Лекция 3
Функции. Простейшие алгоритмы.

Разминка

До 14:02

Регистры

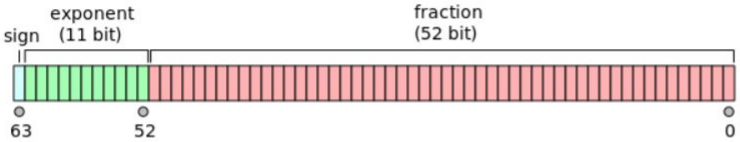
Регистр — устройство для записи, хранения и считывания *n*-разрядных двоичных данных и выполнения других операций над ними. Регистр может быть реализован посредством набора RS-триггеров или D-триггеров. Размер регистра определяет длину **машинного слова** процессора.
Хранение целых чисел в регистре тривиально – в регистре хранится двоичное представление числа (+бит знака)

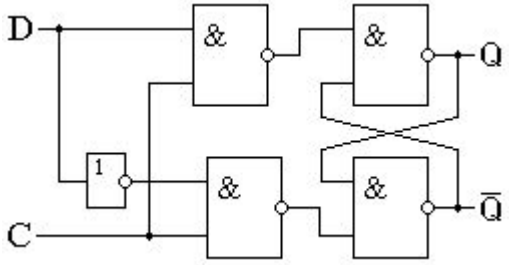
C	D	Q(t)	Q(t+1)	Пояснения
0	x	0	0	Режим хранения информации
0	x	1	1	
1	0	x	0	Режим записи информации
1	1	x	1	

Как хранятся вещественные числа?

Любое число можно представить как $(-1)^S \times M \times B^E$

S – знак, M – мантисса (англ. fraction), E – порядок, B – основание



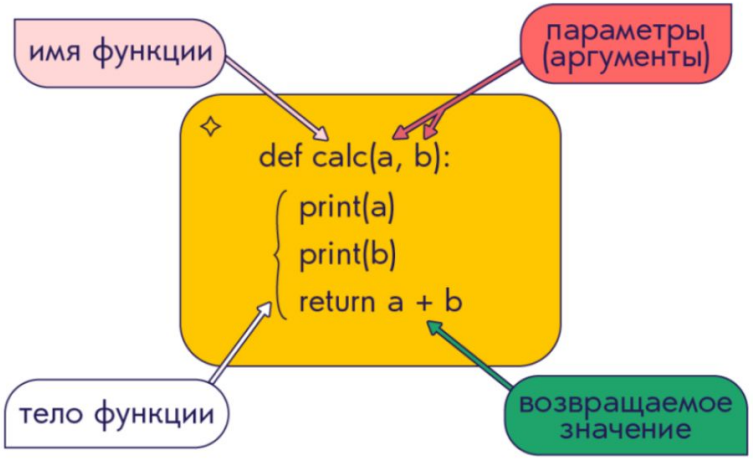


D-триггер

Функции

Функция в Python - объект, принимающий аргументы, исполняющая некоторый набор команд.

Иногда различают **функции** (подпрограммы, выполняющие какие-либо операции и возвращающие значение) и **процедуры** (подпрограммы, выполняющие какие-либо операции без возврата значения, например random.shaffle()). В Python разграничения нет, процедуры – это функции, в которых нет return. Следует также отличать от **методы** – функции или процедуры, вызываемые от объектов [например A.pop()]. Мы рассмотрим их позднее.




```python
var1 = 2213 # глобальная переменная

# глобальная функция, использующая глобальную переменную
def func_one():
    print(var1)

# глобальная функция, использующая локальную переменную
def func_two():
    var1 = "Local value" # создание локальной переменной
    print(var1)

# функция, изменяющая значение глобальной переменной
def func_three():
    global var1
    var1 = "Local to global"
```

Декомпозиция алгоритмов. Стек вызовов. Рекурсия.

Хорошо известно, что перед тем, чтобы начать решать задачу, надо разбить ее на подзадачи. Это действие получило название **декомпозиции**. Декомпозиция алгоритма на частные алгоритмы и модули проводится для сокращения сроков создания программ и упрощения проверки работоспособности.

Рекурсия – вызов функции из нее же самой. Пример: вычисление факториала:
Любой рекурсивный алгоритм может быть реализован через цикл for.
Но это не всегда удобно.

```
def fac(n):
    if n == 1:
        return 1
    return fac(n - 1) * n


print(fac(5))
```

Алгоритм Евклида

1. Большее число делим на меньшее.
2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла).
3. Если есть остаток, то большее число заменяем на остаток от деления.
4. Переходим к пункту 1.

```
a = int(input())
b = int(input())

while a != 0 and b != 0:
    if a > b:
        a = a % b
    else:
        b = b % a

print(a + b)
```

Интересные ссылки

https://tirinox.ru/float-python/

https://proglib.io/p/samouchitel-po-python-dlya-nachinayushchih-chast-11-funkcii-s-pozicionnymi-i-imenovannymi-argumentami-2023-01-09

https://github.com/redb0/python-bp/blob/master/python_pd/badcode.ipynb

https://techrocks.ru/2020/02/16/recursion-and-the-call-stack-explained/