|  |  |  |
| --- | --- | --- |
| **№** | **Условие** | **Решение** |
| 1 | Является ли слово палиндромом  str.split('').filter((letter)=>str!==-1).join('') | function palindrom (word){  word=word.toLowerCase (); //перевод в нижнний регистр  let wordChanged=word.split(''); //представление строки в виде массива с помощью метода split и присвоение др. переменной\*  wordChanged=wordChanged.reverse(); //разворачиваем массив  wordChanged=wordChanged.join(''); //обратно "слепляем" массив в строку  if(wordChanged==word) return true; //эту строку можно упростить как return (wordChanged==word). Все будет работать  else return false  }  console.log(palindrom ('строковое значение'))  ЛУЧШЕ!!!! использовать короткую запись кода: return word==word.split('').reverse().join(''); |
| 2 | После тяжелой четверти в офисе вы решаете немного отдохнуть в отпуске. Итак, вы забронируете рейс для себя и своей девушки и постараетесь оставить весь беспорядок позади.  Вам понадобится арендованный автомобиль, чтобы передвигаться в отпуске. Менеджер по аренде автомобилей делает вам несколько хороших предложений. Каждый день аренды автомобиля  стоит 40 долларов. Если вы арендуете автомобиль на 7 или более дней, вы получаете скидку в размере 50 долларов США. В качестве альтернативы, если вы арендуете автомобиль на 3  или более дней, вы получаете скидку в размере 20 долларов США. Напишите код, который выдает общую сумму за разные дни (d). | function rentalCarCost(d) {  return d\*40-(d < 3 ? 0 : d < 7 ? 20 : 50) |
| 3 | Реализовать функцию с тремя параметрами: function mathOperation(arg1, arg2, operation), где arg1, arg2 — значения аргументов, operation — строка с названием операции.  В зависимости от переданного значения выполнить одну из арифметических операций (использовать функции из пункта 5) и вернуть полученное значение (применить switch). | function mathOperation (arg1,arg2,operation){  let result;  switch(operation){  case'sum':result = arg1+arg2;  break;  case'ruz':result = arg1-arg2;  break;  case'proiz':result = arg1\*arg2;  break;  case'del':result = arg1/arg2;  break;  }  return result  }  console.log(mathOperation(1,2,'proiz'))// console.log вызываем за пределами функции!!! |
| 4 | Учитывая строку, вы должны вернуть строку, в которой каждый символ (с учетом регистра) повторяется один раз.(пример: получаем 'Игорь'-> возвращаем 'Ииггооррьь') | function doubleChar(str){  let string=str.split('');  string=string.map((symb)=>symb+symb);  string=string.join('');  return string  }  console.log(doubleChar('Игорь'))  короткая запись кода:function doubleChar(str){  return str.split('').map((symb)=>symb+symb).join('')  } |
| 5 | Учитывая массив целых чисел, вернуть новый массив с удвоением каждого значения. | function maps(x){  return x.map((a)=>a\*2)  } |
| 6 | Это ката об умножении заданного числа на восемь, если оно четное, и на девять в противном случае. | function simpleMultiplication(number) {  return number%2==0?number\*8:number\*9  }  короткая запись кода: return number \* (number%2?9:8) |
| 7 | Тимми и Сара думают, что они влюблены, но там, где они живут, они узнают об этом, только когда каждый сорвет по цветку. Если у одного из цветов четное количество лепестков, а у другого нечетное количество лепестков, это означает, что они влюблены.  Напишите функцию, которая будет принимать количество лепестков каждого цветка и возвращать true, если они влюблены, и false, если нет. | function lovefunc(flower1, flower2){  return flower1 % 2 !== flower2 % 2;  } |
| 8 | Напишите программу, которая находит сумму всех чисел от 1 до num. Число всегда будет положительным целым числом больше 0. | let result = 0;  for (var i = 1; i <= num; i++) {  result += i;  }    return result;  } |
| 9 | Вы получаете массив чисел, возвращаете сумму всех положительны// [1,-2,3,4,5]->[1,3,4,5]->13 | function positiveSum(arr) {  return arr.filter((number)=>number>0).reduce((a,b)=>a+b,0)// 0-это инициализация цикла,начальное значение  } |
| 10 | Дезоксирибонуклеиновая кислота, ДНК является основной молекулой хранения информации в биологических системах. Он состоит из четырех оснований нуклеиновых кислот гуанина («G»), цитозина («C»), аденина («A») и тимина («T»).  Рибонуклеиновая кислота, РНК, является основной молекулой-мессенджером в клетках. РНК немного отличается от ДНК своей химической структурой и не содержит тимина. В РНК тимин заменен другой нуклеиновой кислотой урацилом («U»).  Создайте функцию, которая переводит заданную строку ДНК в РНК. | function DNAtoRNA(dna) {  return dna.replace(/T/g,'U')// create a function which returns an RNA sequence from the given DNA sequence  } |
| 11 | Создайте функцию, которая принимает 2 строковых аргумента и возвращает целое число, соответствующее количеству вхождений второго аргумента в первом.  Если вхождений не найдено, должно быть возвращено число 0. | return str.split('').filter(c=>c==letter).length  strCount(‘слово’,’искомая буква’) |
| 12 | Кто помнит свое время на школьном дворе, когда девочки брали цветок и рвали его лепестки, произнося каждую из следующих фраз каждый раз, когда отрывался лепесток:  я тебя люблю  немного  много  страстно  безумно  нисколько  Когда был сорван последний лепесток, раздались крики волнения, мечты, нахлынувшие мысли и эмоции.  Ваша цель в этом ката — определить, какую фразу девушки сказали бы о цветке с заданным количеством лепестков, где nb\_petals > 0. | function howMuchILoveYou(nbPetals) {  let phrase = {  0: "not at all",  1: "I love you",  2: "a little",  3: "a lot",  4: "passionately",  5: "madly"  }  return phrase[nbPetals%6]  }  // небольшой комментарий- деление на 6 применяется для нахождения остатка. К примеру мы передаем функции число 20. 20 делится нацело на 6- остается 3 (6х3=18) остаток от деления 2 (20-18). Соответственно вернется 2: ‘a little’ |
| 13 | написать функцию - countBasketPrice - которая считает стоимость корзины [100, 200, 300, 400, 500, 6] | const basket = [100, 200, 300, 400, 500, 6]  function countBasketPrice(arr) {  return arr.reduce((acc, price) => acc + price, 0)  }  console.log(countBasketPrice(basket))  //reduce это как накопление, acc это результат предыдущего действия, price текущий элемент массива  так как при первом вызове небыло предыдущего действия нужно стартовое значение для acc  это и будет второй агрумент reduce – 0  иначе андефайн |
| 14 |  |  |
| 15 |  |  |
| 16 |  |  |