

Задание

Сделайте функцию, которая считает и выводит количество своих вызовов.

```
func(); //выведет 1  
func(); //выведет 2  
func(); //выведет 3  
func(); //выведет 4
```

Задание

создаем файл **buttons-counter.html**, в котором верстаем три кнопки, стилизуем их на свой вкус, создаем [обработчик кликов](#) (достаточно будет просто метода `onclick()`) для кнопок и считаем количество этих кликов по каждой кнопке, выводя результат в саму кнопку, т.е. как текстовое значение кнопки. Количество нажатий для каждой кнопки должно храниться в замыкании, т.е. должна быть только одна функция подсчета кликов, но использоваться для любого количества кнопок (подразумевается, что кнопок может быть как одна, так и пара сотен...).

Задание

создаем файл **pow-and-calculate.html** и реализуем функции, которые будут использовать понятие замыкание и каррирование и удовлетворять условиям:

- 1) возводить в степень число, а вызываться будет как `pow(x)(y)`;
- 2) суммировать числа и выводить результат, а вызываться как `calculate(a)(“+”)(b)`; или `calculate(a)(“-”)(b)`; (возможные операции - “+”, “-”, “/”, “*");
- 3) Не использовать метод `Math.pow()`;
- 4) Желательно сделать с помощью рекурсии функцию `pow(x)(y)`.

Пример:

1) **Вызов:**

```
pow(-2)(3);  
calculate(1)(“+”)(2);
```

Результат:

Результат $-2^3 = -8$

Результат $1+2 = 3$

2) **Вызов:**

```
pow(4)(2);  
calculate(3)(“*”)(7);
```

Результат:

Результат $4^2 = 16$

Результат $3 \cdot 7 = 21$

3) **Вызов:**

`pow(16)(0);`

`calculate(15)("/")(3);`

Результат:

Результат $16^0 = 1$

Результат $15/3 = 5$

ЗАДАНИЕ

создаем файл ***compression.html***, в котором в js-скрипте создаем две функции *compress* и *uncompress*, которые будут выполнять следующее:

Пример:

1) **Вызов:**

`compress("a");`

Результат:

Результат: "a"

2) **Вызов:**

`compress("aaa");`

Результат:

Результат: "a3"

3) **Вызов:**

`compress("aabbb");`

Результат:

Результат: "a2b3"

4) **Вызов:**

`compress("aaabcc");`

Результат:

Результат: "a3b1c2"

5) **Вызов:**

`uncompress("a");`

Результат:

Результат: "a"

6) **Вызов:**

`uncompress("a5");`

Результат:

Результат: "aaaaa"

7) **Вызов:**

`uncompress("a2b3");`

Результат:

Результат: "aabbb"

8) **Вызов:**

`uncompress("a2b1c3");`

Результат:

Результат: "aabccc"

ЗАДАНИЕ

создаем файл **calcWithFunc.html** и пишем функции, которые будут выполнять следующие условия:

- 1) Для каждого числа должна быть своя функция, т.е. от 0 ("zero") до 9 ("nine");
- 2) Должна быть функция для каждой из следующих математических операций: *plus*, *minus*, *times* (умножение), *dividedBy* (деление);
- 3) Самая внешняя функция представляет собой левый операнд, самая внутренняя функция представляет собой правый операнд;
- 4) По возможности пользоваться стрелочными функциями;

Пример:

1) **Вызов:**

`seven(times(five()));`

Результат:

Результат: $7 * 5 = 35$

2) **Вызов:**

`four(plus(nine()));`

Результат:

Результат: $4 + 9 = 13$

3) **Вызов:**

`eight(minus(three()));`

Результат:

Результат: $8 - 3 = 5$

4) **Вызов:**

`six(dividedBy(two()));`

Результат:

Результат: $6 / 2 = 3$

Написать функции, которые будут выполнять следующие условия:

- 5) Для каждого числа должна быть своя функция, т.е. от 0 ("zero") до 9 ("nine");
- 6) Должна быть функция для каждой из следующих математических операций: *plus*, *minus*, *times* (умножение), *dividedBy* (деление);
- 7) Самая внешняя функция представляет собой левый операнд, самая внутренняя функция представляет собой правый операнд;
- 8) По возможности пользоваться стрелочными функциями;

Пример:

- 5) **Вызов:**

`seven(times(five()));`

Результат:

Результат: $7 * 5 = 35$

- 6) **Вызов:**

`four(plus(nine()));`

Результат:

Результат: $4 + 9 = 13$

- 7) **Вызов:**

`eight(minus(three()));`

Результат:

Результат: $8 - 3 = 5$

- 8) **Вызов:**

`six(dividedBy(two()));`

Результат:

Результат: $6 / 2 = 3$

Задание

создаем файл **countVowels.html** и создаем функцию `countVowelLetters`, которая будет возвращать количество русских гласных букв в строке, которая будет являться аргументом функции. Не использовать switch, вложенные циклы, регулярные выражения или 9-10 if для проверки... Можно использовать массивы или объекты для решения задачи, найти оптимальное и эффективное из доступных решение.

Пример:

- 1) **Вызов:**

countVowelLetters ("Пришла зима, запахло...");

Результат:

Количество гласных = 7

2) **Вызов:**

countVowelLetters ("Ghbdtn, z r dfv bp Hjccbb");

Результат:

Количество гласных = 0

3) **Вызов:**

countVowelLetters ("длинношнее");

Результат:

Количество гласных = 5

Задание

Создать страницу и написать javascript-сценарий, который спросит у пользователя:

- фамилию, имя, отчество РАЗДЕЛЬНО (тремя запросами prompt());
- возраст в годах (через prompt());
- пол (используя confirm(), например, "Ваш пол - мужской?");

и выведет с помощью alert() анкету пользователя по примеру:

Ваше ФИО: Иванов Иван Иванович

Ваш возраст в годах: 20

Ваш возраст в днях: 7300

Вы уже совершеннолетний

Ваш пол: мужской

Вы на пенсии: нет

Пенсионным считать возраст для обоих полов одинаковым и большим или равным 60 годам. Совершеннолетие >18 лет. Обязательно проверять вводимые данные от пользователя на не-число/число, на логичность, на длину (касается ФИО).

Задание

создаем страницу **testEven.html**. Добавляем js-сценарий, который будет принимать от пользователя число (используем `prompt()`) и вставлять дефис между двумя четными соседними числами, т.е. если пользователь ввел 025765824, то результатом будет 0-257658-2-4.

Так же создаем страницу **fizzbuzz.html**. Подключаем javascript-сценарий, в котором необходимо реализовать функцию `fizzBuzz(*)`, выполняющую вывод в консоль чисел от 1 до 100, но с условиями:

- a. если число кратно 3, то вместо числа вывести Fizz.
- b. если кратно 5, то вывести вместо числа Buzz.
- c. если число кратно 3 и 5 одновременно, то вывести вместо числа FizzBuzz.

для начала по простому пробуем с ифами свитчамиб тернарными – абы б работало. Потом пробуем вот так:

*В теле функции нельзя использовать **if, switch, тернарный оператор (a?b:c).**

Задание

создаем файл **quadratic.html** и пишем функцию `quadraticEquation`, которая на вход принимает коэффициенты квадратного уравнения, а возвращает вещественные корни этого или сообщает об их отсутствии.

теория

https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5

Пример:

Вызов:

`quadraticEquation(1, -8, 72);`

Результат:

уравнение $x^2 - 8x + 72 = 0$ не имеет вещественных корней

Вызов:

`quadraticEquation(1, 12, 36);`

Результат:

уравнение $x^2 + 12x + 36 = 0$ имеет один корень $x = -6$

Вызов:

quadraticEquation(4, -8, 1);

Результат:

уравнение $4x^2 - 8x + 1 = 0$ имеет корни $x_1 = 1.8660254037844386$ и $x_2 = 0.1339745962155614$