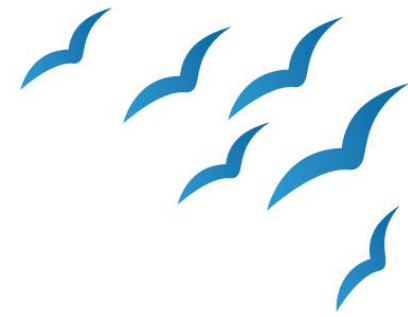




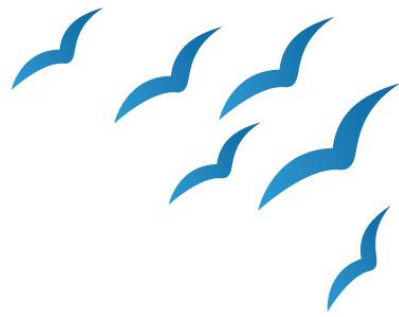
- Система контроля версий



# История создания

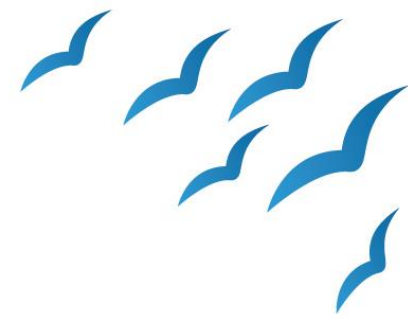
Проект был создан Линусом Торвальдсом (Linus Benedict Torvalds) для управления разработкой ядра Linux.

Первая версия выпущена 7 апреля 2005 года меньше чем за неделю разработки.



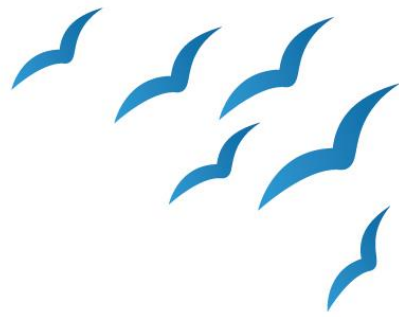
# Git – хостинг

- GitHub
- GitLab
- Bitbucket
- SourceForge
- Codebase
- и другие



# Git – система контроля версии

- Летописец
- Машина времени
- Резервная копия
- Мастер параллельных миров
- Народное вече

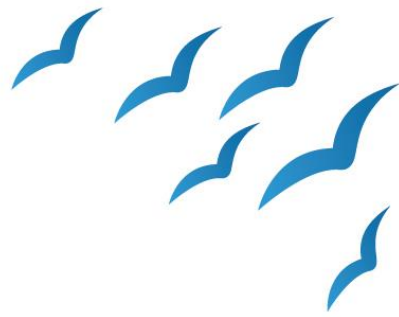


# Летописец

Git – ведет всю историю разработки начиная от сотворения проекта. Делает подшивку черновиков в ветку по контрольным точкам (commit)

> git log

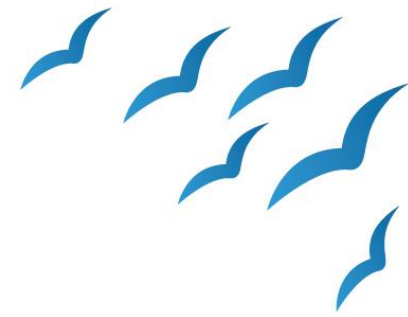
commit c80803cc67c80de23be9fd32626a49dc8c20895e (HEAD -> master, origin/master, origin/HEAD)



# Машина времени

Предоставляет возможность путешествовать в прошлое по дереву летописных сводов.

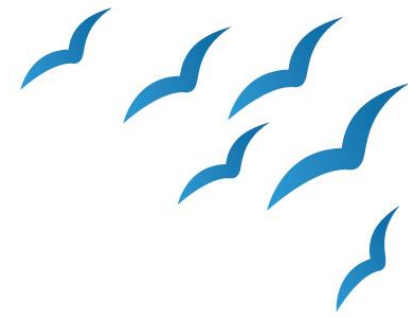
```
>git checkout <hash>
```



# Резервная копия

Создание резервных копий на удаленных серверах git репозитория.

```
>git push origin master
```

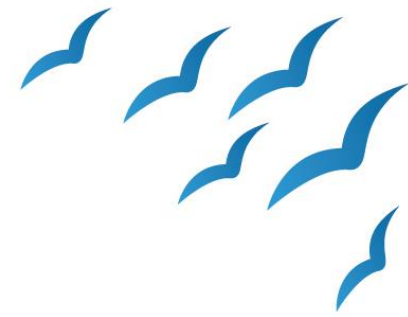


# Мастер параллельных миров

В Git существует возможность распареллелить ветку разработки на несколько , посредством создания дополнительных веток.

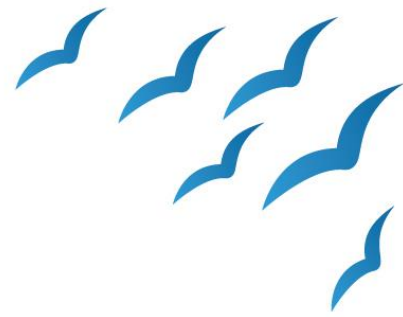
```
>git branch new_branch
```





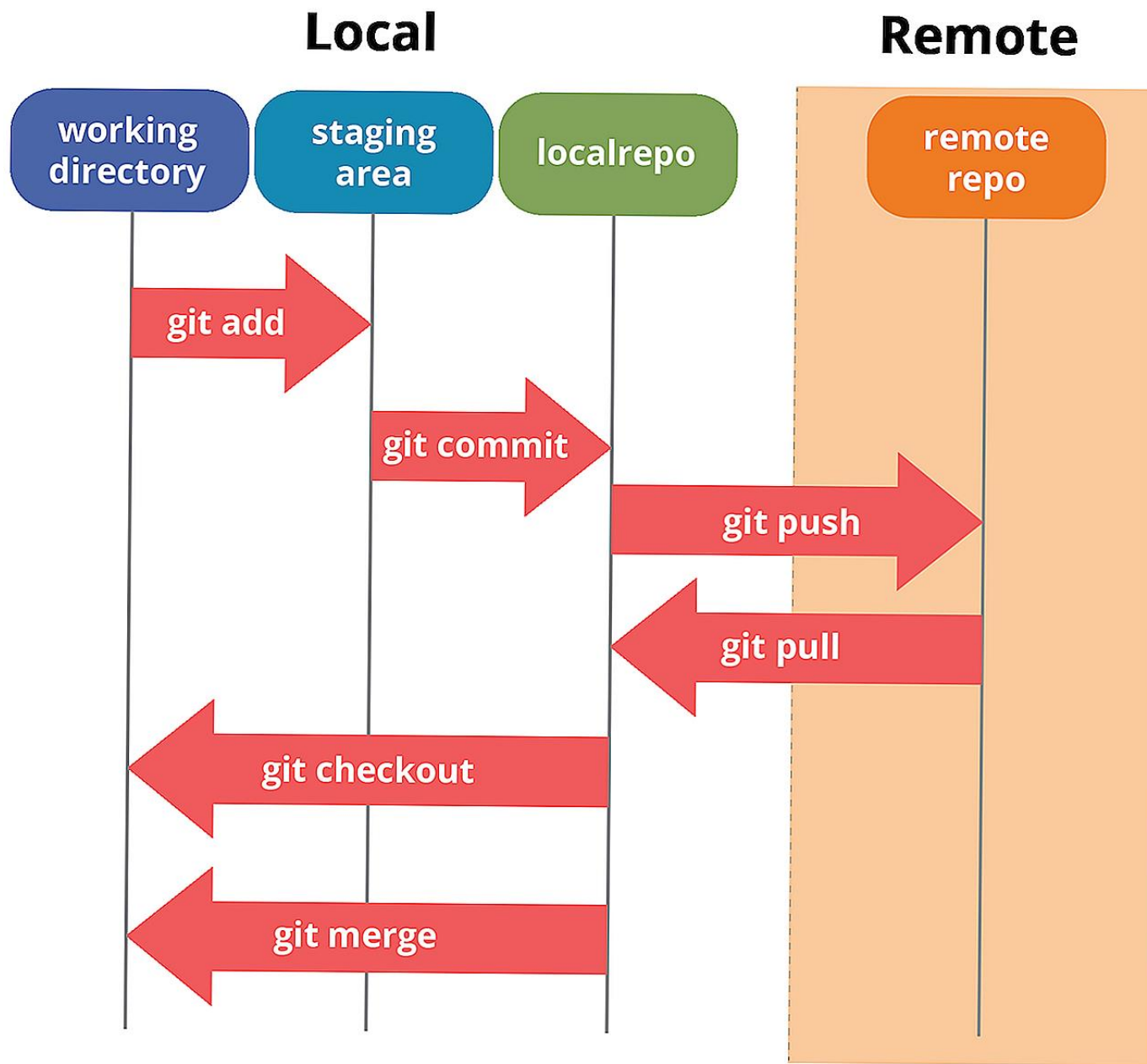
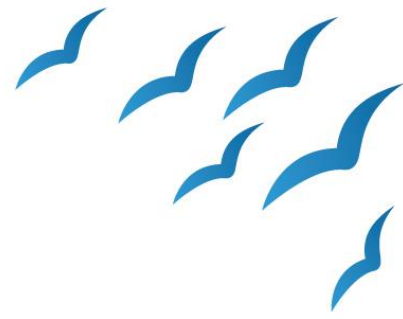
# Народное вече

Git воплощает в себе механизмы коллективного общения разработчиков. Принятия и обсуждения тех или иных технических решений. Идея pull request



Управление репозиторием Git  
сводится к набору команд.  
Рассмотрим диаграмму.





# Установка Git

## (<https://git-scm.com/downloads>)



git-scm.com/downloads

Haskell Python JS

**git** --fast-version-control

Search entire site...

**About**  
**Documentation**  
**Downloads**  
GUI Clients  
Logos  
**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release  
**2.39.0**  
[Release Notes \(2022-12-12\)](#)  
[Download for Windows](#)

### GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.  
[View GUI Clients →](#)

### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.  
[View Logos →](#)

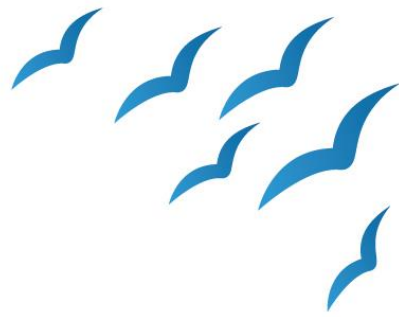
### Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

# Зарегистрироваться на GitHub



<https://github.com/>

Выбрать разумное имя, типа MikhailZimnev59 )))

Создать репозиторий: **python\_winter\_work\_2024**

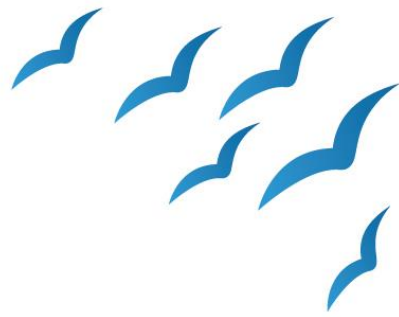
Послать мне полное имя репозитория на почту [m.zimnev@yandex.ru](mailto:m.zimnev@yandex.ru)  
для проверки домашних заданий.

Срок помещения заданий: 14.00 – 16.00 перед следующей лекцией.

Если вы мне пришлете письмо, что вы выложили решения, то ускорите процесс проверки.

Если добавите ссылку на репозиторий, то еще больше ускорите проверку ))

Кто не справится с GitHub, тот может присылать тексты программ прямо в письме.



# Настройка

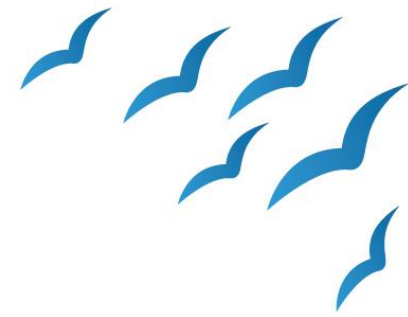
#Устанавливаем в командной строке

```
git config --global user.name "<ваше_имя>"
```

```
git config --global user.email "<адрес_почты@email.com>"
```

# Просмотр настроек

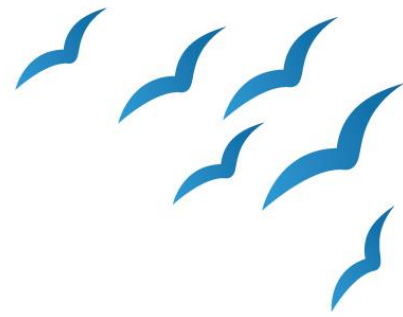
```
>git config --list
```



# Превратить каталог, который не находится под  
# версионным контролем, в репозиторий Git

>**git** init

# После того как была создана папка .git  
# добавляем в текущей каталог файл .gitignore  
# В него заносим все файлы и папки которые не хотим # индексировать



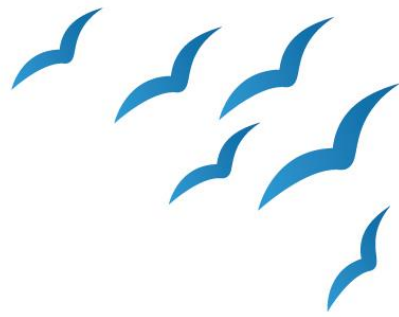
# Индексировать измененный файл

>**git** add [путь к файлу]

# Индексировать измененные файлы

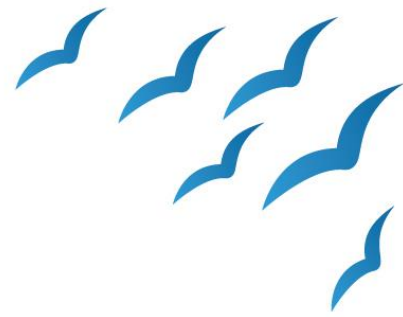
>**git** add .





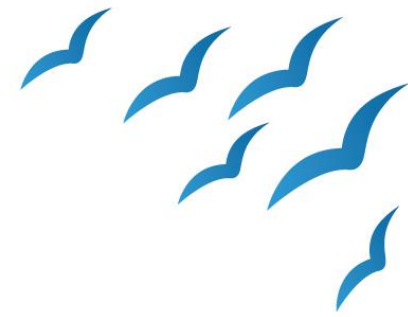
# Просмотр изменений

>**git** status



# Фиксация изменения в локальное хранилище  
# В коммит попадут (будут сохранены) только файлы,  
# которые были проиндексированы командой git add

>**git commit -m** “комментарии к коду”



# Просмотр истории коммитов

>**git log**

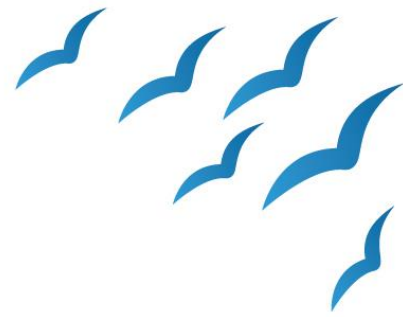
# Информация о коммите (метаданные):

# 1. уникальный идентификатор коммита (хеш);

# 2. имя и email автора коммита;

# 3. дата создания коммита;

# 4. комментарий к коммиту.



# Связывание локального репозитория с GitHub

```
>git remote add origin  
  git@github.com:my\_name/my\_repo.git
```

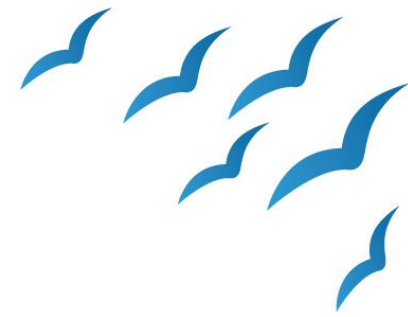
# Где `my_name` — имя пользователя на GitHub  
 `my_repo` — название созданного репозитория

# Проверка свyzывания

```
>git remote get-url origin
```

# Удаление свyzывания

```
>git remote remove origin
```



# Отправка изменений в удаленный  
# репозиторий origin ветки master

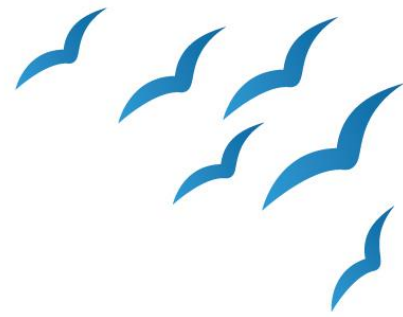
>**git push** origin master

# Отправка изменений с текущей ветки

>**git push**

# Получение изменений

>**git pull**



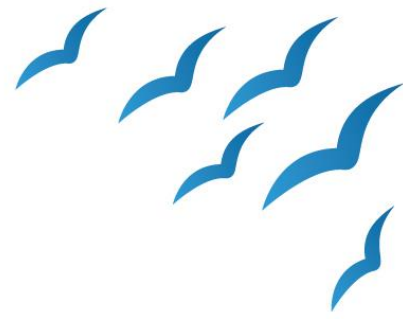
# Клонировать существующий репозиторий

>**git clone** ссылка-на-репозиторий

# Задание



- 1) Создать репозиторий на GitHub с названием **python\_winter\_work\_2024**
- 2) Создать папку на локальной машине для домашних работ с названием **python\_winter\_work\_2024**
- 3) Превратить созданный каталог в репозиторий Git
- 4) Создать в папке каталога файл README.MD
- 5) Добавить его в локальный репозиторий
- 6) Связать удаленный репозиторий с локальным
- 7) Отправить изменения в удаленный репозиторий



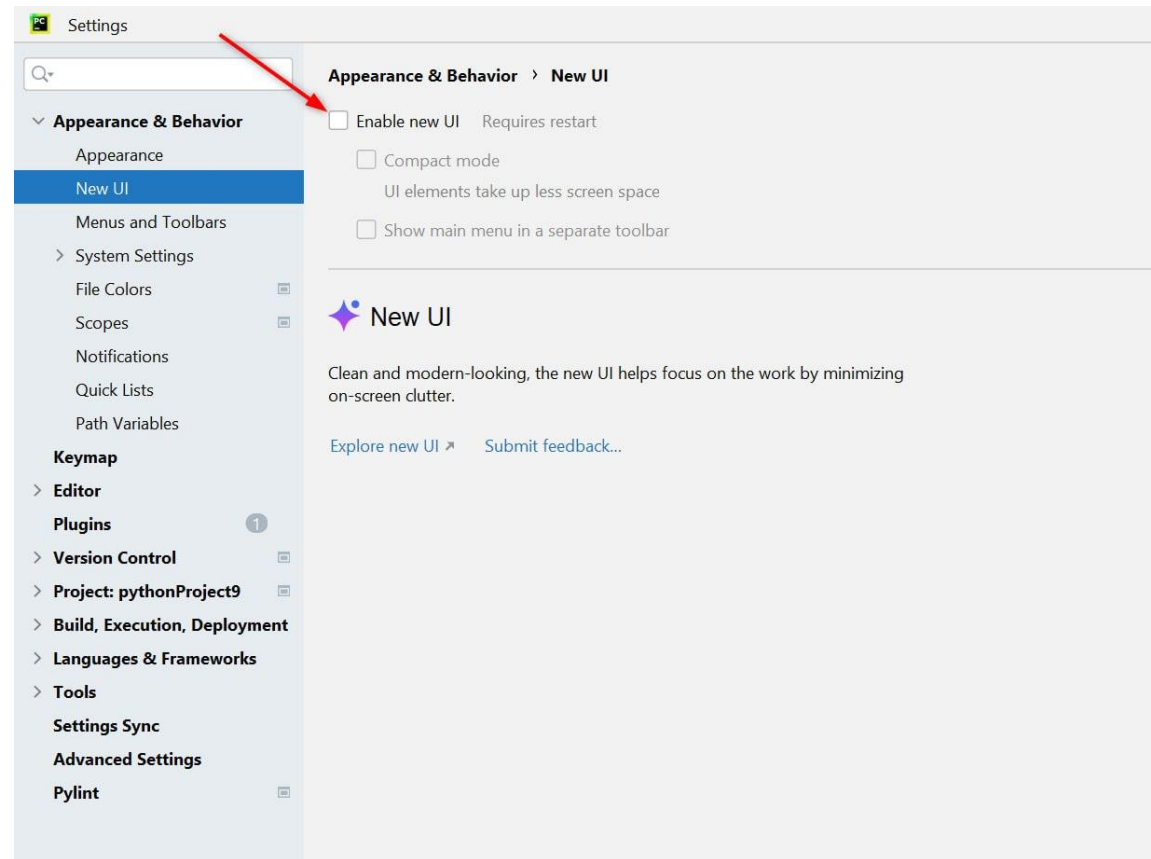
# PyCharm / Github



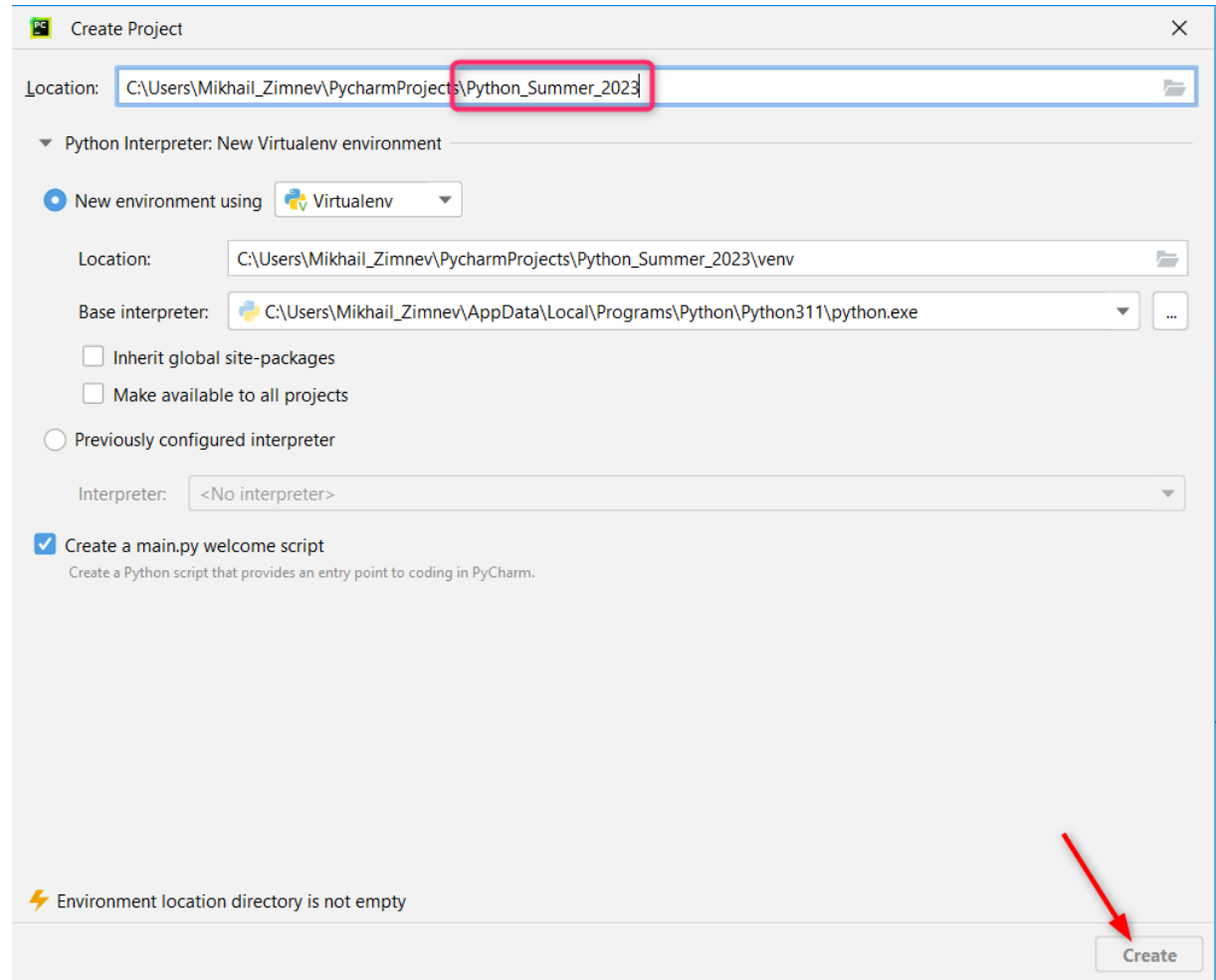
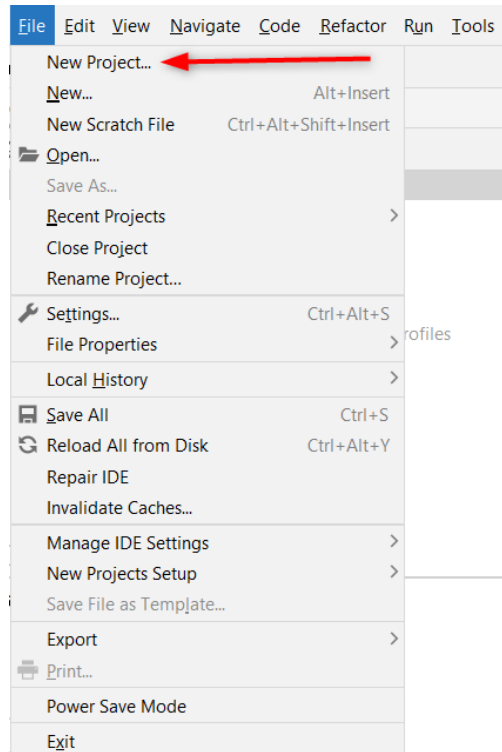


# Настройка UI

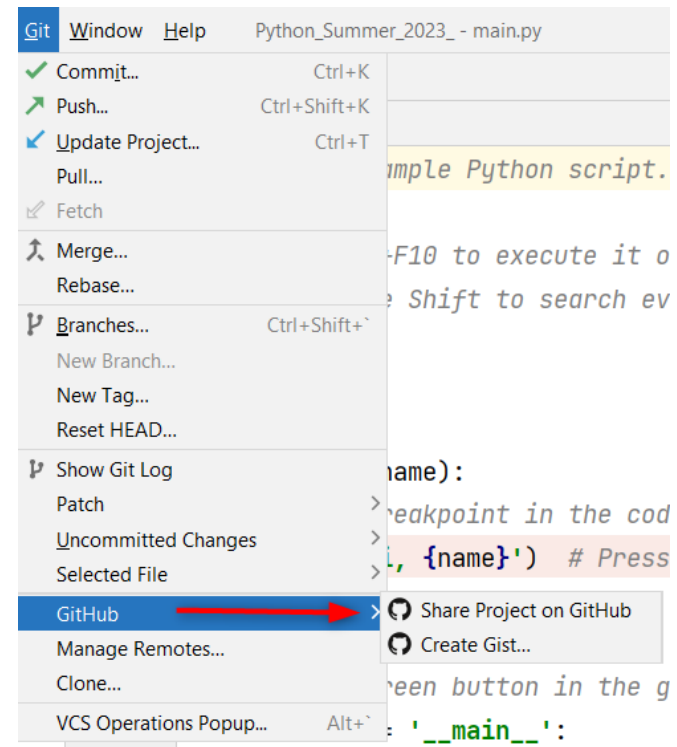
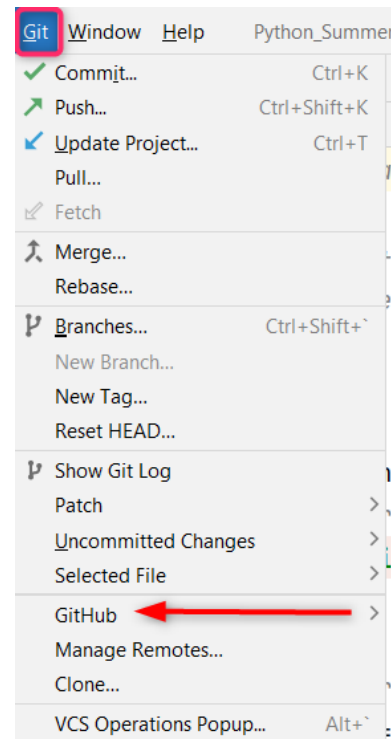
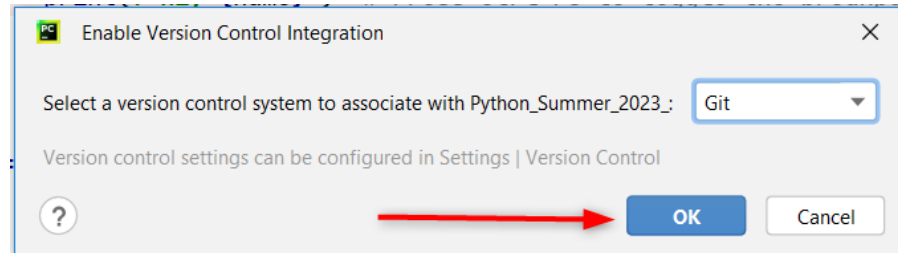
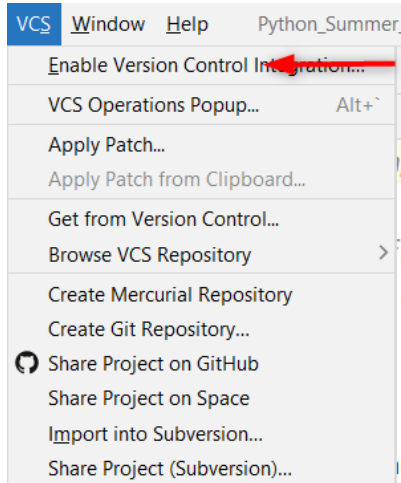
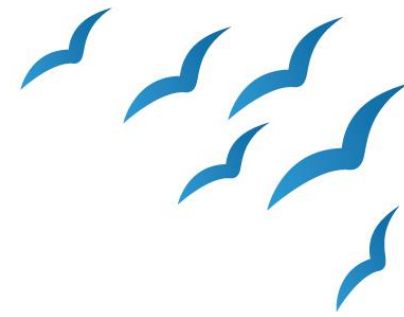
Дальнейшие слайды относятся к традиционному интерфейсу. Поэтому убедитесь, что в настройках File/Settings/New UI настройка Enable new UI отключена.



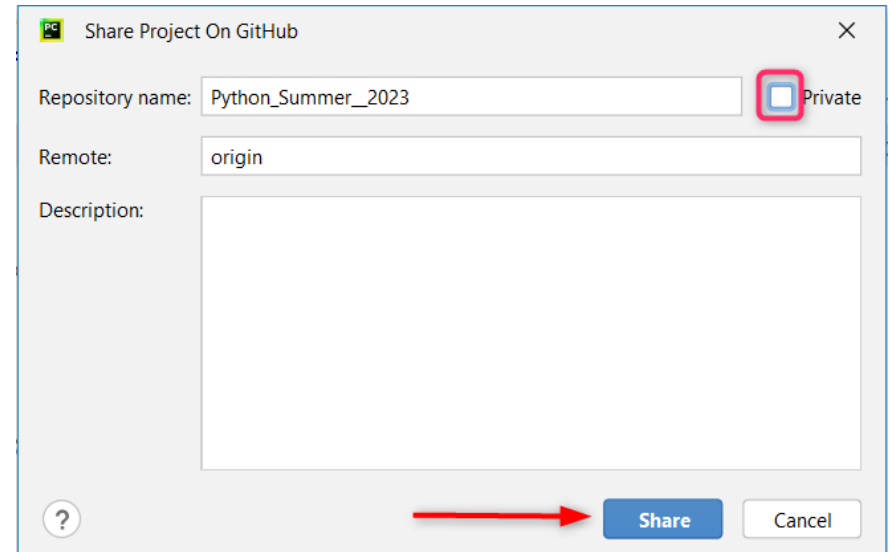
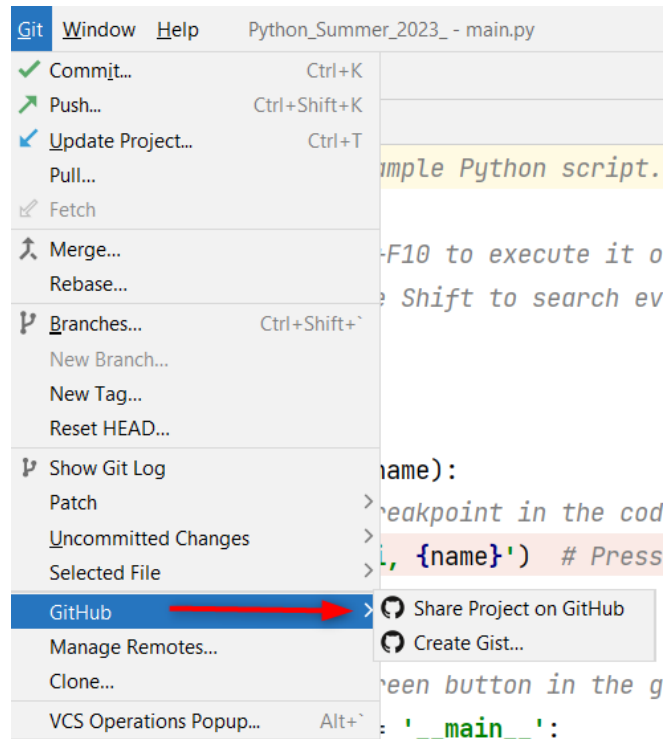
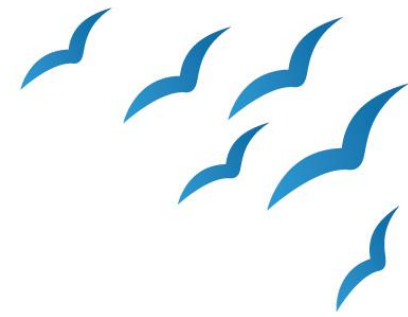
# Создание нового проекта в PyCharm

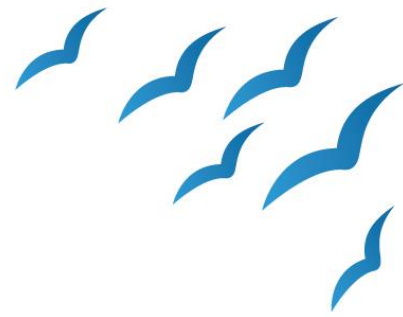


# Git



# Git





**Share Project On GitHub**

Repository name:  ☐ Private

Remote:

Description:

Share by:  [Add account](#)

Account cannot be empty

**Share Project On GitHub**

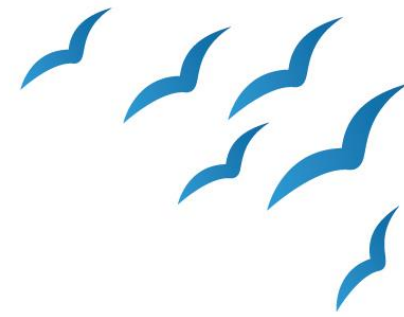
Repository name:  ☐ Private

Remote:

Description:

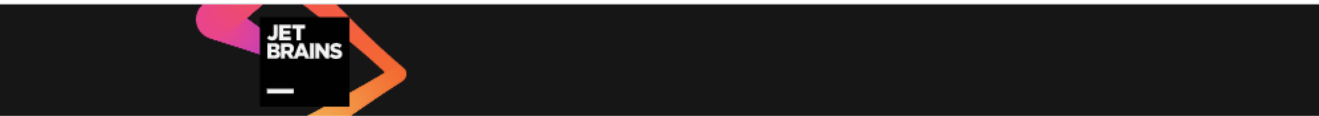
Share by:  [Add account](#)

- Log In via GitHub...
- Log In with Token...
- Log In to GitHub Enterprise...




← → ↻ 🏠 🔒 https://account.jetbrains.com/github/oauth/intellij/authorize?code\_challenge=hWwwiqAeOsyG6RjjUhd)

🧩 Apps 📄 English 📄 Haskell 📄 Python 📄 JS



Please continue only if this page is opened from a [JetBrains IDE](#).

Authorize in GitHub

 Share Project On GitHub ✕

Repository name:  ☐ Private

Remote:

Description:

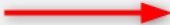
Share by: 

github.com/MikhailZimnev59

▼

[Add account ▼](#)

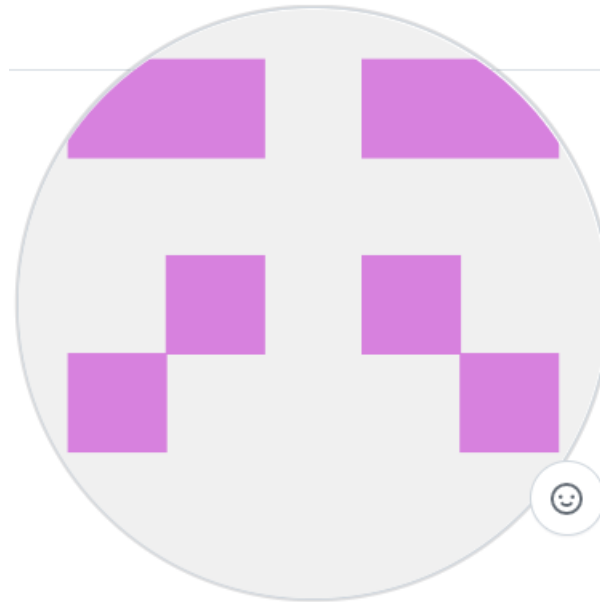
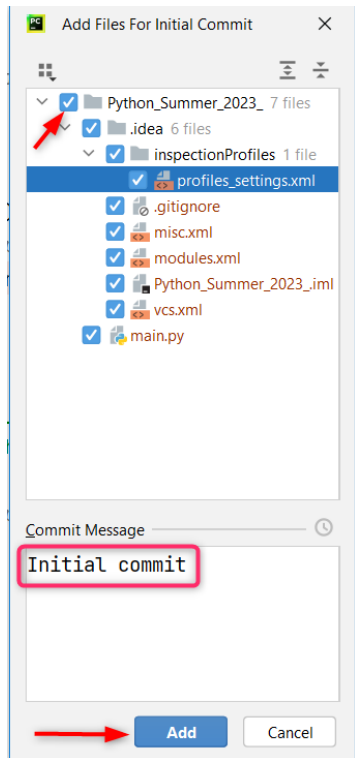
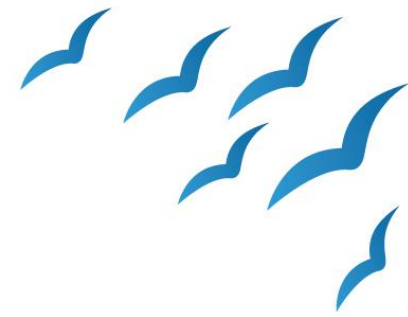
?



Share

Cancel

# Initial Commit



Overview Repositories 14 Projects

Find a repository...

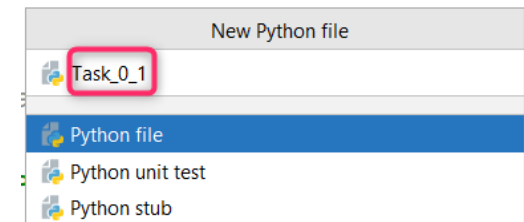
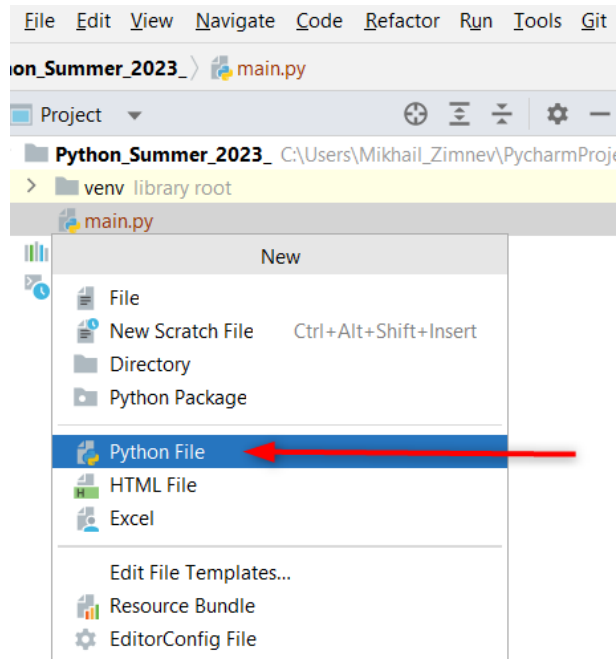
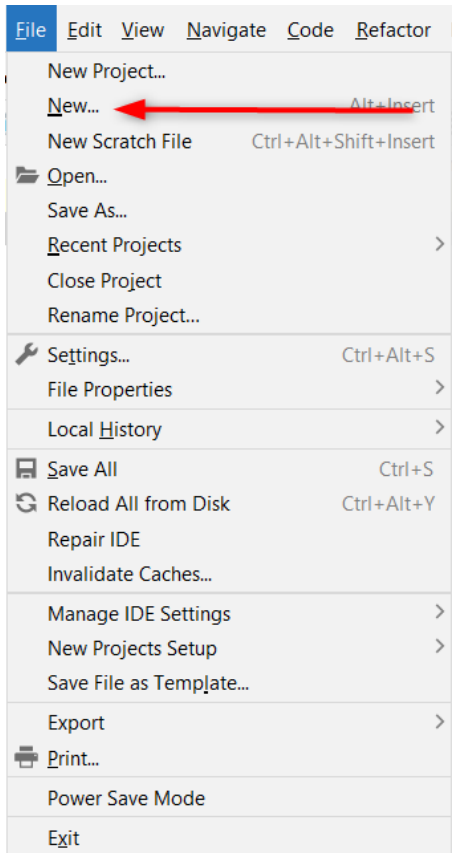
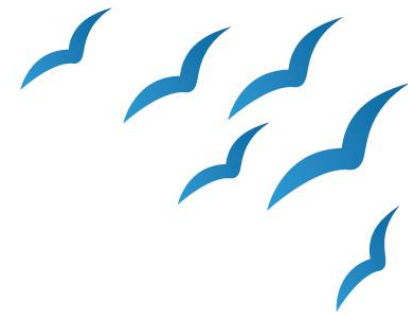
Summer\_2023 Public

Python Updated 6 minutes ago

Python\_Summer\_2023 Public

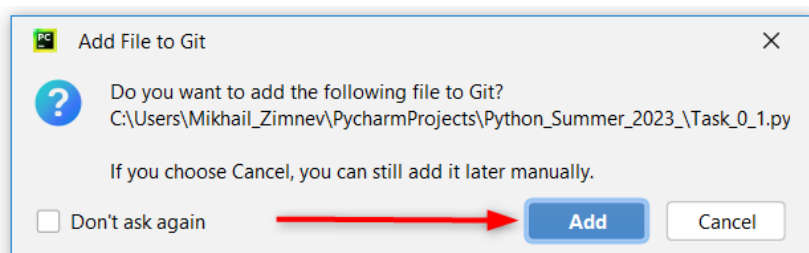
Python Updated 12 hours ago

# Создание нового файла

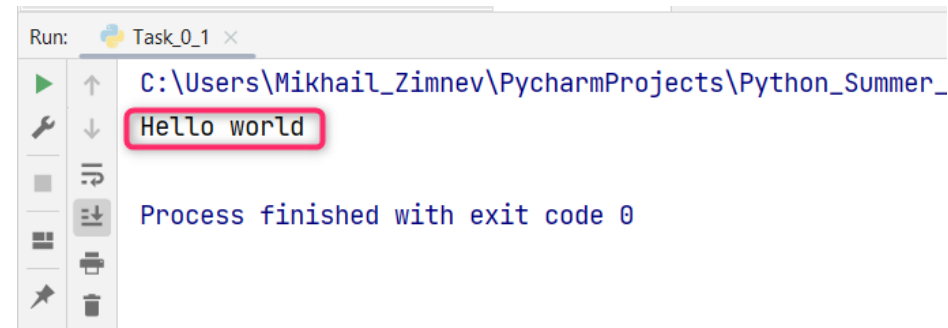
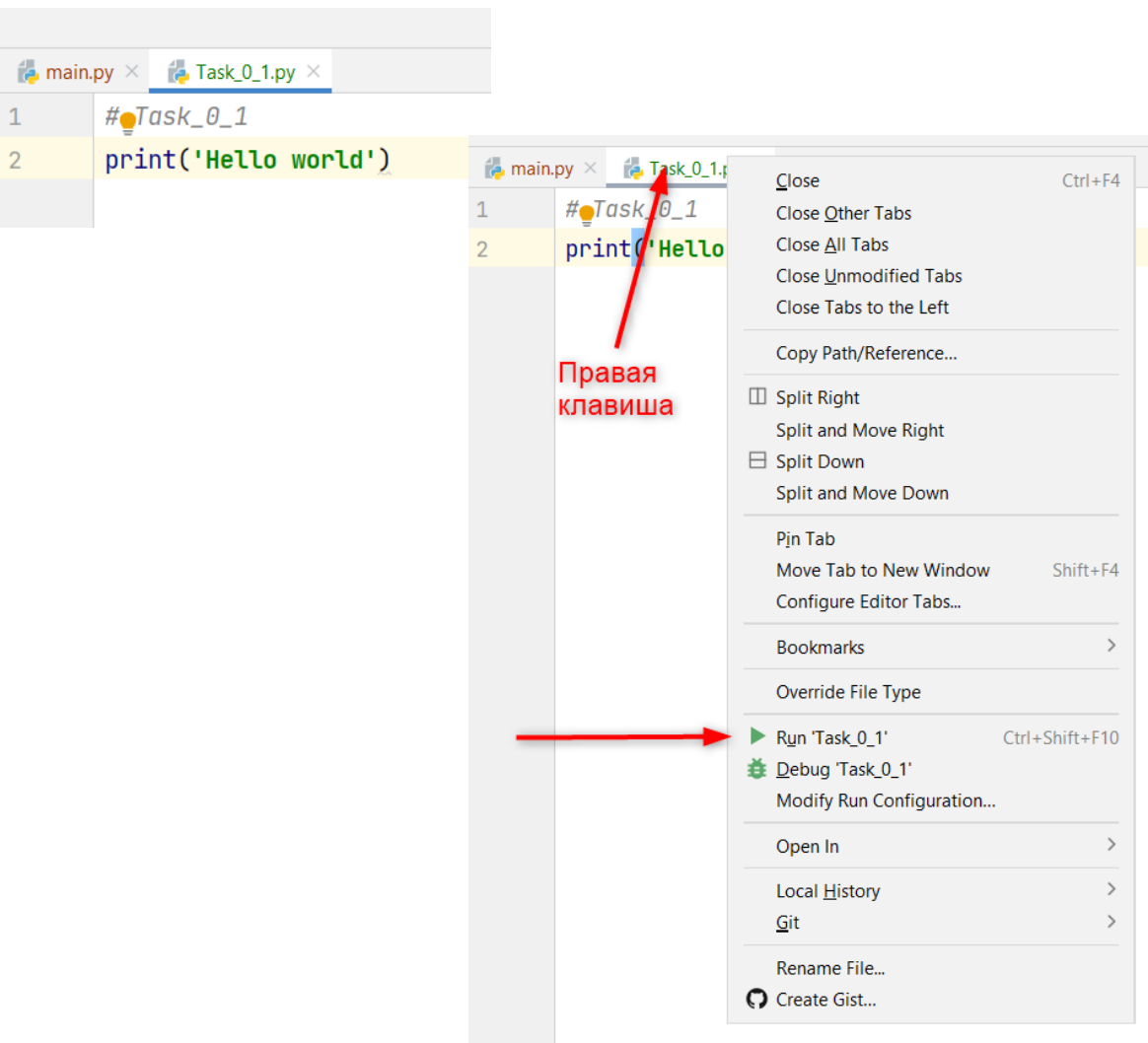
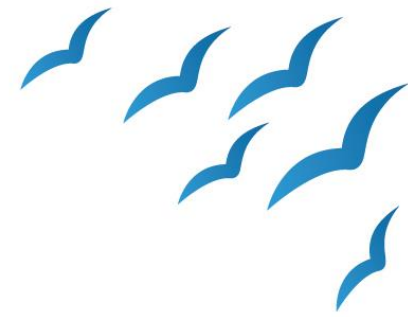




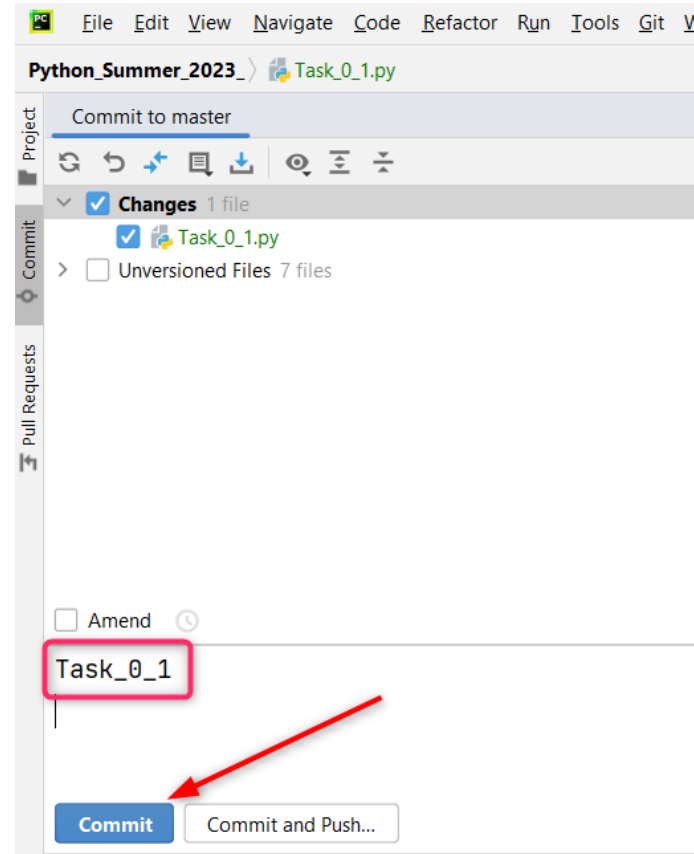
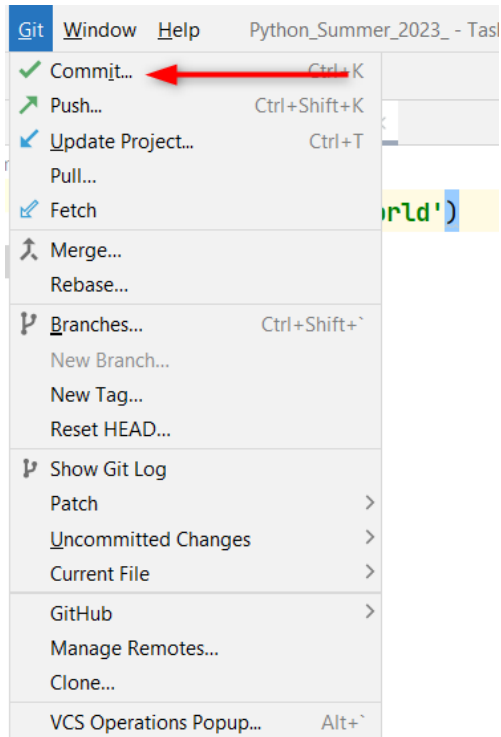
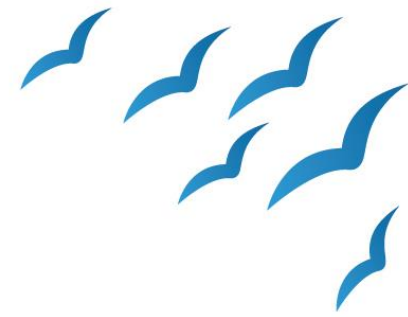
# Добавить файл в Git



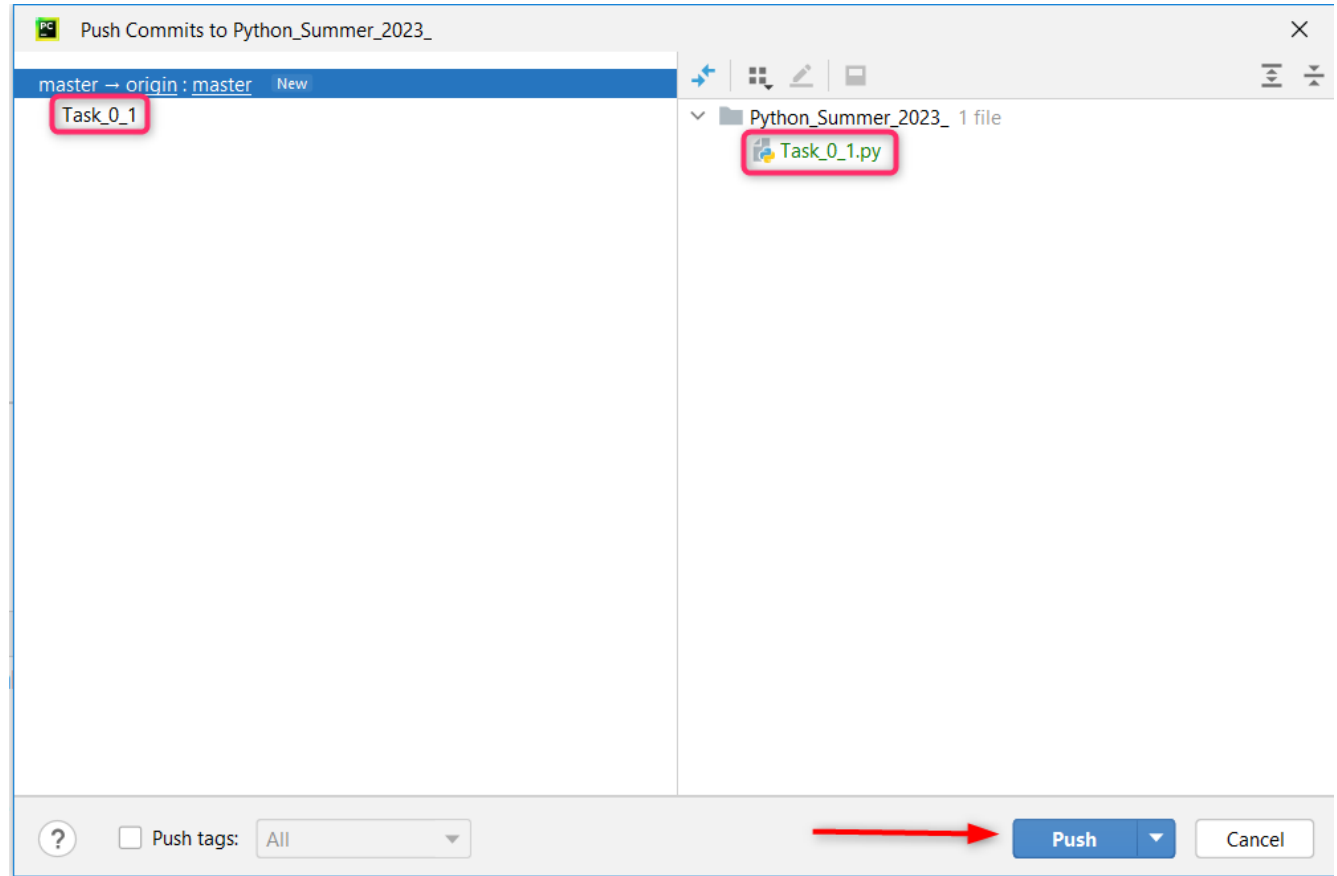
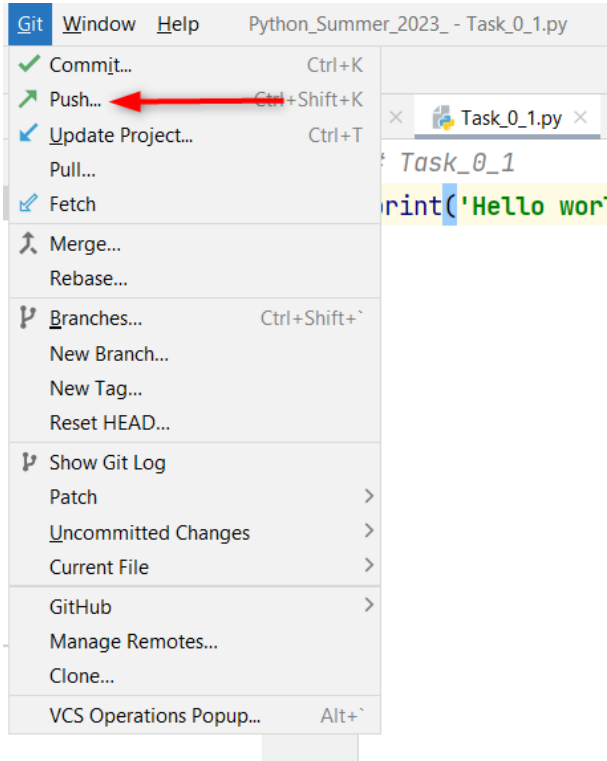
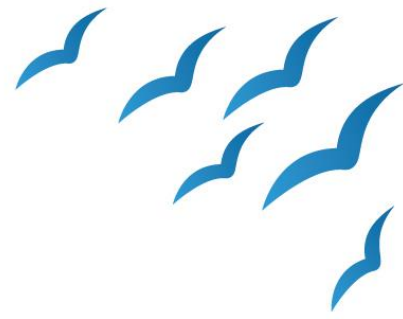
# Программа и запуск ее



# Commit



# Push



# GitHub



[Overview](#) [Repositories](#) 14 [Projects](#) [Packages](#)



Find a repository...

**Summer\_2023** [Public](#)

Python Updated 10 minutes ago

**Python\_Summer\_2023** [Public](#)

Python Updated 13 hours ago

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[master](#) [1 branch](#) [0 tags](#)

[Go to file](#) [Add file](#) [Code](#)

Mikhail\_Z Task\_0\_1

7ea23ca 5 minutes ago 1 commit

 Task\_0\_1.py

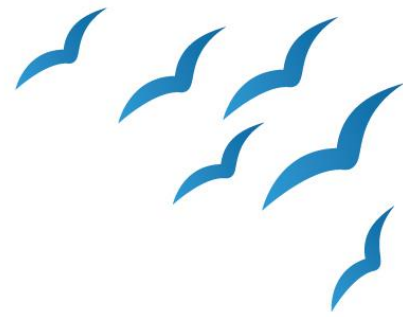
Task\_0\_1

5 minutes ago

Add a README with an overview of your project.

[Add a README](#)

# Задание



Создать файл Task\_0\_2

Написать в нем программу ввода двух чисел  $x$  и  $y$ , печатающую наибольшее число из них.

Отладить программу проверить на нескольких парах чисел, что она работает правильно.

Вопрос. Какие пары чисел необходимо проверить?

Выполнить commit и push.

Проверить в GitHub в репозитории, что программа там появилась.