



**Git. Python tools. Basic operators.**



# Git

Это распределенная система управления версиями



# Установка:

Ubuntu:

```
>>> apt-get install git
```

Windows:

<https://git-scm.com/download/win>

Mac:

<https://git-scm.com/download/mac>



# Bitbucket

Зарегистрируйтесь на сайте <https://bitbucket.org/>.



# Настройка

```
$ git config --global user.name "Ivan Ivanou"
```

```
$ git config --global user.email "ivan_ivanou@mail.com"
```



## Создание проекта

- Создайте директорию: test\_project
- Создайте в директории файл: test.py
- Добавьте в файл строку: print("Hello world")

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects
$ mkdir test_repo

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects
$ cd test_repo/

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo
$ echo "print('Hello world')" >> test_file.py

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo
$ ls -la
total 5
drwxr-xr-x 1 serge 197609  0 Dec  5 21:31 ./
drwxr-xr-x 1 serge 197609  0 Dec  5 21:30 ../
-rw-r--r-- 1 serge 197609 21 Dec  5 21:31 test_file.py

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo
$ cat test_file.py
print('Hello world')
```



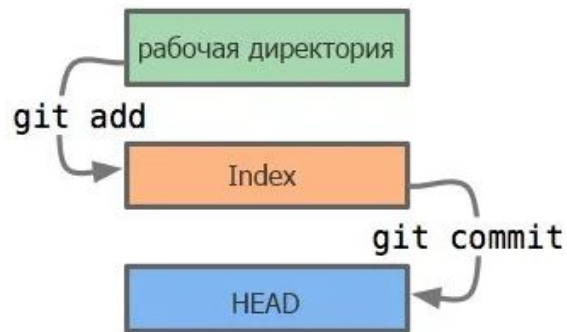
## Создание репозитория:

- git init
- git add <file\_name>
  - git add . - для добавления всех файлов
- git commit -am "message"

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git add test_file.py
warning: LF will be replaced by CRLF in test_file.py.
The file will have its original line endings in your working directory

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git commit -m "First commit message"
[master (root-commit) 1673bd4] First commit message
1 file changed, 1 insertion(+)
create mode 100644 test_file.py
```

Git в своей работе управляет тремя структурами, которые называются деревьями. Первое — это рабочая директория, в ней хранятся файлы, с которыми вы прямо сейчас работаете. Второе — это Index, который позволяет вам вносить изменения и ничего не портить. А третье — это HEAD, который указывает на последний сделанный вами коммит.








# Как проверить состояние?

- `git status`


```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
nothing to commit, working tree clean
```



```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test_file.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/

no changes added to commit (use "git add" and/or "git commit -a")
```



```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git add test_file.py
warning: LF will be replaced by CRLF in test_file.py.
The file will have its original line endings in your working directory

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test_file.py
```



# Git log

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git log
commit 0cda5e109687a080a238607eaf21e5febd65d106 (HEAD -> master)
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 22:02:07 2019 +0300

    Add my name print

commit 1673bd4d6ffbd1605832ab88ece06295f264e8cb
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 21:36:24 2019 +0300

    First commit message
```

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git log --pretty=oneline
0cda5e109687a080a238607eaf21e5febd65d106 (HEAD -> master) Add my name print
1673bd4d6ffbd1605832ab88ece06295f264e8cb First commit message
```



## Как вернуться к старому коммиту?

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git checkout 1673bd4d6ffbd1605832ab88ece06295f264e8cb
Note: switching to '1673bd4d6ffbd1605832ab88ece06295f264e8cb'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to `false`

HEAD is now at 1673bd4 First commit message

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo ((1673bd4...))
$ git status
HEAD detached at 1673bd4
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   test_file.py
```

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .idea/
```

no changes added to commit (use "git add" and/or "git commit -a")



# Отменить изменения до индексации

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test_file.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/

no changes added to commit (use "git add" and/or "git commit -a")

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ cat test_file.py
print('Hello world')
print('My name is Siarhei')
print('My name is Siarhei')
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git checkout test_file.py
Updated 1 path from the index

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ cat test_file.py
print('Hello world')
print('My name is Siarhei')
```



# Отмена проиндексированных изменений

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test_file.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git reset HEAD test_file.py
Unstaged changes after reset:
M       test_file.py

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test_file.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/

no changes added to commit (use "git add" and/or "git commit -a")
```



# Отмена коммитов

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git commit -am "Incorrect commit"
[master 7ca7287] Incorrect commit
1 file changed, 1 insertion(+)

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/

nothing added to commit but untracked files present (use "git add" to track)

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git revert HEAD
[master 1ed0a74] Revert "Incorrect commit"
1 file changed, 1 deletion(-)

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git log
commit 1ed0a74b8d2e1c748a59e721b71d0a4a18a96ec0 (HEAD -> master)
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 22:40:55 2019 +0300

    Revert "Incorrect commit"

    This reverts commit 7ca72875aa437d8fa6061eff5db95bdafad5743a.

commit 7ca72875aa437d8fa6061eff5db95bdafad5743a
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 22:38:38 2019 +0300

    Incorrect commit

commit 0cda5e109687a080a238607eaf21e5febd65d106
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 22:02:07 2019 +0300

    Add my name print

commit 1673bd4d6ffbd1605832ab88ece06295f264e8cb
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 21:36:24 2019 +0300

    First commit message
```





# Изменения коммитов

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git commit -am "Add work description"
[master c08fbe5] Add work description
1 file changed, 1 insertion(+)

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git commit --amend -am "Add work company name and address"
[master 1ad52b2] Add work company name and address
Date: Thu Dec 5 22:45:16 2019 +0300
1 file changed, 1 insertion(+)

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git log
commit 1ad52b258f1b5aab79b4f0afba8579c23d95a9d1 (HEAD -> master)
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 22:45:16 2019 +0300

    Add work company name and address

commit 1ed0a74b8d2e1c748a59e721b71d0a4a18a96ec0
Author: Ivan Ivanou <sergei_xxx2000@mail.ru>
Date: Thu Dec 5 22:40:55 2019 +0300

    Revert "Incorrect commit"

This reverts commit 7ca72875aa437d8fa6061eff5db95bdafad5743a.
```



## Создание новой ветки

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git checkout -b new_feature
Switched to a new branch 'new_feature'

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (new_feature)
$ git status
On branch new_feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .idea/

nothing added to commit but untracked files present (use "git add" to track)
```



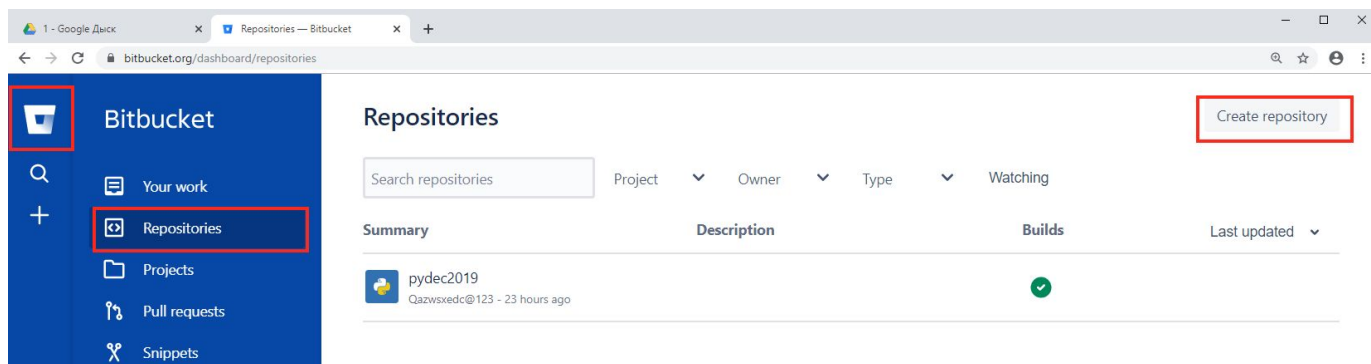
# Переключение между ветками

```
serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (new_feature)
$ git checkout master
Switched to branch 'master'

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (master)
$ git checkout new_feature
Switched to branch 'new_feature'

serge@DESKTOP-N4FDHNP MINGW64 ~/PycharmProjects/test_repo (new_feature)
$ git branch
  master
* new_feature
```

# Создание удаленного репозитория



## Create a new repository

[Import repository](#)

Repository name \*

Access level

☐

This is a private repository

Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README?

Yes, with a tutorial (for beginne... ▼

Version control

☒

Git

system

☐

Mercurial

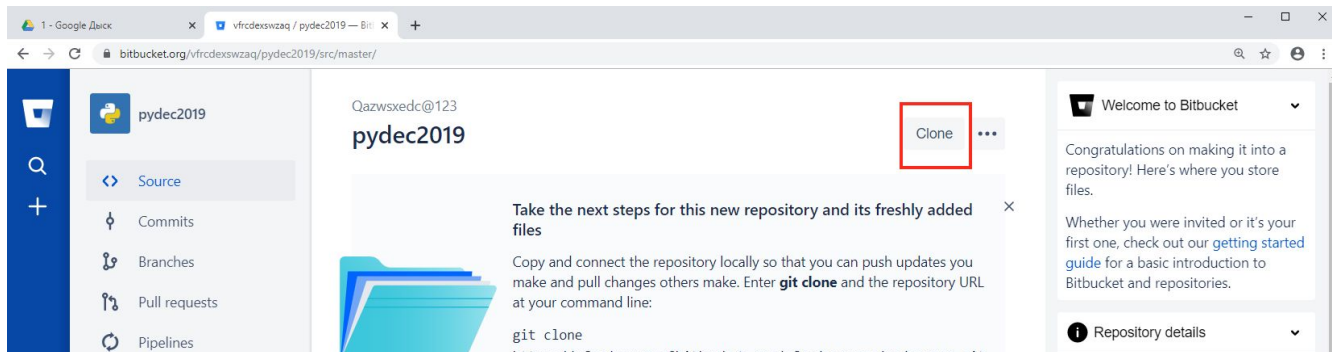
[Advanced settings](#)

Create repository

Cancel

# Clone репозитория

- git clone





# Внесение изменений в оригинальный репозиторий

- `git pull`
- ваши изменения
- `git add .`
- `git commit -m "commit message"`
- `git push`



# Работа с общим репозиторием

Есть права на push:

- `git clone`
- ...
- `git push`

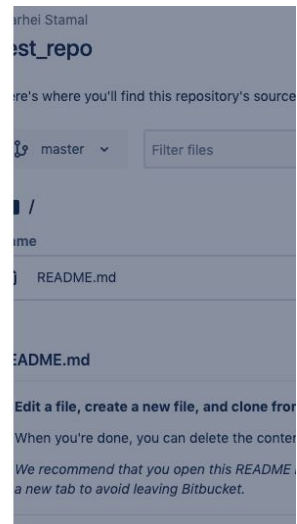
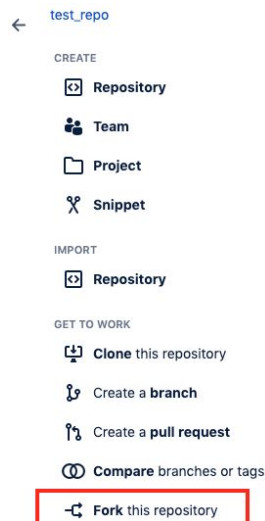
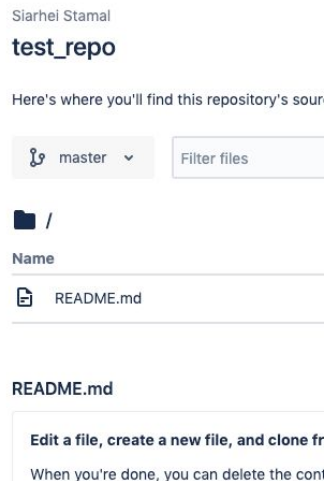
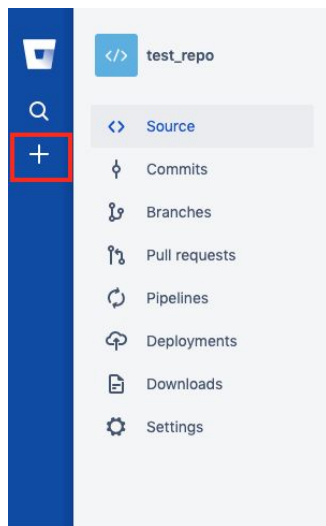
Нет прав на push:

- `fork repository`
- `git clone`
- ...
- `git push`
- `create a pull request`



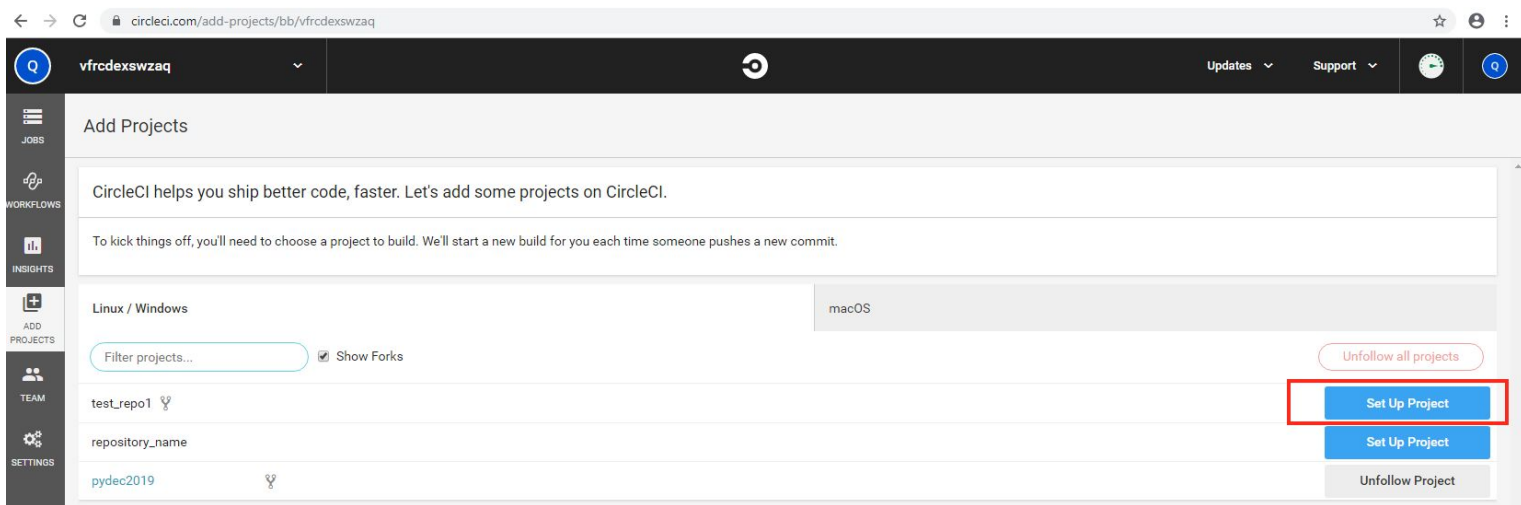
# Fork репозитория

url - [https://bitbucket.org/sergei\\_stamal/git\\_practice/src/master/](https://bitbucket.org/sergei_stamal/git_practice/src/master/)



# Circle CI

Зарегистрируйтесь и войдите под своим аккаунтом на <https://circleci.com/>



← → ↻ circleci.com/add-projects/bb/vfrcdexswzaq ☆ ⓘ ⋮

Q vfrcdexswzaq ▾

JOBS

WORKFLOWS

INSIGHTS

ADD PROJECTS

TEAM

SETTINGS


## Add Projects

CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.


To kick things off, you'll need to choose a project to build. We'll start a new build for you each time someone pushes a new commit.

Linux / Windows macOS

Filter projects... ☒ Show Forks

test\_repo1 

repository\_name

pydec2019 

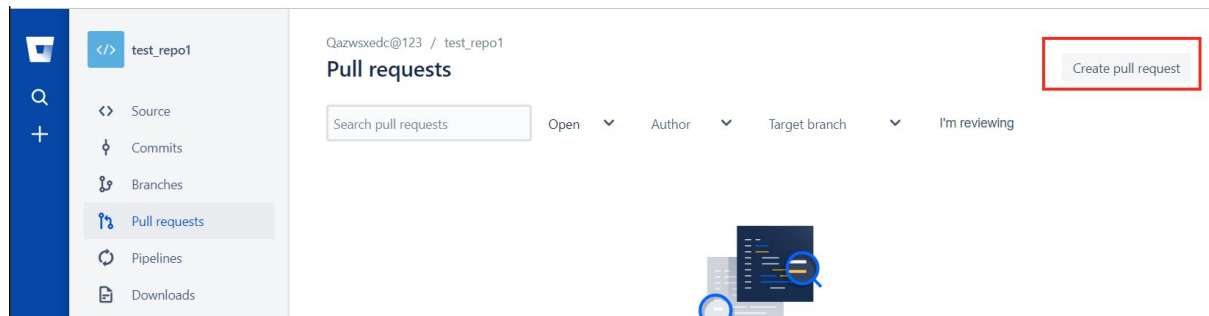
Unfollow all projects

Set Up Project

Set Up Project

Unfollow Project

# Создание pull request



Qazwsxedc@123 / test\_repo1 / Pull requests

## Create a pull request

 vfrcdexswzaq / test\_repo1  
Created 6 minutes ago, updated 3 minutes ago

 feat/test\_branch

→

 sergei\_stamal/test\_repo

 master

Title \* test message

Description

**Aa** **B** *I* ...        +

Description

Feedback ?

Reviewers  Siarhei Stamal x

Close branch ☐ Close **feat/test\_branch** after the pull request is merged

Create pull request

Diff Commits

Author	Commit	Message	Date	Builds
 Qazwsxedc@1...	d634445	test message	5 minutes ago	



## .gitignore

Если у Вас есть файлы, которые вы не только не хотите автоматически добавлять в репозиторий, но и видеть в списках неотслеживаемых. К таким файлам обычно относятся автоматически генерируемые файлы (различные логи, результаты сборки программ и т.п.). В таком случае, вы можете создать файл `.gitignore` с перечислением шаблонов соответствующих таким файлам.

- `*.loa` - игнорировать файлы заканчивающиеся на `*.o` или `*.a`
- `build/` - игнорировать файлы в директории
- `doc/*.txt` - игнорировать файлы `*.txt` в директории `doc`
- `!lib.a` - отслеживать файл `lib.a`



# Правила по наименованию бранчей и КОММИТОВ

<https://www.conventionalcommits.org/en/v1.0.0-beta.2/>



# pip

Pip - система управления пакетами, которая используется для установки пакетов Python.

Установка pip:

```
>> curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py  
>> python get-pip.py
```

Ubuntu:

```
>> sudo apt install python3-pip
```



Установка пакетов: *pip install XXX*

Удаление пакетов: *pip uninstall XXX*

Обновление: *pip install --upgrade XXX* или *pip install XXX -U*

Список установленных пакетов: *pip list*

Поиск пакетов: *pip search XXX*

Версия: *pip --version/pip3 --version*





# Установка пакетов

```
$ pip freeze > requirements.txt
```

```
$ pip install -r requirements.txt
```



## pyenv

Менеджер версий для языка программирования Python. Позволяет изменять версию Python в системе, устанавливать различные версии Python, управлять виртуальным окружением.

Не поддерживает Windows.



Установка (<https://github.com/yyuu/pyenv>). Git должен быть установлен:

```
$ curl -L https://raw.githubusercontent.com/yyuu/pyenv-installer/master/bin/pyenv-installer | bash
```

Список доступных версий Python:

```
$ pyenv install -l
```

Установка Python определенной версии:

```
$ pyenv install <version>
```

Список установленных версий:

```
$ pyenv versions
```

Переключиться на конкретную версию Python:

```
$ pyenv local 3.6.8
```



# virtualenv

Инструмент позволяющий создавать виртуальные окружения с пакетами. Он работает, устанавливая множество файлов в каталоге (например: `env/` ), а затем изменяя переменные окружения среды, так чтобы префикс был (например: `env/bin/` ). После активации вы можете устанавливать пакеты в виртуальной среде с помощью `pip` .



# Работа с окружением virtualenv

Установка:

```
$ sudo pip install virtualenv
```

```
$ sudo apt-get install python-virtualenv
```

Создание:

```
$ cd ~/projects/<project_name>
```

```
$ mkdir venv && "Virtualenv directory" > venv/README
```

```
$ git add venv && echo "/venv/" >> .gitignore && git add -f .gitignore
```

```
$ virtualenv --no-site-packages --prompt="(<project_name>)" <venv_name>
```



# Работа с окружением virtualenv

Активация:

```
$ source ~/venv/<venv_name>/bin/activate
```

Деактивация:

```
$ deactivate
```

Установка пакетов:

```
$ source ~/venv/<venv_name>/bin/activate
```

```
$ pip install <some_package>
```

```
$ easy_install <some_package>
```




# Venv

Рекомендован для использования с python 3.3+

```
>> python3 -m venv name-given  
>> source name-given/bin/activate -> source name-given/Scripts/activate  
>> deactivate  
>> rm -r name-given
```

Устанавливать пакеты через pip



```
├── bin
│   ├── activate
│   ├── activate.csh
│   ├── activate.fish
│   ├── easy_install
│   ├── easy_install-3.5
│   ├── pip
│   ├── pip3
│   ├── pip3.5
│   ├── python -> python3.5
│   ├── python3 -> python3.5
│   └── python3.5 -> /Library/Frameworks/Python.framework/Versions/3.
├── include
├── lib
│   ├── python3.5
│   │   └── site-packages
└── pyvenv.cfg
```

- bin – файлы, которые взаимодействуют с виртуальной средой;
- include – C-заголовки, компилирующие пакеты Python;
- lib – копия версии Python вместе с папкой «*site-packages*», в которой установлена каждая зависимость.






## Проверка работоспособности:

```
>> which python  
>> echo $PATH  
>> pip -q install requests  
>> python -c "import requests"
```



# Базовые операторы





# Арифметические операторы

+

Сложение

-

Вычитание

\*

Умножение

/

Деление

%


Остаток

\*\*

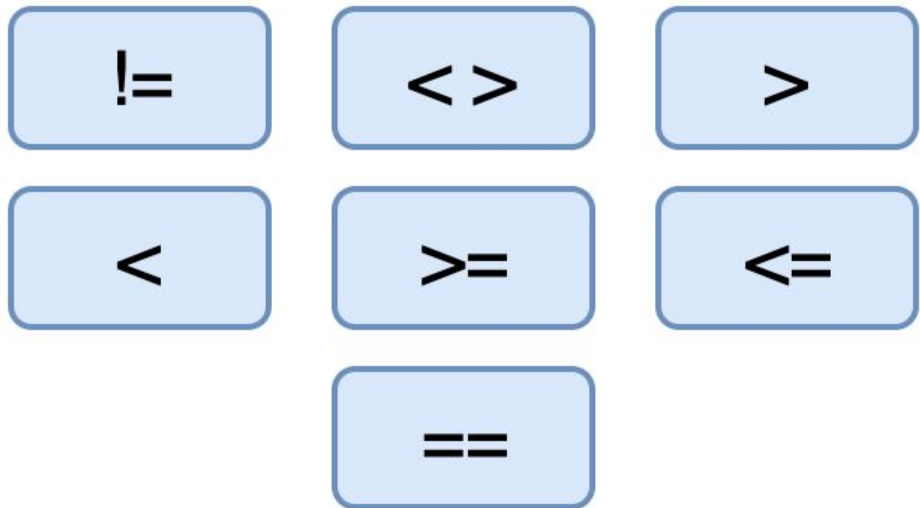
Возведение в  
степень

//

Целочисленное  
деление

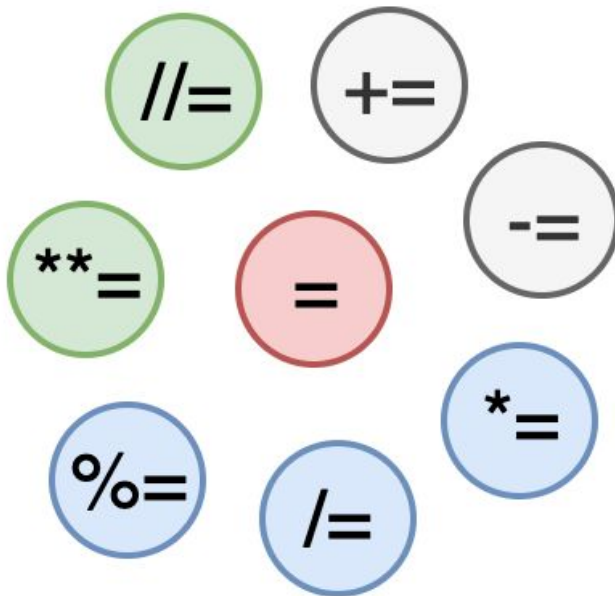


## Операторы сравнения



Note: Оператор "`<>`" не используется с версии Python 3

# Операторы присваивания





# Битовые операции

**&**

Побитовое И (AND)

**|**

Побитовое ИЛИ (OR)

**^**

Сложение по модулю (XOR)

**~**

Инверсия

**>>**

Сдвиг вправо

**<<**

Сдвиг влево



## Операторы принадлежности

in

not in






## Операторы идентичности

is

is not



# Приоритет операторов

OPERATOR	DESCRIPTION
**	Exponentiation (raise to the power)
~ + —	Complement, unary plus and minus (method names for the last two are +@ and -@)
* / % //	Multiply, divide, modulo and floor division
+ —	Addition and subtraction
>> <<	Right and left bitwise shift
&	Bitwise 'AND'
^	Bitwise exclusive 'OR' and regular 'OR'
<= < > >=	Comparison operators
<> == !=	Equality operators
= %= /= //= -= += *= **=	Assignment operators
is is not	Identity operators
in not in	Membership operators
not or and	Logical operators