

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Ю.А. Иванова, В.Ю. Полищук, Е.С. Попова

Программирование на Python

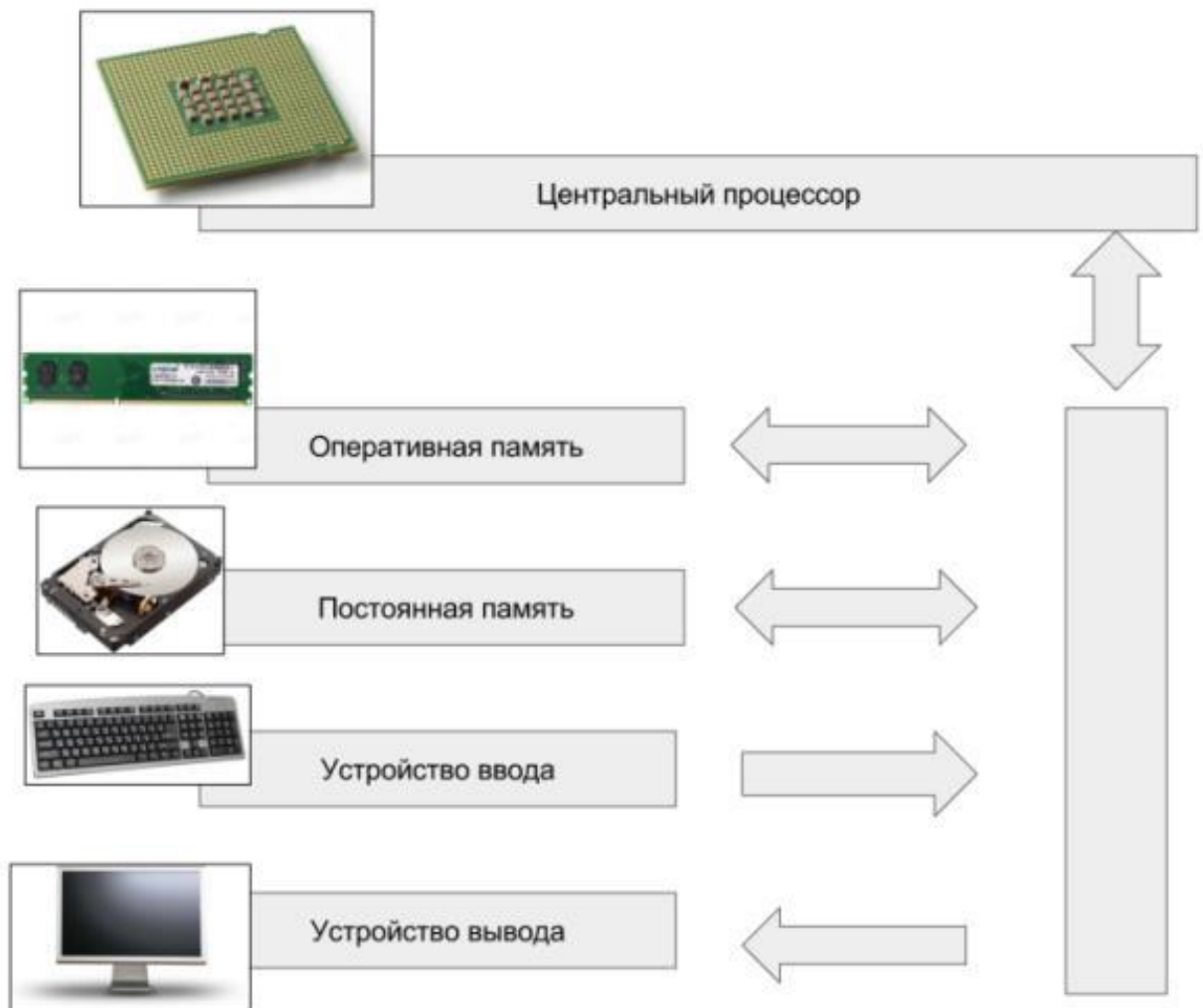
Методические указания для выполнения лабораторных работ

*Рекомендовано в качестве учебного пособия
Редакционно-издательским советом
Томского политехнического университета*

Томский политехнический университет

2020

Инструментом программиста является компьютер, рассмотрим его устройство. Все вычисления в компьютере производятся центральным процессором. Файлы с программами хранятся в постоянной памяти (на жестком диске), а в момент выполнения загружаются во временную (оперативную) память. Ввод информации в компьютер осуществляется с помощью клавиатуры (устройства ввода), а вывод – с помощью монитора (устройства вывода).



Компьютер хранит и обрабатывает данные в бинарном (машинном) коде, представленном 1 и 0. Первые программы писались на перфокартах – прямоугольниках из тонкого картона, представляющих информацию наличием или отсутствием отверстий в определённых позициях карты.









Писать программы вида 1010101010010101010 для человека является трудоемкой задачей, поэтому со временем появились программы-трансляторы с языка программирования, понятного человеку, на машинный язык, понятный компьютеру.

Языки программирования, которые приближены к машинному уровню, называют языками низкого уровня (например, язык ассемблера). Другой вид языков – языки высокого уровня (например, Python, Java, C#) – больше приближены к мышлению человека.



Как правило, языки программирования создавались под конкретную задачу. На сегодняшний день существует множество различных языков программирования, наиболее популярные из которых представлены в таблице.

	Язык ассемблера	Микрокомпьютеры с очень ограниченными ресурсами.		50-ые
	Fortran	Математические расчеты.		
	Basic	Языки для обучения программированию.		60-70-ые
	Pascal			
	C	Системное программирование (драйвера и пр.)		ОС UNIX
	C++	Включает все возможности языка C, реализует ООП подход.	Потребность в больших программах - появился подход ООП.	80-ые
	Java	Крупные программы для бизнеса (ООП). Сложно написать плохую программу.	Потребность в программистах и переносимости. Автоматизировать кофемашину.	90-ые
	Python	Автоматизация рутинной деятельности (быстро), обучение программированию.		Персональные ПК, Интернет
	PHP	Разработка динамических сайтов.		
	C# (.NET)	Крупные программы для бизнеса (ООП). Много общего с Java. Зависимость от продуктов Microsoft.	Обобщение и объединение: собрать всё лучшее, что было до этого.	2000-ые

Началом общения с компьютером послужил машинный код. Затем в 50-ые годы двадцатого века появился низкоуровневый язык ассемблера, наиболее приближенный к машинному уровню. Он привязан к процессору, поэтому его изучение равносильно изучению архитектуры процессора. На языке ассемблера пишут программы и сегодня, он незаменим в случае небольших устройств (микроконтроллеров), обладающих очень ограниченными ресурсами памяти.

Следующий этап (80-ые годы) характеризуется появлением объектно-ориентированного программирования (ООП), которое должно было упростить создание крупных промышленных программ. Появляется ученый – Б. Страуструп, которому недостаточно было возможностей языка C, поэтому он расширяет этот язык путем добавления ООП. Новый язык получил название C++.

В 90-ые годы появляются персональные компьютеры и сеть Интернет, потому требуются новые технологии и языки программирования. Язык Java создавался с оглядкой на C++ и с перспективой развития сети Интернет.

Примерно в одно время с Java появляется Python. Разработчик языка – математик Гвидо ван Россум занимался долгое время разработкой языка ABC, предназначенного для обучения программированию. С ростом сети Интернет потребовалось создавать динамические сайты – появился серверный язык программирования PHP.

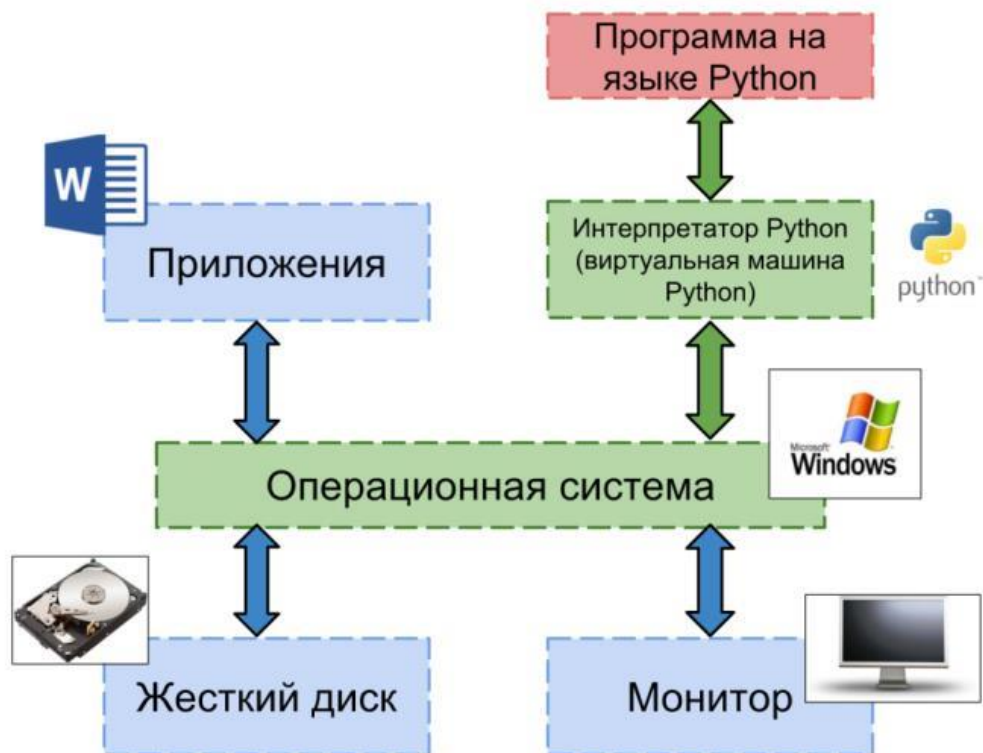
В 2000-ые годы наблюдается тенденция объединения технологий вокруг крупных корпораций. В это время получает развитие язык C# на платформе .NET.

Язык программирования Python

Питон является одним из наиболее широко используемых языков программирования в следующих областях:

1. Системное программирование.
2. Разработка программ с графическим интерфейсом.
3. Разработка динамических веб-сайтов.
4. Интеграция компонентов.
5. Разработка программ для работы с базами данных.
6. Быстрое создание прототипов.
7. Разработка программ для анализа данных.
8. Разработка программ для научных вычислений.
9. Разработка игр.

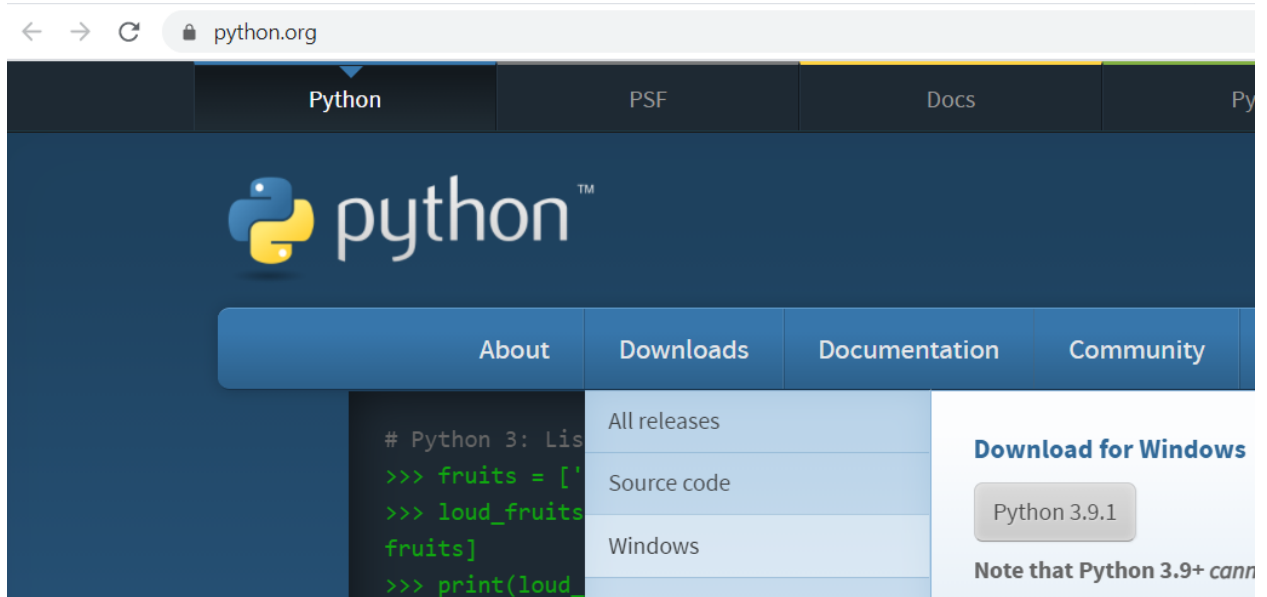
Выполнение программ осуществляется операционной системой Windows, Linux и пр.). В задачи операционной системы входит распределение ресурсов (оперативной памяти и пр.) для программы, запрет или разрешение на доступ к устройствам ввода/вывода и. т. д.



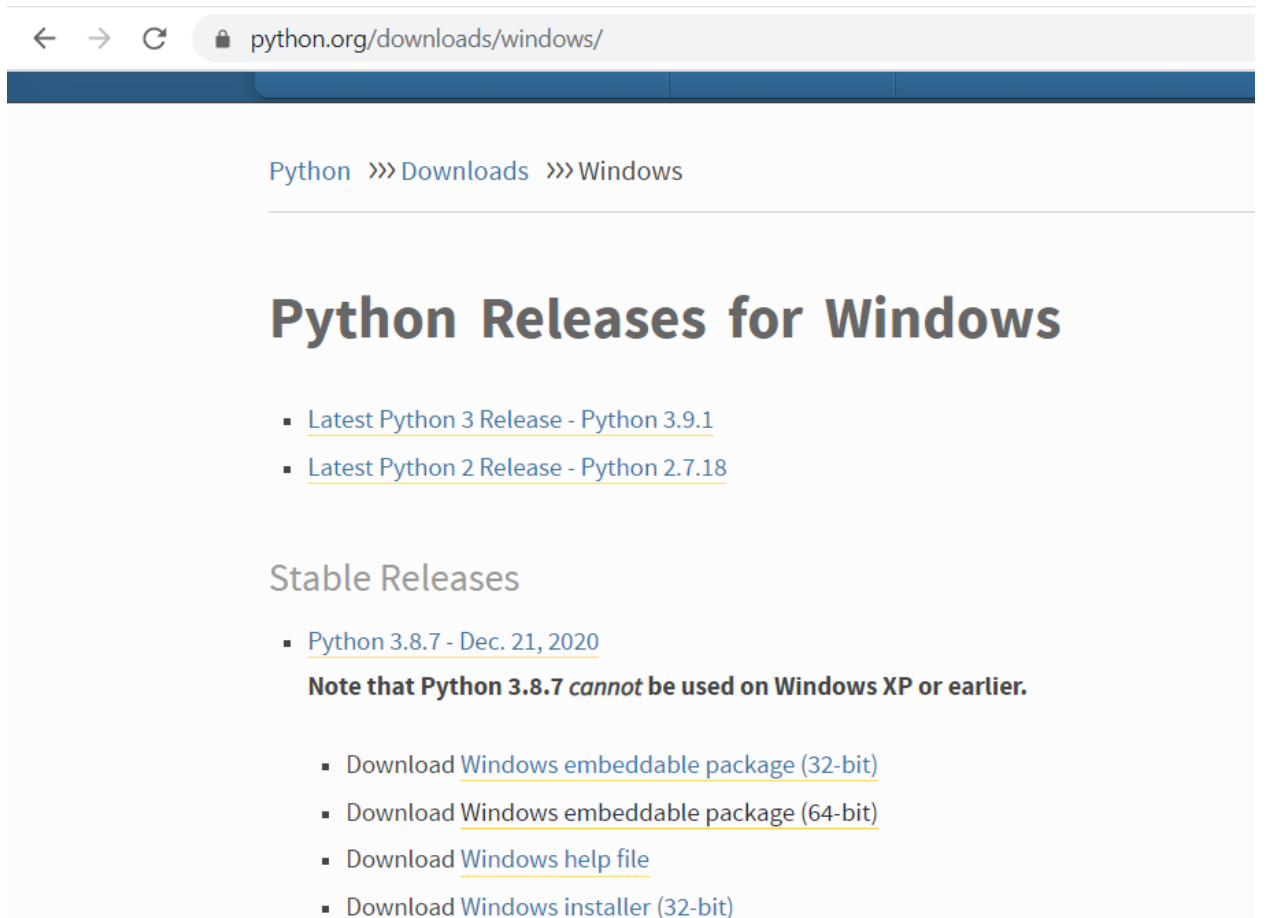
Для запуска программ на языке Python необходима программа-интерпретатор (виртуальная машина) Python. Данная программа скрывает от Python-программиста все особенности операционной системы, поэтому, написав программу на Python в системе Windows, ее можно запустить, например, в GNU/Linux и получить такой же результат.

Установка Python

Для установки python необходимо перейти по ссылке: python.org и выбрать подходящую версию для вашей операционной системы.

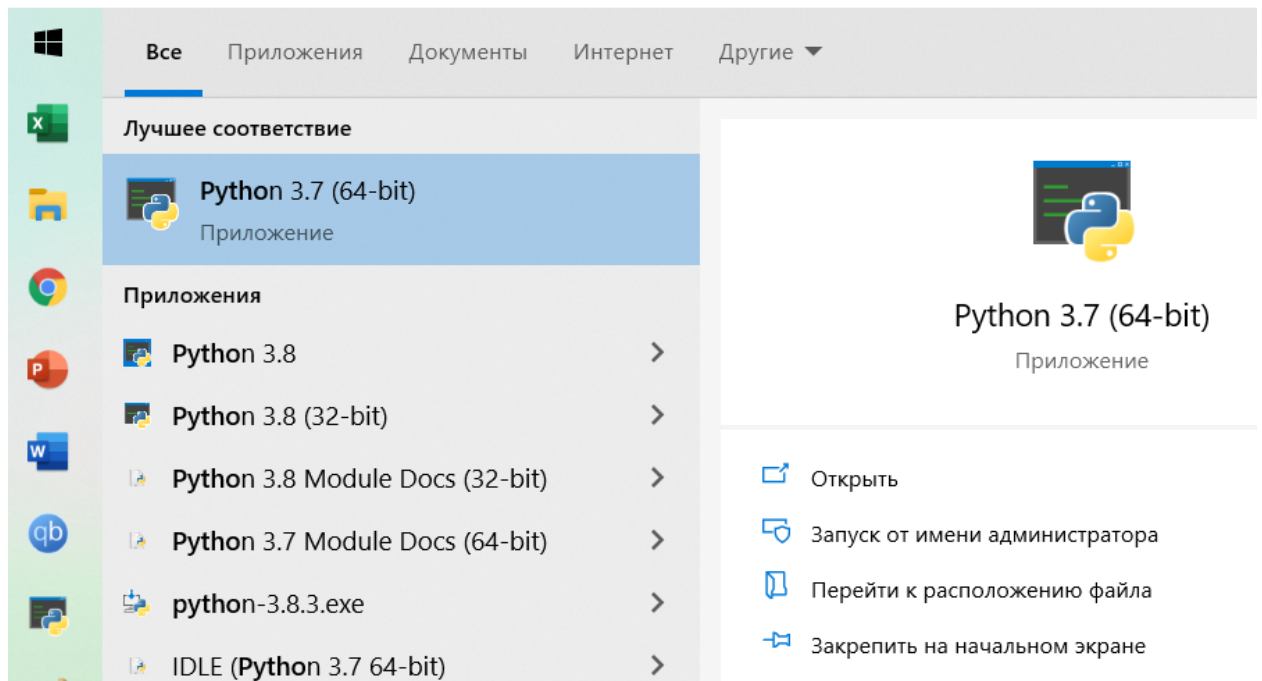


Можно выбрать одну из последних версий python3:



После загрузки устанавливаем выбранную версию python.

Для запуска python переходим в «Пуск» → «python...», и запускаем выбранную версию (или в командной строке «Пуск» → «cmd»→ «ру»):



После чего откроется окно интерпретатора python. Для первой программы вызовем функцию `print(«hello!»)`:

```
Python 3.7 (64-bit)
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello!")
hello!
>>> _
```

Знак «>>>» означает ожидание интерпретатора ввода команды. Вывод результата приводится без этого знака.

Установка интерактивной среды разработки Jupyter

IPython представляет собой мощный инструмент для работы с языком Python. Базовые компоненты IPython – это интерактивная оболочка для с широким набором возможностей и ядро для Jupyter. Jupyter notebook является графической веб-оболочкой для IPython, которая расширяет идею консольного подхода к интерактивным вычислениям.

Основные отличительные особенности данной платформы – это комплексная интроспекция объектов, сохранение истории ввода на протяжении всех сеансов, кэширование выходных результатов, расширяемая система “магических” команд, логирование сессии, дополнительный

командный синтаксис, подсветка кода, доступ к системной оболочке, стыковка с pdb отладчиком и Python профайлером.

IPython позволяет подключаться множеству клиентов к одному вычислительному ядру и, благодаря своей архитектуре, может работать в параллельном кластере.

В Jupyter notebook вы можете разрабатывать, документировать и выполнять приложения на языке Python, он состоит из двух компонентов: веб-приложение, запускаемое в браузере, и ноутбуки – файлы, в которых можно работать с исходным кодом программы, запускать его, вводить и выводить данные и т.п.

Веб приложение позволяет:

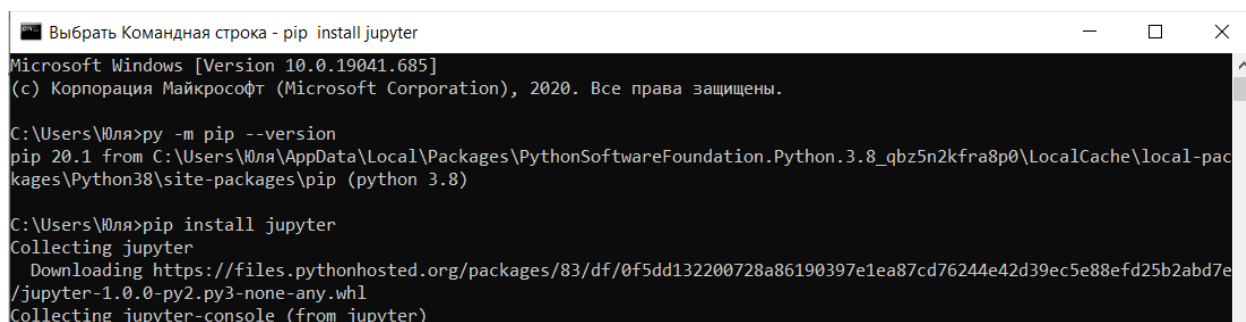
- редактировать Python код в браузере, с подсветкой синтаксиса, автоотступами и автодополнением;
- запускать код в браузере;
- отображать результаты вычислений с медиа представлением (схемы, графики);
- работать с языком разметки Markdown и LaTeX.

Ноутбуки – это файлы, в которых сохраняются исходный код, входные и выходные данные, полученные в рамках сессии. Фактически, он является записью вашей работы, но при этом позволяет заново выполнить код, присутствующий на нем. Ноутбуки можно экспортировать в форматы PDF, HTML.

pip – система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python.

«Пуск» → «cmd». Проверить наличие и версию pip в Windows можно с помощью команды «ру -pip – version».

Установка среды Jupyter осуществляется командой: «pip install jupyter»:



```
Выбрать Командная строка - pip install jupyter
Microsoft Windows [Version 10.0.19041.685]
(c) Корпорация Майкрософт (Microsoft Corporation), 2020. Все права защищены.

C:\Users\Юля>py -m pip --version
pip 20.1 from C:\Users\Юля\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-pac
kages\Python38\site-packages\pip (python 3.8)

C:\Users\Юля>pip install jupyter
Collecting jupyter
  Downloading https://files.pythonhosted.org/packages/83/df/0f5dd132200728a86190397e1ea87cd76244e42d39ec5e88efd25b2abd7e
/jupyter-1.0.0-py2.py3-none-any.whl
Collecting jupyter-console (from jupyter)
```

Принцип запуска и работы со средой можно посмотреть в методических указаниях «instruction_JN.pdf».

Начало работы с Google Colab

Лабораторные работы можно выполнять на удаленном сервере в лаборатории Гугл (Google Colaboratory), расположенном по следующей ссылке: <https://colab.research.google.com/notebooks/welcome> (рис.1).

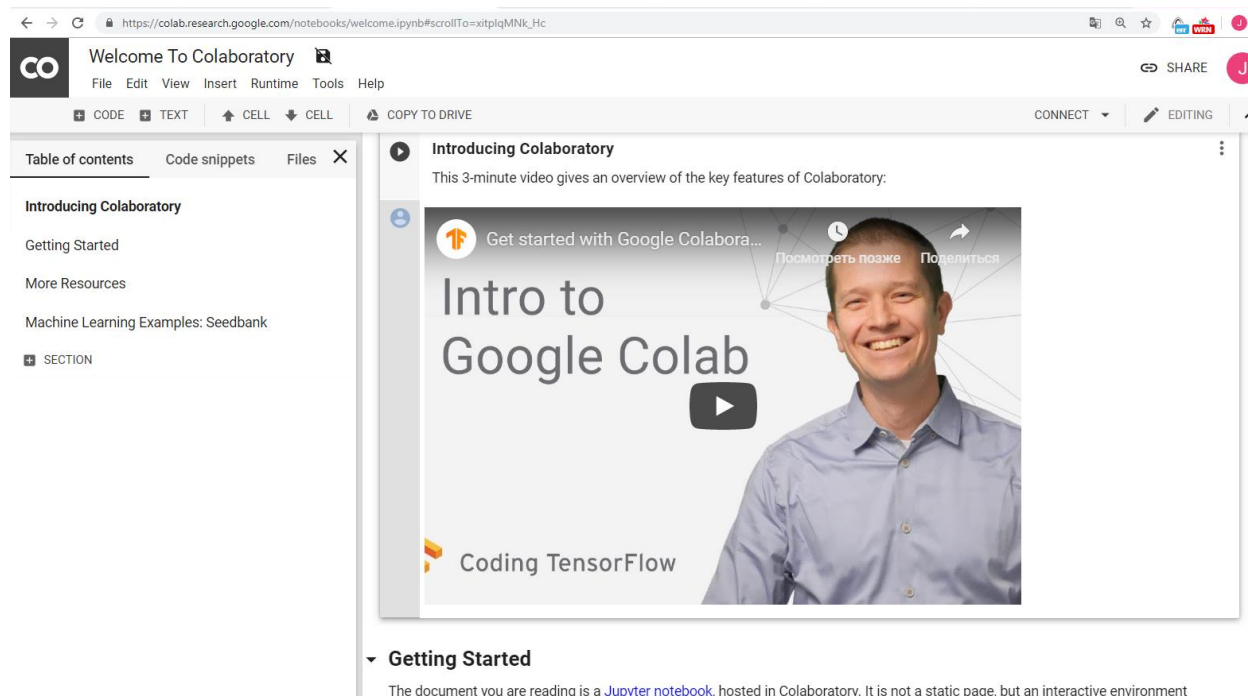


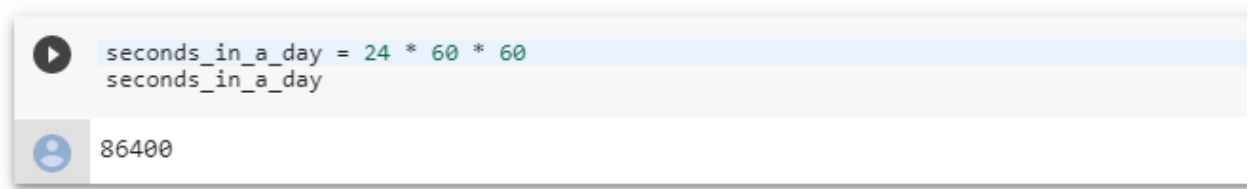
Рис. 1. Начальная страница лаборатории Гугл

Работа будет вестись в блокноте Jupyter (<https://jupyter.org/>), размещенным в лаборатории. Блокнот Jupyter представляет собой не статическую страницу, а интерактивную среду, которая позволяет писать и выполнять код на Python и других языках.

Отличительной особенностью написания кода в блокноте Jupyter является то, что код разбивается на ячейки (cell), каждая из которых может быть выполнена отдельно.

Файлы, создаваемые и редактируемые в блокноте Jupyter имеют расширение **.ipynb**.

Для начала работы можно запустить код, представленный на первой странице лаборатории (рис.2). Для этого нужно нажать на стрелку в соответствующей ячейке или выделить ячейку и нажать **ctrl+enter**. Для выполнения кода последовательно во всех ячейках можно воспользоваться сочетанием клавиш **ctrl+F9**.



```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

Рис. 2. Пример кода на языке Python в блокноте Jupyter

В данной программе производится подсчет количества секунд в сутках. Переменной **seconds_in_a_day** присваивается значение, равное $24(\text{часа}) \cdot 60(\text{минут}) \cdot 60(\text{секунд})$. Вторая строка предназначена для вывода значения, хранящегося в переменной **seconds_in_a_day**.

Для создания нового файла вашей программы нужно перейти в меню File->New Python 3 notebook.

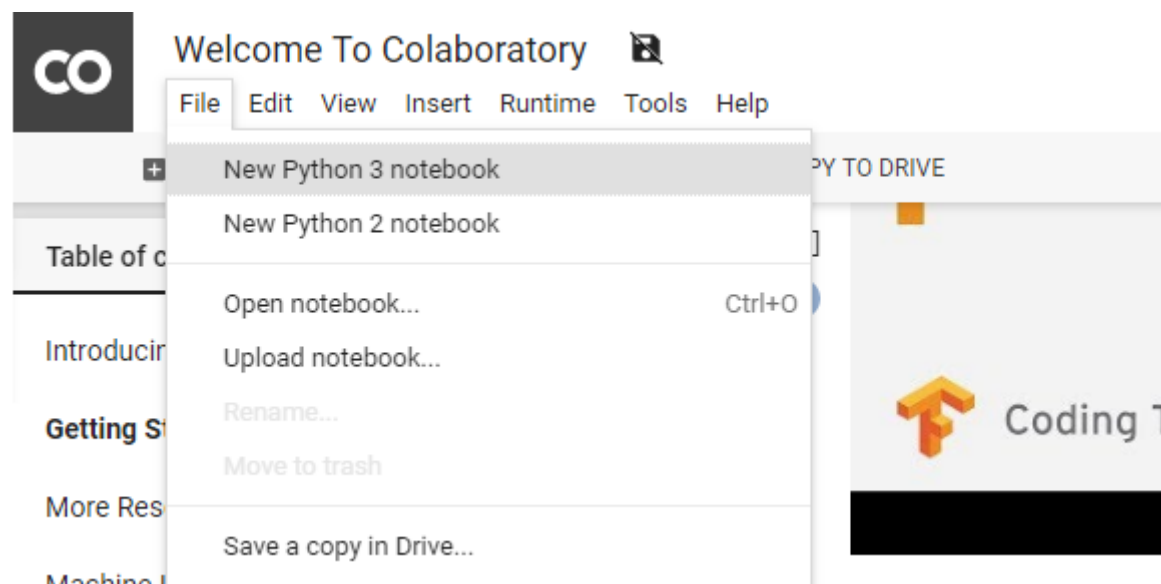


Рис. 3. Создание нового программного файла .ipynb

В новой вкладке откроется файл Untitled2.ipynb (рис. 4), с которым вы будете работать далее.

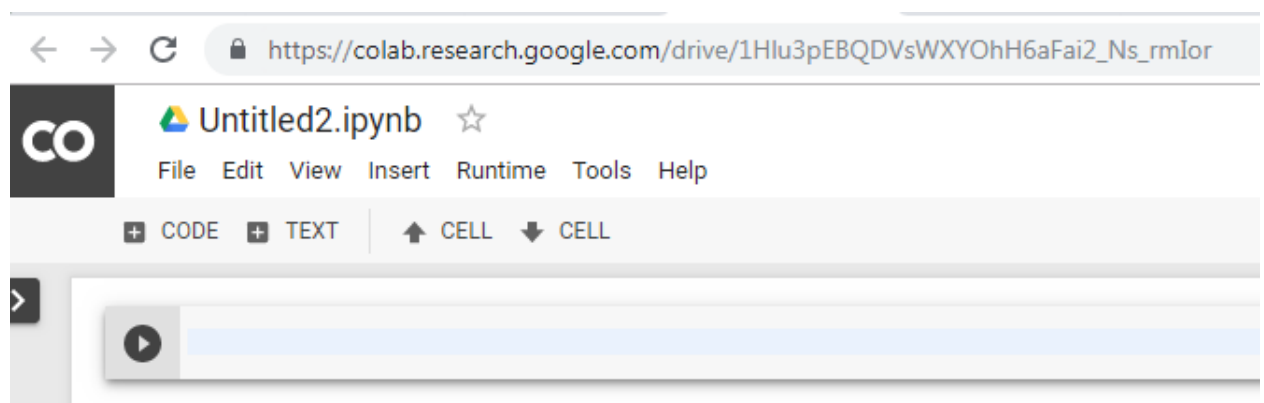
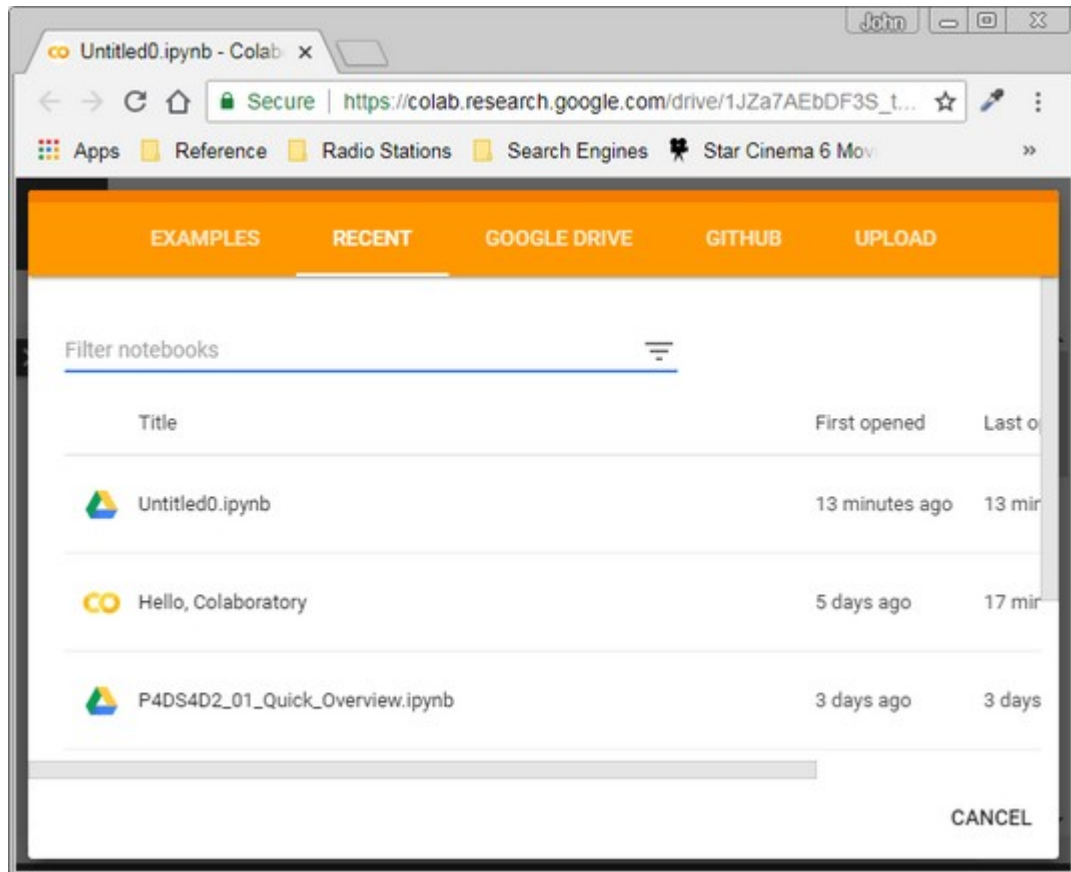


Рис. 4. Пример нового программного файла

Переименовать файл можно выбрав в меню File→ Rename.

Открытие ранее созданных проектов осуществляется выбором File → Open Notebook. Появится диалоговое окно, отображающее недавно открытые файлы.



Для того чтобы сохранить файл с новым именем на жесткий диск необходимо в меню выбрать File->download .ipynb. После выполнения лабораторной работы нужно будет сохранять файлы на диске.

Для добавления новой ячейки выберите Insert->Code cell (рис.5).

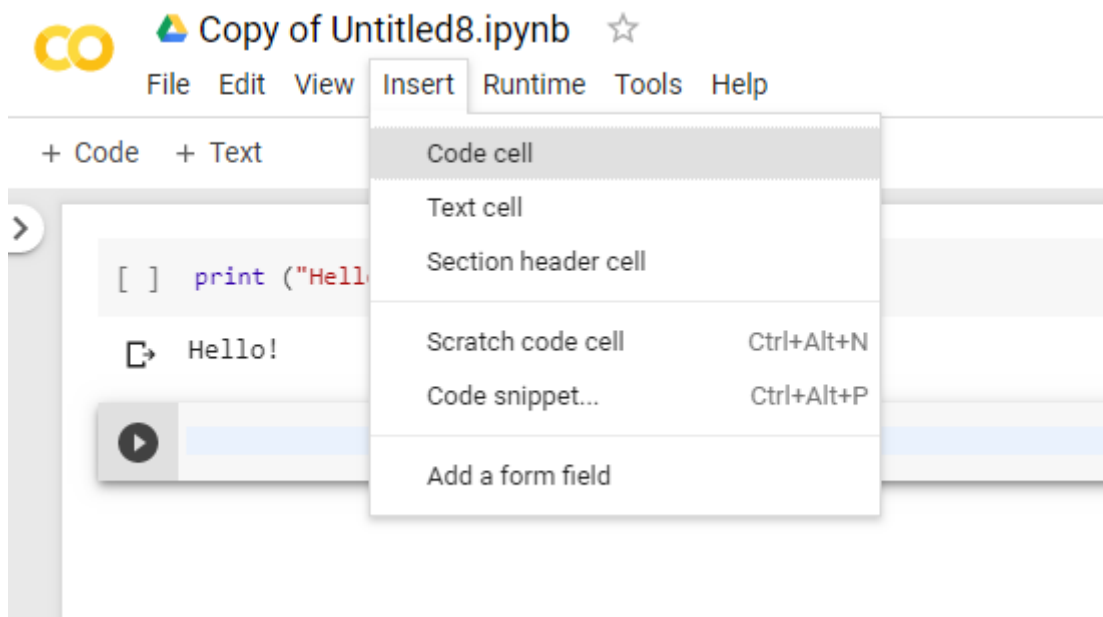


Рис. 5. Добавление новой ячейки кода

При наведении курсора мыши на ячейку кода, справа сверху появляется всплывающее меню (рис. 6). Нажатие на значок «сообщение» позволяет добавить комментарий к ячейке. Нажатие на корзину – удалить ячейку. Нажатие на «шестиренку» даст возможность редактирования настроек данной ячейки.

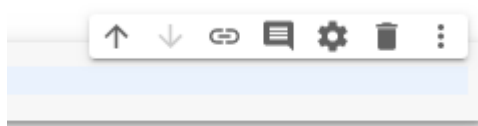


Рис. 6. Всплывающее окно ячейки кода

Автоматическая подсказка синтаксиса кода

Дополнения кода и подсказки из документации происходят автоматически при вводе. Для этого используйте следующие сочетания клавиш:

Ctrl-пробел, чтобы заново открыть завершение кода.

Ctrl-shift-пробел, чтобы заново открыть подсказки параметров.

```
1 import pandas as pd
2
3 pd.D
```

DataFrame	pd.DataFrame(data=None, index=None, columns=None, dtype=None, copy=False)
DateOffset	
DatetimeIndex	
DatetimeTZDtype	

Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure.

Parameters

data : ndarray (structured or homogeneous), iterable, dict, or DataFrame

Для ознакомления с функциями языка Python можно воспользоваться документацией по языку Python: <https://docs.python.org/3/tutorial/>