

Лабораторная работа 6

Циклы

Цель этой лабораторной работы — изучить понятие цикла и продемонстрировать умение записывать циклы в языке JavaScript.

ВНИМАНИЕ! Файлы называть `loop-01.js` и т. д. Массивы, библиотечные функции (кроме `console.log`), не использовать.

1. Напишите программу, которая напечатает 100 строк Hello world! при помощи цикла `for`.
2. Напишите программу, которая напечатает 100 строк Hello world! при помощи цикла `do ... while`.

В заданиях 3–6 необходимо экспортировать вашу функцию:

`module.exports = pyramid;` и т.п.

3. Напишите функцию `gcd(m, n)`, которая возвращает наибольший общий делитель чисел m и n . Например, `gcd(24, 18)` должно быть равно 6.
4. Напишите функцию `pyramid(n)`, которая возвращает текстовую пирамиду из решеток и пробелов высоты n ($1 \leq n \leq 20$), например для $n = 3$ функция должна вернуть три сцепленных строки:

```
#
###
#####
```

т. е. строку `#\n ###\n#####`.

5. Для быстрой, без связи с банком, проверки правильности ввода номера кредитной карты используется алгоритм Х. П. Луна: умножим, двигаясь справа налево, каждую вторую цифру номера на 2. Сложим все *цифры* полученных чисел (внимание, не сами числа!). Теперь прибавим к ним сумму остальных цифр. Если полученная общая сумма не делится на 10, номер неправильный.

Пример: номер 4372 2822 4431 0005 верный: удвоения подчеркнутых цифр равны 8, 14, 4, 4, 8, 6, 0, 0, их цифры в сумме дают $8 + (1 + 4) + 4 + 4 + 8 + 6 + 0 + 0 = 35$, сумма неподчеркнутых цифр номера равна $3 + 2 + 8 + 2 + 4 + 1 + 0 + 5 = 25$, а $35 + 25 = 60$.

Напишите функцию `checkCardNumber(nstr)`, которая возвращает `true`, если номер, записанный в строке `nstr` проходит проверку по алгоритму Луна, и `false` в противном случае. Проверки будут производиться на строках, содержащих от 13 до 16 цифр.

6. К волшебному пределу $e = \lim_{n \rightarrow \infty} (1 + \frac{1}{n})^n$ можно приблизиться по-другому: ряд

$$1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots \quad (*)$$

сходится к e^x при любых x .

Напишите функцию `expDiff(x)`, которая получает вещественное x , суммирует только те слагаемые ряда (*), которые по модулю не меньше 0.0001 (модули слагаемых монотонно убывают), и выводит модуль разности между значением библиотечной функции `Math.exp(x)` и полученным значением суммы.