

Python & Git



python?

Python - основное про язык

- название - в честь шоу
- Появился в 1991 году
- Автор: Гвидо ван Россум
- Особенности языка:
 - интерпретируемый тип исполнения (нам не нужен машинный код, нашу программу прочтет и выполнит специальная программа-интерпретатор)
 - неявная динамическая типизация (когда пишем код, мы не обозначаем тип данных в явном виде, это выясняется на этапе выполнения программы)

Давайте поставим **python** на Ваш компьютер

[У меня Windows](#)

[У меня MacOS](#)

[У меня Linux](#)

[Я хочу необычного](#)

Не забудьте, что мы
используем python версии
3.6 и позднее

где работать в Питоне:

- командная строка/терминал (быстро, но не всегда удобно, не подсвечивает синтаксис)
- Rycharm (хороший, но съедает оперативную память, помните об этом)
- Jupyter Notebook (формат тетради с изолированными ячейками для code и markdown)
- Google Colab (тетрадь, но на сервере, по ссылке есть тьюториал)
 - (а вот и наша тетрадка с парой)

И более подробно:

[Как работать в PyCharm](#)

[Как работать в Jupyter Notebook](#)

[Как работать в GoogleColab](#)

Как использовать
git и **github**?

Git & Github

git - это система контроля версий, с которой можно:

- хранить историю изменений при работе в папке
- иметь удаленный доступ к документам
- следить за изменениями в проекте
- делиться work-in-progress работой

github.com - это сервис/сайт/соцсеть, где реализуется функционал гита как системы

как использовать **github.com**

зарегистрируйтесь на github.com

Потребуется почта, пароль и username

Два варианта работы с гитом:

Вариант 1: более легкий

1. Создать аккаунт на github.com
2. Зайти в свой аккаунт
3. Создать репозиторий, в веб-интерфейсе (кнопочки на странице)
4. Готово! У Вас есть репозиторий.
5. Вы можете добавлять в него файлы и скачивать, используя веб-интерфейс и окна загрузки
6. Но локального репозитория у вас нет :(

Вариант 2: (сложнее, но функциональнее)

1. Создать аккаунт на github.com
2. [Скачать git](#) на свой компьютер
3. Установить git следуя [этому тьюториалу](#)
4. Создать репозиторий:
 - a. Создать локально на своем компьютере и запушить на гитхаб
 - b. или создать на гитхабе и клонировать себе локально (на Ваш выбор)
5. Готово! У вас репозиторий и полный функционал гита с доступом к удаленным и локальным репозиториям!

Git | материалы

интерактивный [курс по Git в командной строке](#)

официальные [руководства по использованию github](#)

Git | репозитории

Репозитории бывают локальными(**local**) и удаленными(**remote**, не deleted!!)

Опубликовать файл в удаленный репозиторий - это как скинуть на гугл диск, только специальный

- Удобно делиться
 - устоявшийся централизованный подход к работе с данными
- Элемент соцсети
 - можно смотреть, что делают друзья, отмечать понравившееся, настраивать, кому что видно в вашей активности
- Удаленными считаем репозитории на сайте github.com, локальными - на вашем компьютере

ССЫЛКИ НА
ПОЛЕЗНЫЕ РЕСУРСЫ

ресурсы для начинающих:

[сайт, куда все идут гуглить свои ошибки кода](#)

[полезные примеры кода](#)

[введение в командную строку \(терминал\)](#)

[введение в Python для не-программистов](#)

[Немного инфы о самом языке](#)

Игры про основы программирования

подойдут для понимания базовых понятий программирования: переменная, цикл, условия.

- [Блочное программирование](#) — дудл Гугла, посвящённый 50-летию детского программирования
- [Ещё одна игра от Google с блочным программированием](#)

Полезные ссылки | **Python**

- [ru](#) Интерактивный учебник по Python
- [ru](#) Интерактивный задачник по Python — не все примеры решения хороши, но много хороших задач
- [ru](#) Курс Вышки на Coursera
- [ru](#) Книга «Укус питона» — подробное вводное пособие, в HTML и PDF
- [en](#) Игра, в которой герой управляется вашей программой
- [en](#) Курс на Codeacademy
- [en](#) Тренажер с упражнениями
- [en](#) Официальный туториал по языку — Очень хороший материал для последовательного изучения языка
- [en](#) Игра для тех, кто уже освоил азы Python

Полезные ссылки | **Git**

- [ru GitHowTo](#) — хороший курс по Git в командной строке
- [ru Git Book](#) — официальный учебник по Git, тоже про командную строку
- [en GitHub Tutorials](#) — официальные руководства по использованию [github](#)
- [en Bitbucket tutorials](#) — руководства по использованию Git на сайте Bitbucket, сервиса-конкрента GitHub
- [en A Quick Introduction to Version Control with Git and GitHub](#) — короткое введение в git и github

Полезные ссылки | Текстовые редакторы

- [Visual Studio Code](#) — отличный бесплатный редактор от Microsoft
- [Notepad++](#) — удобный редактор для быстрых правок
- [Sublime Text](#) — для macOS, Linux и Windows

ссылка на наш репозиторий
(в нем все материалы с пары)

homework

1. [Поставить python](#) на свой компьютер (проверить, что он как минимум версии 3.6)
2. поставить [Pycharm/ Jupyter Notebook/](#) (на Ваш выбор, можно оба, чтобы сравнить)
3. Сделать задачи в [google colab-тетрадке](#)
4. Зарегистрироваться на гитхабе, скачать git на свой компьютер
5. Пройти [вопросы в гугл-форме](#)
6. Проверить, что Вы есть в Telegram-чате
7. Дополнительно: (по желанию)
 - a. Создать свой репозиторий на гитхабе (через командную строку или браузер)
 - b. Скачать колаб-тетрадку с домашкой, запустить ее на гитхаб

в завершение:

- Ошибки в коде - это нормально, они бывают у всех, они помогают понять, что пошло не так
 - *(можно прям гуглить ошибки на stackoverflow, например)*
- Гуглить решение проблем в коде - это полезный навык, смотреть на чужой код, чтобы улучшить свой - это хорошая практика
- Комментарии в коде (которые `#...` или `“”” ...”””`) сохраняют Вам время
- Если код не работает, но не ясно, почему, помогает пройтись от строки к строке, рассказывая себе, что делает каждая из них
- Терминал страшен/сложен только сначала, а вообще очень полезен

Спасибо за
внимание!

удачи, успехов

Как поставить **Jupyter Notebook**

Заходим в командную строку, пишем:

чтобы установить: `pip3 install jupyter notebook`

(эту команду нужно выполнить всего 1 раз, затем он будет сохранен)

чтобы запустить уже установленный: `jupyter notebook`

Готово!

Тетрадка откроется в браузере (лучше всего - Chrome|Chromium)

Если вы используете Linux, скорее всего, у вас уже есть Python 3. Проверьте версию, написав в командной строке `python3 --version`



Python в командной строке (1/2)

Откройте командную строку (она же терминал)

- **MacOS:** приложение Terminal (найдите его через поиск)
 - В окне будет `mycomputer:~ myusername$`
- **Linux:** вероятно, Вы уже знаете многое о терминале
 - В окне будет `myusername@mycomputer:~$`
- **Windows:** наберите в поиске `cmd.exe`
 - В окне будет `C:\Users\myusername>`

Python в командной строке - (2/2)

В появившемся окне напишите `python3`

В питоне из терминала есть 2 режима: интерактивный или пакетный.

- Пакетный выполнит уже готовый, заранее написанный файл (в котором код) с расширением .py Например, `python3 my_file.py`
- Интерактивный будет ждать от Вас питонской команды, например, `print('hello')`
- [Здесь](#) можно почитать более подробно

чуть более необычные способы поставить **python**

1. Можно кодить не с компьютера, а с гаджета на iOS, через приложение [Pythonista](#) (но оно платное)
2. Можно -- с Android-гаджета через приложение [Pydroid3](#) (требуется Android 4.4), еще можно через [QPython](#)
3. Еще одна установка питона на iOS-компьютер, если у вас стоит менеджер пакетов [homebrew](#) (требуется использования командной строки)
 - а. После установки самого Homebrew выполните в командной строке `brew install python`