<https://www.w3schools.com/js/js_loop_forof.asp>