

## Лабораторная работа 0 — Setup

Для выполнения задач мы будем использовать Python 3. Необязательно иметь установленный Python на своем компьютере, однако это желательно для дальнейшей работы. Вы можете пользоваться платформой Trinket и/или установить себе на компьютер PyCharm.

Официальный сайт Python, где можно скачать интерпретатор (Python 3): <https://www.python.org/>

Официальная документация по Python: <https://docs.python.org/3/>

Веб-сервис, позволяющий исполнять программы на Python прямо в браузере: <https://trinket.io/python/>

Среда для написания программ PyCharm Educational Edition или PyCharm Community Edition: <https://www.jetbrains.com/pycharm-educational/> или <https://www.jetbrains.com/pycharm/>

Интерактивный учебник языка Python (на русском языке): <http://pythontutor.ru/>

Книга A Byte of Python: <https://python.swaroopch.com/>

### 1. Определение функций

Функции в Python обычно определяются внутри файла. Создайте файл с именем `myfirstscript.py` и реализуйте следующую функцию:

$$f(x) = \cos(x) \exp(x)$$

Запустите свой скрипт в командной строке.

В Python несколько функций могут быть определены в одном файле, а имя файла не зависит от имен функций, используемых в нем.

### 2. Визуализация данных

Каждый Python-файл представляет собой скрипт, который может быть запущен позднее. В одном скрипте может содержаться несколько функций и/или других вычислений.

Для построения графиков можно использовать модуль `matplotlib.pyplot`.

- (a) В том же скрипте `myfirstscript.py` постройте график функции  $f$  на интервале  $[-2\pi, 2\pi]$ . *Подсказка:* в модуле `numpy` есть специальная переменная для числа  $\pi$  — `numpy.pi`.
- (b) Сохраните полученный график как PNG-файл.

### 3. Генерация случайных чисел

Случайные числа играют важную роль в вероятностной робототехнике, поэтому полезно знать, какие возможности для работы со случайными величинами предоставляет Python. *Подсказка:* используйте `numpy`.

- (a) Создайте вектор, содержащий 100 000 случайных величин, которые нормально распределены с математическим ожиданием 5.0 и среднеквадратичным отклонением 2.0.
- (b) Создайте вектор, содержащий 100 000 случайных величин, равномерно распределенных на интервале от 0 до 10.
- (c) Для каждого вектора вычислите математическое ожидание и среднеквадратичное отклонение. Совпадает ли полученный результат с вашим ожиданием?
- (d) Постройте гистограммы распределения случайных величин и функции плотности вероятности. *Подсказка:* используйте команду `hist`, подробнее см. справку `help(matplotlib.pyplot.hist)`.
- (e) Измените скрипт так, чтобы каждый раз, когда вы его запускаете, распределения получались одни и те же.

### 4. Работа со списками (Optional)

Напишите программу, которая по заданному  $n$  выводит таблицу размером  $n \times n$ , заполненную числами от 1 до  $n^2$  по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь  $n = 5$ ):

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9