

Введение в Python

Инструменты разработки, типы данных

Амбросимова Виктория, 28.02.2022



Установка python

- Узнаём, 32-битная или 64-битная система на компьютере
 - Нажимаем «Пуск»
 - В поле поиска вводим «Система»
 - Выбираем «Сведения о системе»
 - Смотрим: компьютер на базе x64 — 64-битная,
x86 — 32-битная
- Переходим на <https://www.python.org>



Установка python

- Делаем зависание мышью в строке меню на «Downloads»
- В правой части развернувшегося окошка будет кнопка скачивания на подходящую вам версию Python

Если так не получилось, то

- Нажимаем на «Downloads» —> «Windows»
- В Stable Releases выбираем Windows installer (xx-bit)

Вместо xx должна быть указана 32 или 64-битная система



Установка python

Замечание: все советы написаны для windows старше 7-й версии. Если у вас более старая, то за руководством по установке обращайтесь ко мне в чате

- После загрузки дистрибутива устанавливаем его двойным щелчком и следуем инструкциям
- На экране Setup пролистываем вниз, видим «add python xx.xx to PATH» (или добавить питон к системной переменной), ставим галочку
- После установки возможно предложение узнать больше о Python. Но установка на этом закончена



Установка редактора кода

Редактор кода выбирается из личных предпочтений. Word и подобные использовать не рекомендуется, потому что они хранят текст с форматированием, шрифтами и прочими «радостями», а нам нужен код в виде простого, так сказать, текста.

К тому же, специализированные редакторы кода содержат много полезных функций и возможностей.

Подойдет Gedit, Sublime Text, Atom, PyCharm, Spyder



Anaconda

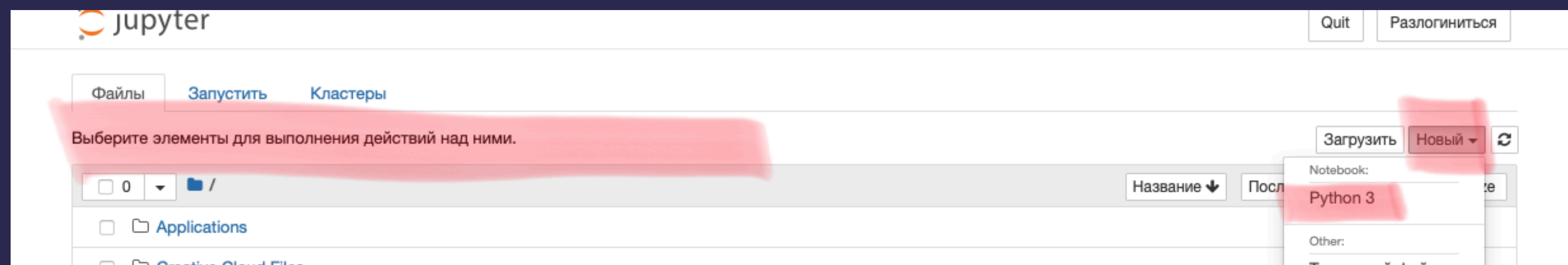
Anaconda — это пакет библиотек python (и, в принципе, версия самого python). В anaconda.navigator есть возможность установки PyCharm, Spyder, JupyterLab, открытия jupyterNotebook и многого другого, необходимого для анализа данных.

- Чтобы установить анаконду, переходим по ссылке <https://www.anaconda.com/products/individual>
- Выбираем свою операционную систему
- Скачиваем дистрибутив, соответствующий разрядности вашей системы
- А дальше, как обычно. Готово!



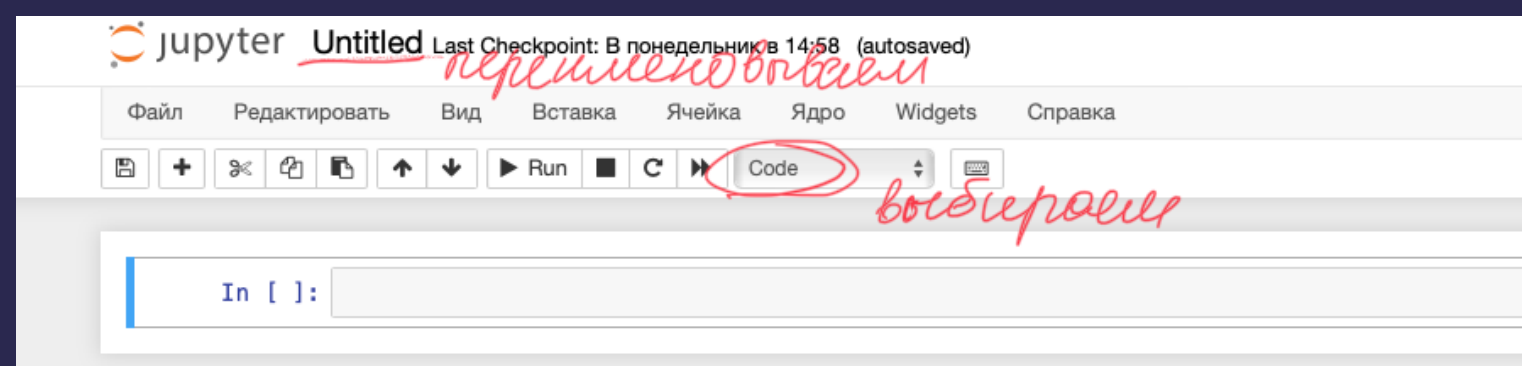
Выполнение кода

- Выполнять операции на языке Python можно в командной строке, набрав python3 (выход ctrl+d)
- Можно в **jupyterNotebook** (либо в colab)
- JupyterNotebook запускаем из Anaconda
- В браузере выбираем, что указано на картинке

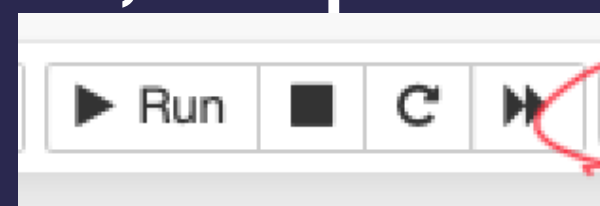


Выполнение кода

- На открывшейся странице проверяем необходимые настройки и даем новое осмысленное имя файлу



- В поле In []: записываем команды на ЯП, для запуска, остановки, перезапуска кода в поле нажимаем одну из клавиш



- Когда работу завершили, сохранили свой ноутбук, закрыли вкладку браузера, на рабочем столе в черном окошке командной строки дважды нажимаем ctrl+c



Введение в Python

Говорят, что человек знает ЯП, если может на нем вывести фразу «Hello, World!», потому что для этого необходимо знать основные требования оформления кода, чтобы компьютер мог понять и выполнить задачу.

В Python таких требований немного: необходимо знать, где прописываются импортированные файлы (если вы их импортируете), блоки действий выделяются отступами, одно действие - одна строка. В принципе, всё. По ходу дела вспомню еще какие-нибудь (если вспомню ;))

Стиль оформления кода — PEP8



Ввод и вывод

- Команда вывода данных выглядит так:
`print(то_что_хотим_вывести)`

```
In [1]: print ("hello, world")  
hello, world
```

- Команда ввода данных с клавиатуры:
`a = input()`, где `a` — имя участка памяти компьютера, в котором будут храниться полученные данные (иначе, переменная)

```
In [2]: a = input()  
print(a)  
wed  
wed
```



Типы данных

Типы данных в Python делятся на две группы

- **Неизменяемые**
 - Числовые (`int`, `float`)
 - Логические (`bool`)
 - Пустой (`None`)
 - Символьные строки (`class 'str'`)
 - Кортежи (`tuple`)
- **Изменяемые**
 - Списки (`list`)
 - Множества (`set`)
 - Словари (`dict`)

Данные, определенные пользователем, могут быть определены как те, так и другие



Переменные и тип данных

Чаще всего, в программировании приходится иметь дело с переменными, т.к. они позволяют различать, где какие данные мы храним, следовательно, так проще этими данными манипулировать.

В большинстве ЯП программирования необходимо заранее и явно указывать, к какому типу данных наша переменная относится. Python в ряде случаев умеет сам определять тип и не требует его явного указания. Если вы хотите явно указать, какого типа будет переменная, то нужно выполнить приведение типов. При этом, если вы захотите перезаписать в нее данные, а они будут нового типа, ошибки не возникнет



Переменные и тип данных

Вот пример такой возможности (`type(переменная)` — команда определения типа переменной),

```
In [5]: a = 'str'
        print(type(a))
        a = 7
        print(type(a))

<class 'str'>
<class 'int'>
```

А вот с явным указанием типа

```
In [10]: a = int(9)
         print(type(a))
         a = 'rer'
         print(type(a))

<class 'int'>
<class 'str'>
```



Тип данных `int`

Целочисленный тип данных `int` в питоне не имеет строгой размерности в битах, потому что для очень больших чисел в нем предусмотрена автоматическая реализация длинной арифметики

Для `int` доступны операции сложения, умножения, вычитания, возведения в степень, целочисленного деления (соответственно, `+`, `*`, `-`, `**`, `//`)

Операции деления и извлечение корня n -степени переводят `int` \rightarrow `float`



Тип данных int

Посмотрим

```
In [11]: a = 8  
         b = 5  
         print('type_a', type(a), ' type_b', type(b))  
         print(a/b, ' ', type(a/b))  
  
type_a <class 'int'>  type_b <class 'int'>  
1.6    <class 'float'>
```

Первоначально переменные a и b были восприняты как целочисленные, но при делении их результат получил тип «с плавающей точкой». Python в этом случае преобразовал только тип ответа, а типы самих переменных оставил прежним



Тип данных float

Float — это тип данных, соответствующий дробям (бесконечным и конечным). Но тут очень длинный «хвост» дроби сохранить не получится, потому что на хранение float отводится всего 64 бита, следовательно, заявлять, что дробь может быть бесконечной, строго говоря, в питоне мы не можем.

Для данных типа float доступны те же операции, что для int

При выводе дробного числа иногда интересно указать количество цифр после запятой. Это можно сделать так

```
In [14]: format(0.1+0.1+0.1 - 0.3, '.34f')  
Out[14]: '0.000000000000000000555111512312578270'
```



Тип данных float

Для float есть команда округления числа **round(число)**

```
In [17]: print(round(0.479), ' ', round(0.78))  
0      1
```

Она округляет по правилам математики. **Не путать с целочисленным делением!!**

```
In [18]: print(round(0.479), ' ', round(0.78))  
          print(7 // 2)  
0      1  
3
```

Еще существует операция взятия остатка от деления. Если с int всё понятно, то у float не так всё просто. Чтобы понять принцип необходимо обладать продвинутыми знаниями математики

```
In [22]: print(7.1 % 4.3)  
          print(7 % 4)  
2.8  
3
```



None и bool

None — специальное слово в питоне, которое позволяет сделать любую переменную пустой. Это совсем не одно и то же с обнулением. Не путать

Bool принимает всего одно из двух значений **true** или **false**

```
In [23]: a = 5 > 6
          print(type(a))
          a = None
          print(type(a))
          a = 'None'
          print(type(a))

<class 'bool'>
<class 'NoneType'>
<class 'str'>
```



Тип данных `str` и доступные для него функции рассмотрим отдельно на следующих занятиях

