|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных Технологий

Кафедра ИППО

**ОТЧЕТ**

по лабораторной работе №2

по дисциплине

«Проектирование сред программирования и их компонентов»

Выполнил студент группы ИКМО-01-17 Алешин К.В.

Принял ассистент Гречин А.А.

Выполнено «\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 г.

Москва 2018

1. **Перепишите функцию, используя оператор '?' или '||**'

Следующая функция возвращает true, если параметр age больше 18. В ином случае она задаёт вопрос confirm и возвращает его результат.

function checkAge(age) {

if (age > 18) {

return true;

} else {

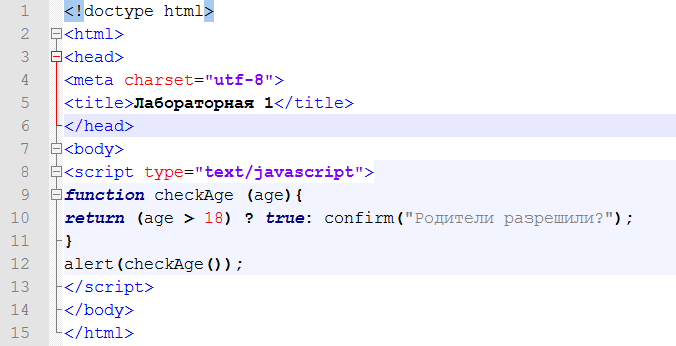
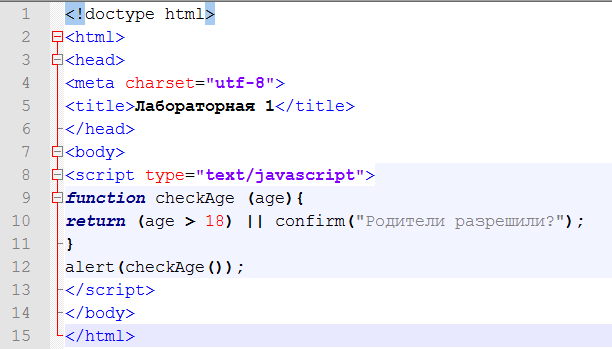
return confirm('Родители разрешили?');

}

}

Перепишите функцию, чтобы она делала то же самое, но без if, в одну строку. Сделайте два варианта функции checkAge:

Используя оператор '?' Используя оператор ||

1. **Функция min**

Задача «Hello World» для функций :)

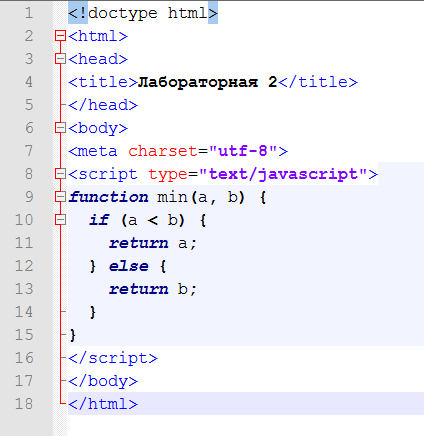
Напишите функцию min(a,b), которая возвращает меньшее из чисел a,b.

Пример вызовов:

min(2, 5) == 2

min(3, -1) == -1

min(1, 1) == 1



1. **Функция pow(x,n)**

Напишите функцию pow(x,n), которая возвращает x в степени n. Иначе говоря, умножает x на себя n раз и возвращает результат.

pow(3, 2) = 3 \* 3 = 9

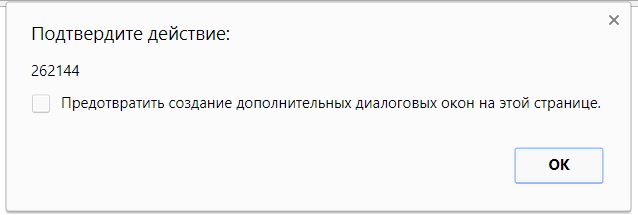
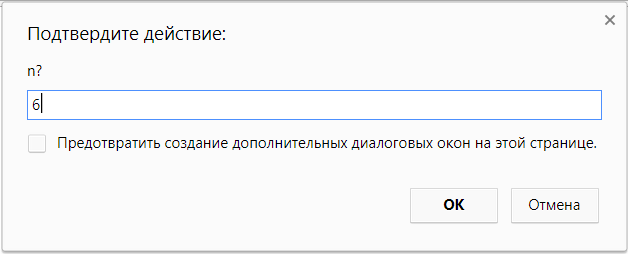
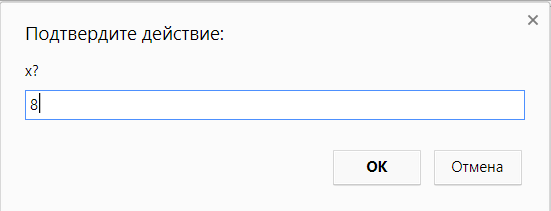
pow(3, 3) = 3 \* 3 \* 3 = 27

pow(1, 100) = 1 \* 1 \* ...\*1 = 1

Создайте страницу, которая запрашивает x и n, а затем выводит результат pow(x,n).

P.S. В этой задаче функция обязана поддерживать только натуральные значения n, т.е. целые от 1 и выше.





1. **Вычислить сумму чисел до данного**

Напишите функцию sumTo(n), которая для данного n вычисляет сумму чисел от 1 до n, например:

sumTo(1) = 1

sumTo(2) = 2 + 1 = 3

sumTo(3) = 3 + 2 + 1 = 6

sumTo(4) = 4 + 3 + 2 + 1 = 10

...

sumTo(100) = 100 + 99 + ... + 2 + 1 = 5050

Сделайте три варианта решения:

1. С использованием цикла.
2. Через рекурсию, т.к. sumTo(n) = n + sumTo(n-1) для n > 1.
3. С использованием формулы для суммы арифметической прогрессии.

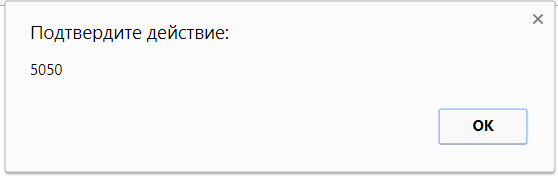
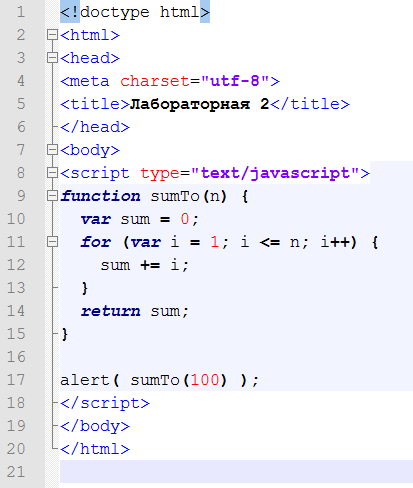
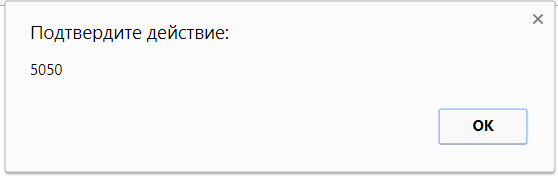
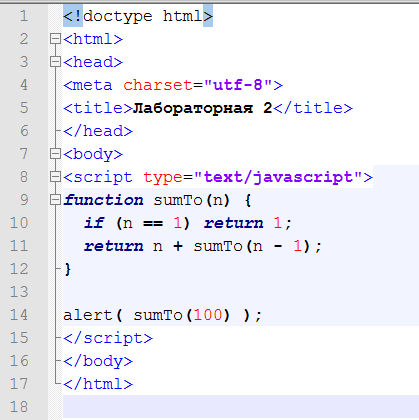
Пример работы вашей функции:

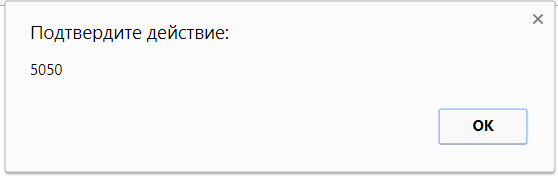
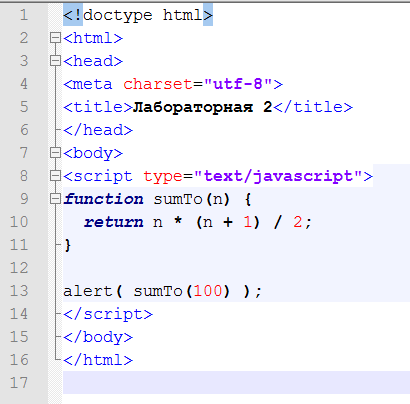
function sumTo(n) { /\*... ваш код ... \*/ }

alert( sumTo(100) ); // 5050

1.Какой вариант решения самый быстрый? Самый медленный? Почему?

2.Можно ли при помощи рекурсии посчитать sumTo(100000)? Если нет, то почему?

1. 
2. 
3. Решение **по формуле**: sumTo(n) = n\*(n+1)/2:



1. Решение по формуле работает быстрее всех, оно использует всего три операции для любого n, а цикл и рекурсия требует как минимум n операций сложения.

Вариант с циклом – второй по скорости, он быстрее рекурси, так как операций сложения столько же, но нет дополнительных вычислительных затрат на организацию вложенных вызовов. Рекурсия в данном случае работает медленнее всех.

1. Существует ограничение глубины вложенных вызовов, поэтому рекурсивный вызов sumTo(100000) выдаст ошибку.

**5.Вычислить факториал**

Факториа́л числа – это число, умноженное на «себя минус один», затем на «себя минус два» и так далее, до единицы. Обозначается n!

Определение факториала можно записать как:

n! = n \* (n - 1) \* (n - 2) \* ...\*1

Примеры значений для разных n:

1! = 1

2! = 2 \* 1 = 2

3! = 3 \* 2 \* 1 = 6

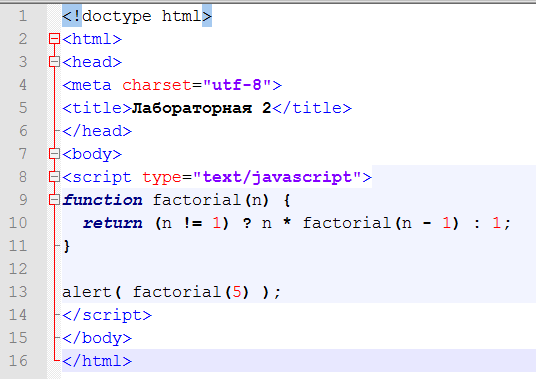
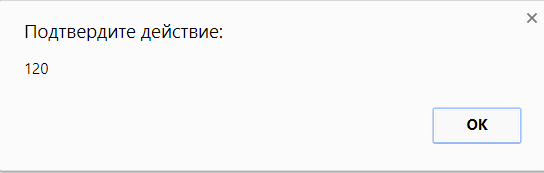
4! = 4 \* 3 \* 2 \* 1 = 24

5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

Задача – написать функцию factorial(n), которая возвращает факториал числа n!, используя рекурсивный вызов.

alert( factorial(5) ); // 120

Подсказка: обратите внимание, что n! можно записать как n \* (n-1)!. Например: 3! = 3\*2! = 3\*2\*1! = 6

**6.Числа Фибоначчи**

Последовательность чисел Фибоначчи имеет формулу Fn = Fn-1 + Fn-2. То есть, следующее число получается как сумма двух предыдущих.

Первые два числа равны 1, затем 2(1+1), затем 3(1+2), 5(2+3) и так далее: 1, 1, 2, 3, 5, 8, 13, 21....

Числа Фибоначчи тесно связаны с золотым сечением и множеством природных явлений вокруг нас.

Напишите функцию fib(n), которая возвращает n-е число Фибоначчи. Пример работы:

function fib(n) { /\* ваш код \*/ }

alert( fib(3) ); // 2

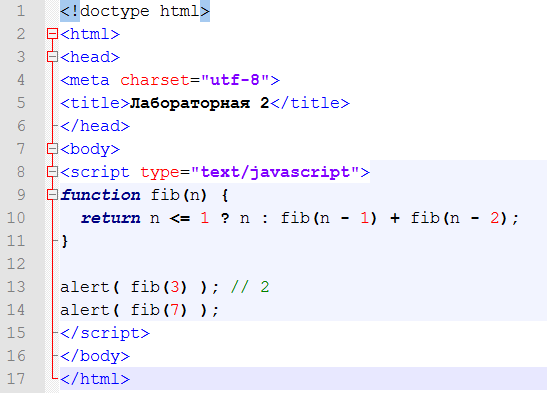
alert( fib(7) ); // 13

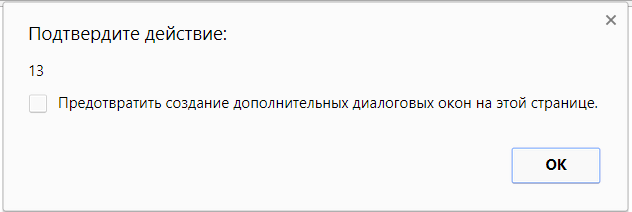
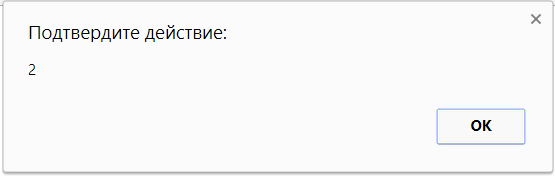
alert( fib(77)); // 5527939700884757

Все запуски функций из примера выше должны срабатывать быстро.

1.Вычисление рекурсией (медленное)

Решение по формуле, используя рекурсию:

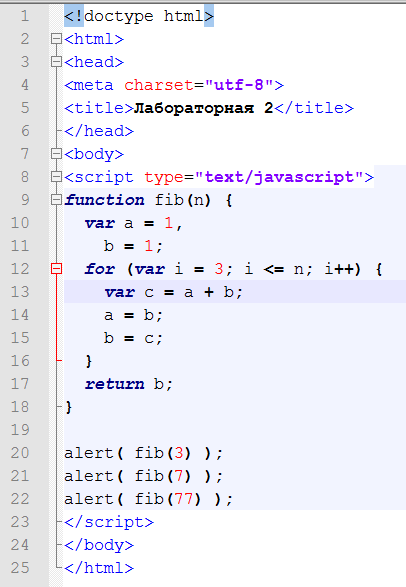
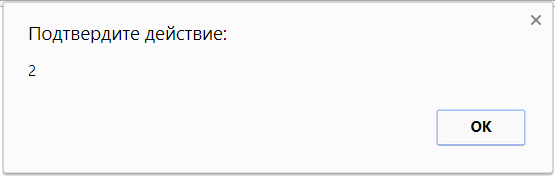
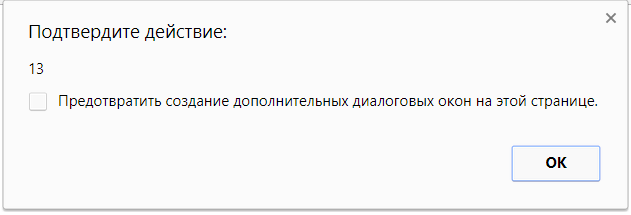
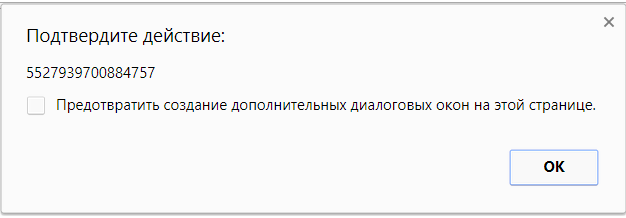




При больших значениях n оно будет работать очень медленно. Например, fib(77) уже будет вычисляться очень долго.

Это потому, что функция порождает обширное дерево вложенных вызовов. При этом ряд значений вычисляется много раз.

#### 2. Код для вычисления в цикле

Цикл здесь начинается с i=3, так как первое и второе числа Фибоначчи заранее записаны в переменные a=1, b=1.