



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")

```
sadovsky@ws167:~/PycharmProjects$ mkdir test_project
sadovsky@ws167:~/PycharmProjects$ cd test_project/
sadovsky@ws167:~/PycharmProjects/test_project$ ls -la
total 8
drwxr-xr-x  2 sadovsky sadovsky 4096 Nov  1 19:42 .
drwxr-xr-x 17 sadovsky sadovsky 4096 Nov  1 19:42 ..
sadovsky@ws167:~/PycharmProjects/test_project$ nano test.py
sadovsky@ws167:~/PycharmProjects/test_project$ echo print("Hello world") >> test.py
bash: syntax error near unexpected token `('
sadovsky@ws167:~/PycharmProjects/test_project$ echo "print('Hello world')" >> test.py
sadovsky@ws167:~/PycharmProjects/test_project$ ls -la
total 12
drwxr-xr-x  2 sadovsky sadovsky 4096 Nov  1 19:43 .
drwxr-xr-x 17 sadovsky sadovsky 4096 Nov  1 19:42 ..
-rw-r--r--  1 sadovsky sadovsky  22 Nov  1 19:43 test.py
sadovsky@ws167:~/PycharmProjects/test_project$ cat test.py

print('Hello world')
```

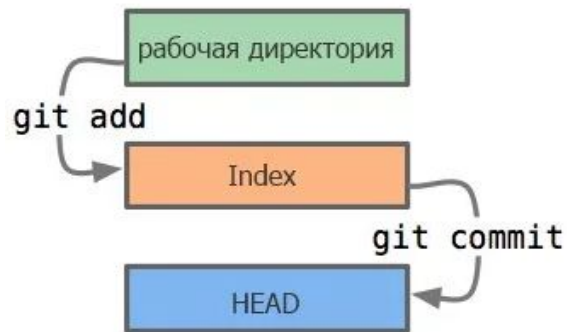


Создание репозитория:

- `git init`
- `git add <file_name>`
 - `git add .` - для добавления всех файлов
- `git commit -am "message"`

```
sadovsky@ws167:~/PycharmProjects/test_project$ git add test_file.py
sadovsky@ws167:~/PycharmProjects/test_project$ git commit -am "First commit"
[master e90184b] First commit
 1 file changed, 1 insertion(+)
 create mode 100644 test_file.py
sadovsky@ws167:~/PycharmProjects/test_project$
```

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`

```
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch master
nothing to commit, working tree clean
sadovsky@ws167:~/PycharmProjects/test_project$
```



```
sadovsky@ws167:~/PycharmProjects/test_project$ git status
```

```
On branch master
```

```
Changes not staged for commit:
```


```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   test_file.py
```

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

```
sadovsky@ws167:~/PycharmProjects/test_project$
```



```
sadovsky@ws167:~/PycharmProjects/test_project$ git add test_file.py
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   test_file.py

sadovsky@ws167:~/PycharmProjects/test_project$
```



Git log

```
sadovsky@ws167:~/PycharmProjects/test_project$ git log
commit 70bea7967c0be6d0aa190b4c95e1dd6c6258920d (HEAD -> master)
Author: DSadovsky <DSadovsky@arlo.com>
Date:   Sun Nov 1 20:04:23 2020 +0300

    Add My Name

commit 0cc3fe865f9516a3fcaa463075795973ae252fcf
Author: DSadovsky <DSadovsky@arlo.com>
Date:   Sun Nov 1 20:03:23 2020 +0300

    First commit
sadovsky@ws167:~/PycharmProjects/test_project$
```

```
sadovsky@ws167:~/PycharmProjects/test_project$ git log --pretty=oneline
70bea7967c0be6d0aa190b4c95e1dd6c6258920d (HEAD -> master) Add My Name
0cc3fe865f9516a3fcaa463075795973ae252fcf First commit
```



Как вернуться к старому коммиту?

```
sadovsky@ws167:~/PycharmProjects/test_project$ git checkout 0cc3fe865f9516a3fcaa463075795973ae252fcf
Note: checking out '0cc3fe865f9516a3fcaa463075795973ae252fcf'.
```

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 performing another checkout.

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

```
git checkout -b <new-branch-name>
```

```
HEAD is now at 0cc3fe8 First commit
```



Отменить изменения до индексации

```
sadovsky@ws167:~/PycharmProjects/test_project$ cat test_file.py
print('Hello world Dmitry')
print('Hi Dmitry!')
print('Hi Dmitry?')
sadovsky@ws167:~/PycharmProjects/test_project$ git checkout test_file.py
sadovsky@ws167:~/PycharmProjects/test_project$ cat test_file.py
print('Hello world Dmitry')
sadovsky@ws167:~/PycharmProjects/test_project$
```

Отмена проиндексированных изменений

```
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   test_file.py

sadovsky@ws167:~/PycharmProjects/test_project$ git reset HEAD test_file.py
Unstaged changes after reset:
M   test_file.py
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   test_file.py

no changes added to commit (use "git add" and/or "git commit -a")
sadovsky@ws167:~/PycharmProjects/test_project$
```



Отмена КОММИТОВ

```
sadovsky@ws167:~/PycharmProjects/test_project$ git commit -am "Wrong commit"
[master f44f528] Wrong commit
 1 file changed, 1 insertion(+)
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch master
nothing to commit, working tree clean
sadovsky@ws167:~/PycharmProjects/test_project$ git revert HEAD
[master 7e09ff0] Revert "Wrong commit"
 1 file changed, 1 deletion(-)
```

```
sadovsky@ws167:~/PycharmProjects/test_project$ git log
commit 7e09ff064f817368addc12f5abe9058178572bfa (HEAD -> master)
Author: DSadovsky <DSadovsky@arlo.com>
Date: Sun Nov 1 20:16:04 2020 +0300

    Revert "Wrong commit"

    This reverts commit f44f528b984e77dfa7a7f8edf8a337384c618486.

commit f44f528b984e77dfa7a7f8edf8a337384c618486
Author: DSadovsky <DSadovsky@arlo.com>
Date: Sun Nov 1 20:15:38 2020 +0300

    Wrong commit

commit 70bea7967c0be6d0aa190b4c95e1dd6c6258920d
Author: DSadovsky <DSadovsky@arlo.com>
Date: Sun Nov 1 20:04:23 2020 +0300

    Add My Name

commit 0cc3fe865f9516a3fcaa463075795973ae252fcb
Author: DSadovsky <DSadovsky@arlo.com>
Date: Sun Nov 1 20:03:23 2020 +0300

    First commit
```




Изменения коммитов

```
sadovsky@ws167:~/PycharmProjects/test_project$ git commit -am "Add Day number"
sadovsky@ws167:~/PycharmProjects/test_project$ git commit --amend -am "Add Day number and year"
[master 606aa15] Add Day number and year
Date: Sun Nov 1 20:18:50 2020 +0300
1 file changed, 1 insertion(+)
```

```
sadovsky@ws167:~/PycharmProjects/test_project$ git log
commit 606aa15d5fbcf27f2e99c0eaadc703d0599684f5 (HEAD -> master)
Author: DSadovsky <DSadovsky@arlo.com>
Date: Sun Nov 1 20:18:50 2020 +0300

    Add Day number and year

commit 7e09ff064f817368addc12f5abe9058178572bfa
Author: DSadovsky <DSadovsky@arlo.com>
Date: Sun Nov 1 20:16:04 2020 +0300

    Revert "Wrong commit"

    This reverts commit f44f528b984e77dfa7a7f8edf8a337384c618486.
```



Создание новой ветки

```
sadovsky@ws167:~/PycharmProjects/test_project$ git checkout -b new_branch
Switched to a new branch 'new_branch'
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch new_branch
nothing to commit, working tree clean
sadovsky@ws167:~/PycharmProjects/test_project$
```



Переключение между ветками

```
sadovsky@ws167:~/PycharmProjects/test_project$ git checkout master
Switched to branch 'master'
sadovsky@ws167:~/PycharmProjects/test_project$ git status
On branch master
nothing to commit, working tree clean
sadovsky@ws167:~/PycharmProjects/test_project$
```

Создание удаленного репозитория

1 - Google Диск x Repositories — Bitbucket x +

bitbucket.org/dashboard/repositories

Bitbucket

- Your work
- Repositories**
- Projects
- Pull requests
- Snippets

Repositories

Create repository

Search repositories Project Owner Type Watching

Summary	Description	Builds	Last updated
pydec2019 Qazwsxedc@123 - 23 hours ago			



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

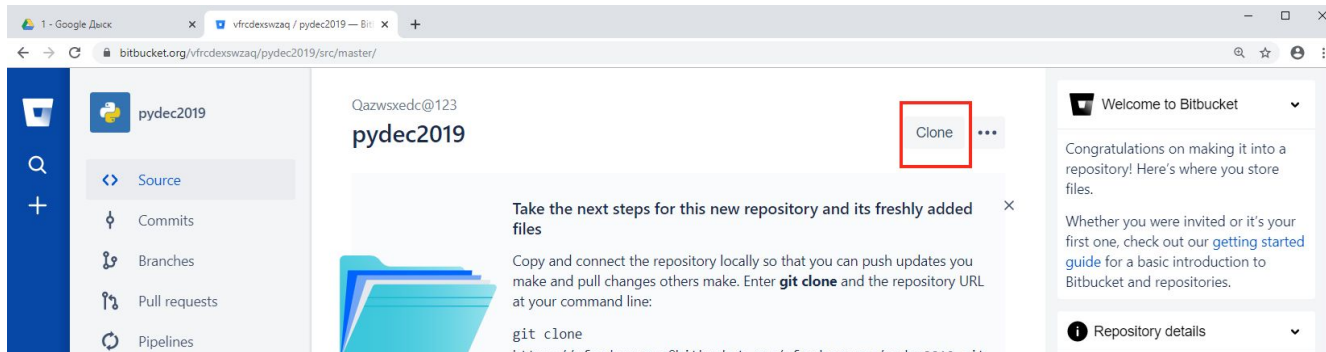
[> Advanced settings](#)

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:

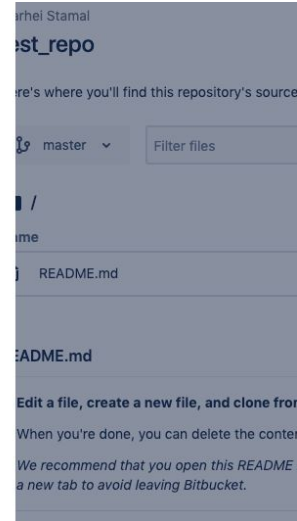
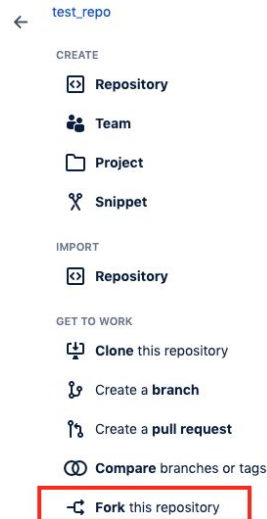
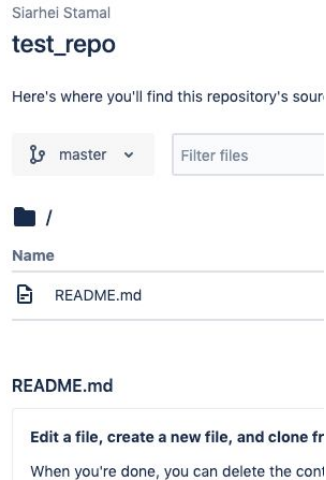
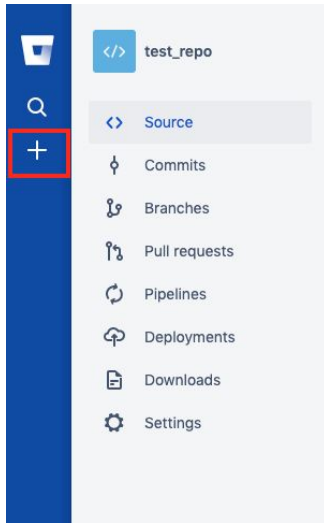
- git clone
- ...
- git push

Нет прав на push:

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

Fork репозитория

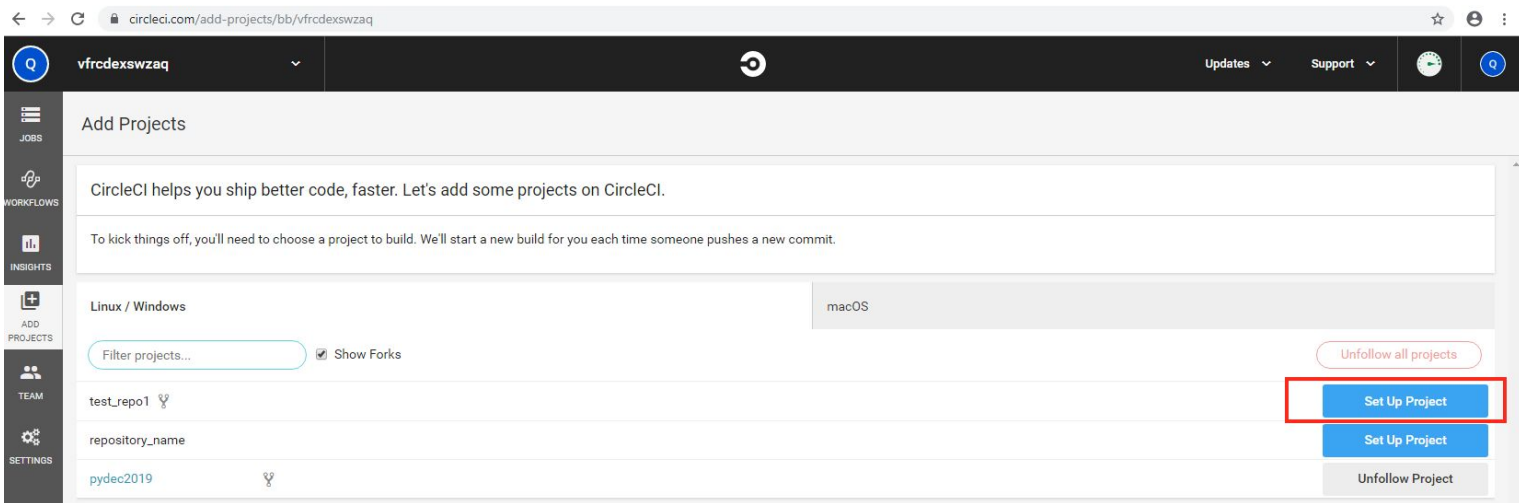
url - https://bitbucket.org/dmitrysadovskii/git_class_practice/src/master/





Circle CI

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



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

vfrcdexswzaq Updates Support

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.

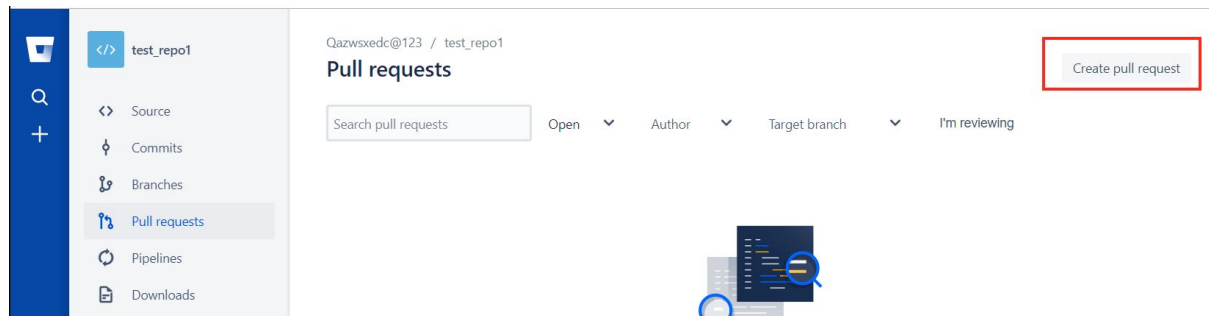
Linux / Windows macOS

Filter projects... Show Forks Unfollow all projects

test_repo1	Set Up Project
repository_name	Set Up Project
pydec2019	Unfollow Project




Создание pull request





The screenshot shows the GitHub interface for a repository named 'test_repo1'. On the left is a navigation sidebar with options: Source, Commits, Branches, Pull requests (highlighted), Pipelines, and Downloads. The main content area is titled 'Pull requests' and includes a search bar labeled 'Search pull requests'. To the right of the search bar are filters: 'Open', 'Author', 'Target branch', and 'I'm reviewing'. A red rectangular box highlights the 'Create pull request' button in the top right corner of the main content area. At the bottom right, there is a small graphic of a computer monitor with a magnifying glass over it.


Dmitry Sadovskii / test_project / test_repository / Pull requests


Create a pull request

 bitsad / test_repository
Created 12 minutes ago, updated 6 minutes ago

 circleci-project-setup 

→

 bitsad/test_repository

 master


Title* Add.circleci/config.yml

Description

Aa **B** **I** **...** **☰** **☷** **🔗** **📎** **@** **👤** **🔍** **🔗** **🔗** **🔗** **+** **...**

Feedback ?


Attachments [Browse to upload](#)

Reviewers  Dmitry Sadovskii x

Close branch Close **circleci-project-setup** after the pull request is merged

Create pull request

Diff Commits

Author	Commit	Message	Date	Builds
 Dmitry Sadovskii	231edf8	Add.circleci/config.yml	6 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.

- `*.[oa]` - 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 ветвей 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/yuyu/pyenv>). Git должен быть установлен:

```
$ curl -L https://raw.githubusercontent.com/yuyu/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

**

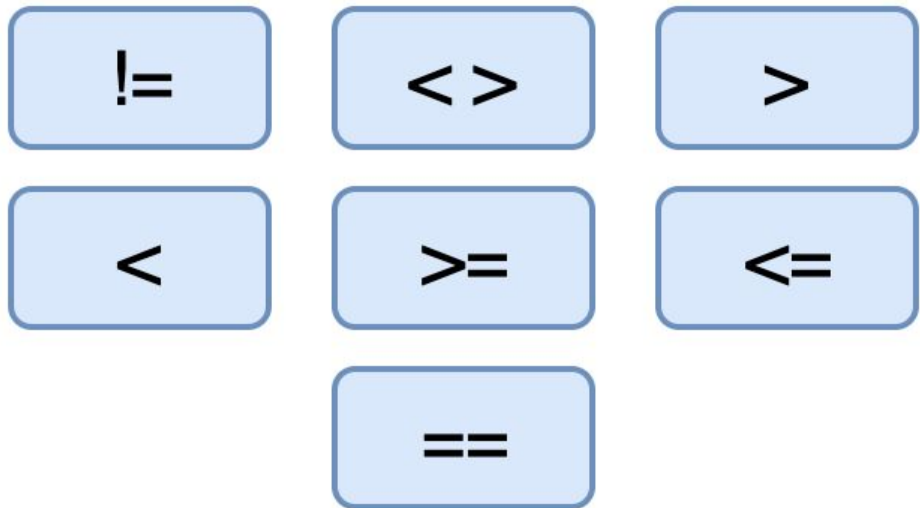
Возведение в
степень

//


Целочисленное
деление



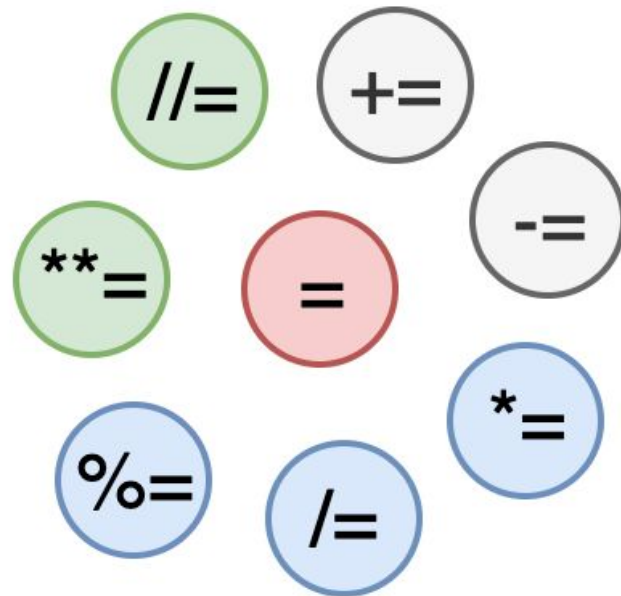
Операторы сравнения



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