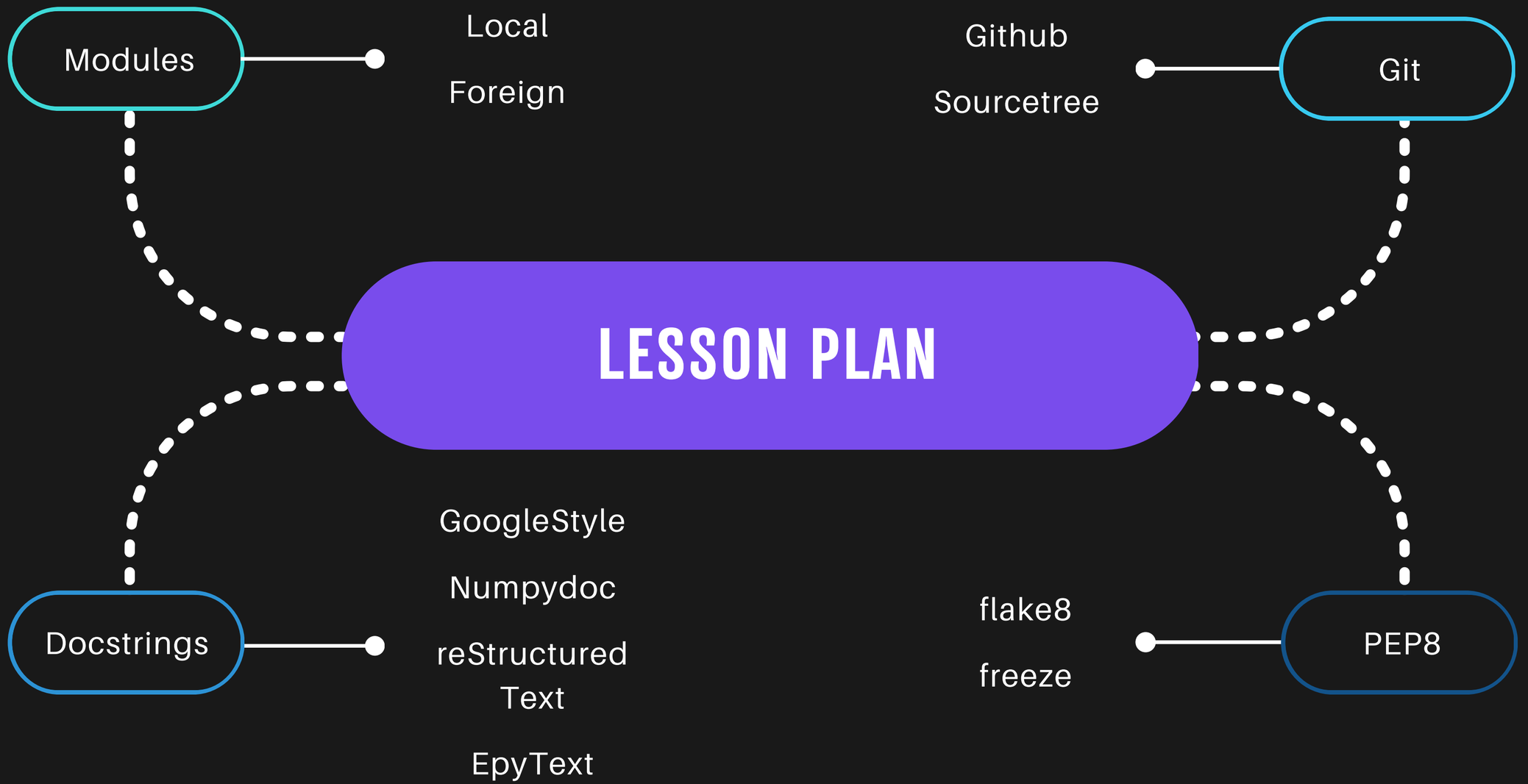
CREATIVE & TECH PROJECTR ONLINE INSTITUTE

Good programming style

Additional topic

Presented by

Oleksandr Ukrainets


Modules
Local
Foreign
Git
Github
Sourcetree
LESSON PLAN
Docstrings
GoogleStyle
Numpydoc
reStructured Text
EpyText
PEP8
flake8
freeze

Docstring Best Practices

Docstring (Documentation String) у Python являє собою рядок документації, який пояснює, що робить функція, метод, модуль або клас, а також дає інструкції з використання.

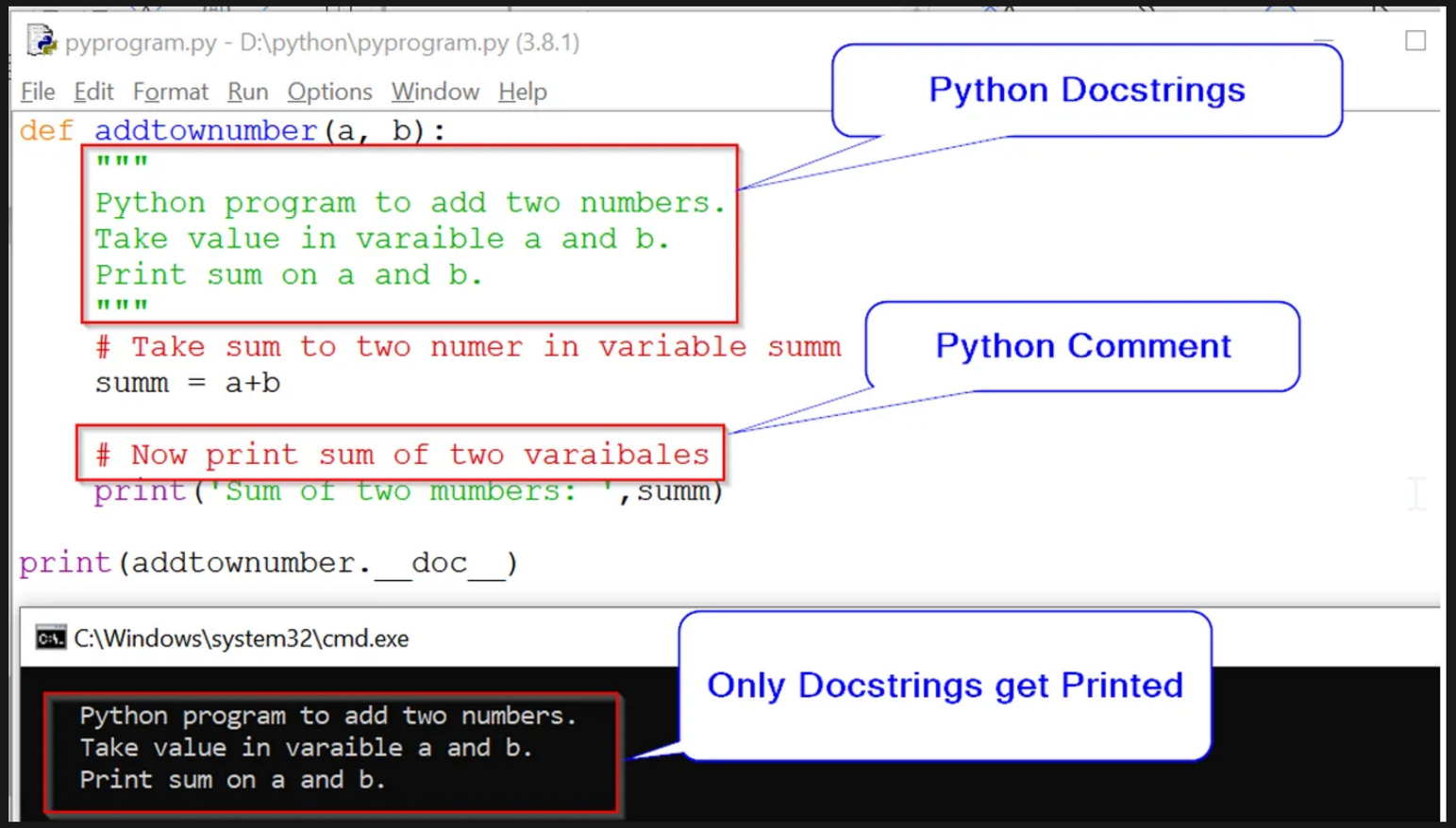
Зазвичай docstring розташовується у верхній частині функції, методу, модуля або класу, і укладається в потрійні лапки.

What's docstring for?

```python
# Коментар до функції
def generate_password(password_len: int = 10) -> str:
    '''…

    if not isinstance(password_len, int):
        raise TypeError('Invalid Type...')

    choices = string.ascii_letters + string.digits + '#$%^'
    result = ''

    for _ in range(password_len):
        result += random.choice(choices)

    return result
```

Docstringformats

1. GoogleStyle
2. Numpydoc



*reStructuredText, EpyText

GoogleStyle

```
def generate_password(password_len: int = 10) -> str:
    '''
    * Description of what the function does.
    Generate password

    * Parameters in input
    Args:
        password_len(int, optional): Description of arg_1 that can break onto the next line  if needed

    * What function return?
    Returns:
    str: generate password

    * Any raise?
    Raises:    ValueError: Include any error types that the function intentionally raises.

    Notes:
    Read more about docstings https://www.programiz.com/python-programming/docstrings
    for more info.
    '''
```

Numpydoc

```python
def translate_in_cel(temp: float, var = 0) -> float:
    '''
    Description of what the function does.
    Temperature convert to cel

    Parameters
    ----------
        arg_1 : expected type of arg_1
                Description of arg_1.
        temp:  float
               The data to convery

        arg_2 : int, optional
                Write optional when an argument has a default value.   Default=default_value.
        var : int, optional
              Default=0

    Returns
    -------
    The type of the return value
        Can include a description of the return value.
    float: result of convert.
    '''
```

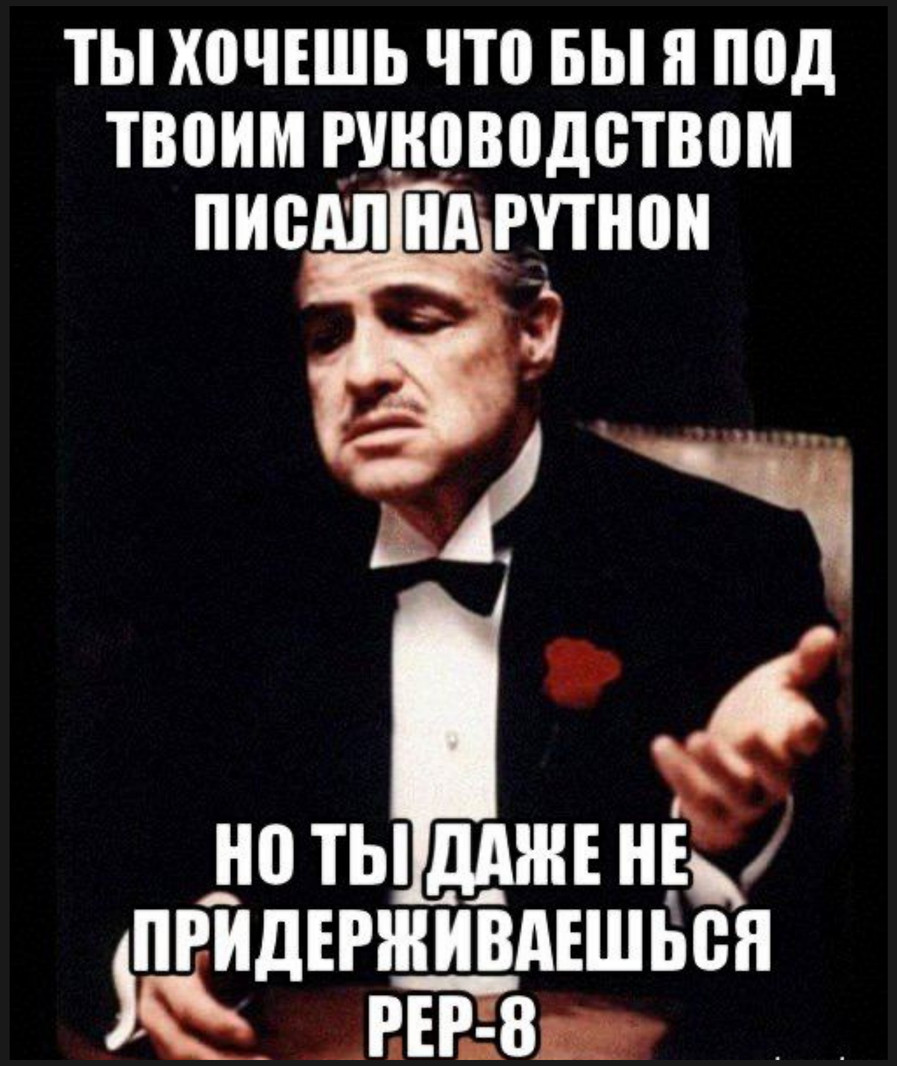
Best Practice

funct.__doc__ - return r - strings

use library: inspect

```
import inspect
print(inspect.getdoc(func_name))
```

PEP8


ТЫ ХОЧЕШЬ ЧТО БЫ Я ПОД
ТВОИМ РУКОВОДСТВОМ
ПИСАЛ НА PYTHON
НО ТЫ ДАЖЕ НЕ
ПРИДЕРЖИВАЕШЬСЯ
PEP-8

Why do we need it?

```
# 3 - var
arr = [ 1, 2, 3 ]

# 4 - var
arr = [
    1, 2,
    3,
]
```

All code is right

```
# Second variant
arr = [
    1,
    2,
    3,
]
```

```
# First variant to create
arr = [
    1, 2, 3
]
```

```python
def sim_pearson(prefs,p1,p2):
    si = {}
    for item in prefs[p1]:
        if item in prefs[p2]:
            si[item]=1
            n = len(si)
    if len(si) == 0:
        return  0

    sum1=sum([prefs[p1][it] for it in si])
    sum2=sum([prefs[p2][it] for it in si])
    sum1Sq=sum([pow(prefs[p1][it],2) for it in si])
    sum2Sq=sum([pow(prefs[p2][it],2) for it in si])
    pSum=sum([prefs[p1][it]*prefs[p2][it] for it in si])
    num = pSum - (sum1 * sum2 / n)

    den = sqrt((sum1Sq - pow(sum1, 2) / n) * (sum2Sq - pow(sum2, 2) / n))
    if den == 0:
        return 0
    r = num / den
    return r
```

```
:231 missing whitespace after ','
:231 missing whitespace after ','
:225 missing whitespace around operator
:271 multiple spaces after keyword
:225 missing whitespace around operator
:225 missing whitespace around operator
E225 missing whitespace around operator
E231 missing whitespace after ','
E225 missing whitespace around operator
E231 missing whitespace after ','
:225 missing whitespace around operator
F821 undefined name 'sqrt'
W292 no newline at end of file
```

What problems can solve flake8?

1. Зменшити кількість багів
2. Отримати складність коду
3. Забезпечити універсальність коду

Quickstart flake8

python -m pip install flake8	Install flake8
flake8 name_of_file	Use flake8

More information: https://flake8.pycqa.org/en/latest/

FREEZE

```
python3 -m pip freeze > requirements.txt
```

What it consist

package_name==x.y.z

x - major version

y - minor version

z - patch version

Example uwsgi==2.0.*

Tune

1. Replace == with >=
2. Add maximum version: lxml>=2.2.0,<2.3.0

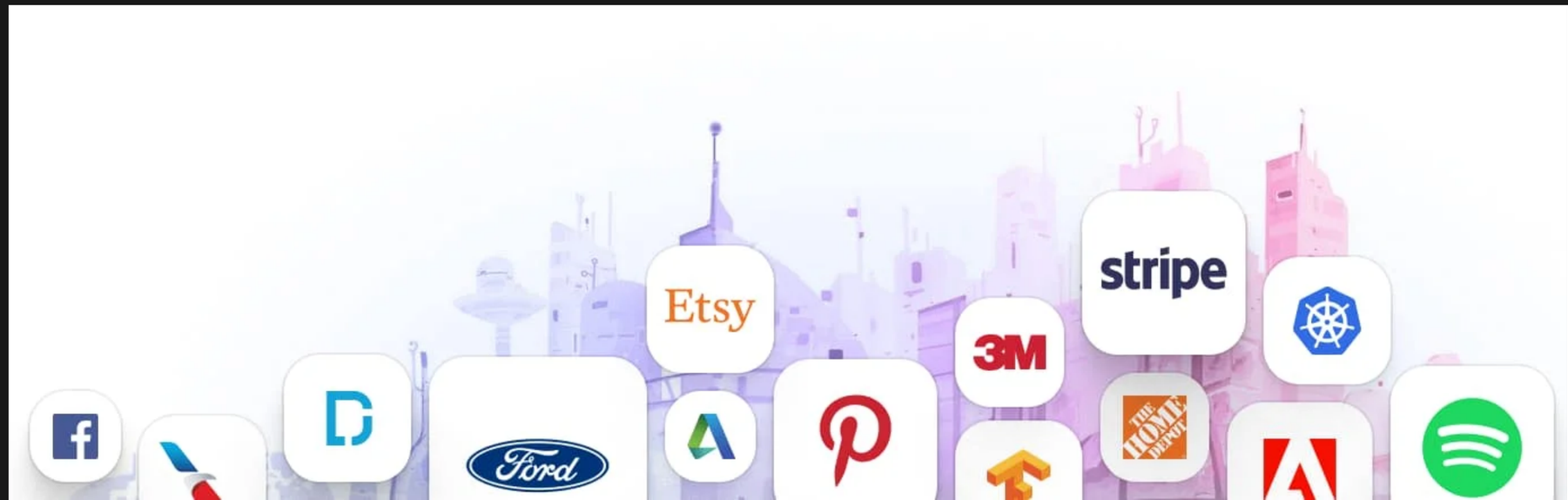
Install

```
python3 -m pip install -U -r requirements.txt
```

Git — система керування версіями з розподіленою архітектурою. На відміну від колись популярних систем на кшталт CVS і Subversion (SVN), де повна історія версій проєкту доступна лише в одному місці, у Git кожна робоча копія коду сама по собі є репозиторієм. Це дає змогу всім розробникам зберігати історію змін у повному обсязі.

Розробка в Git орієнтована на забезпечення високої продуктивності, безпеки та гнучкості розподіленої системи.

Companies that use git



Why do we need it?

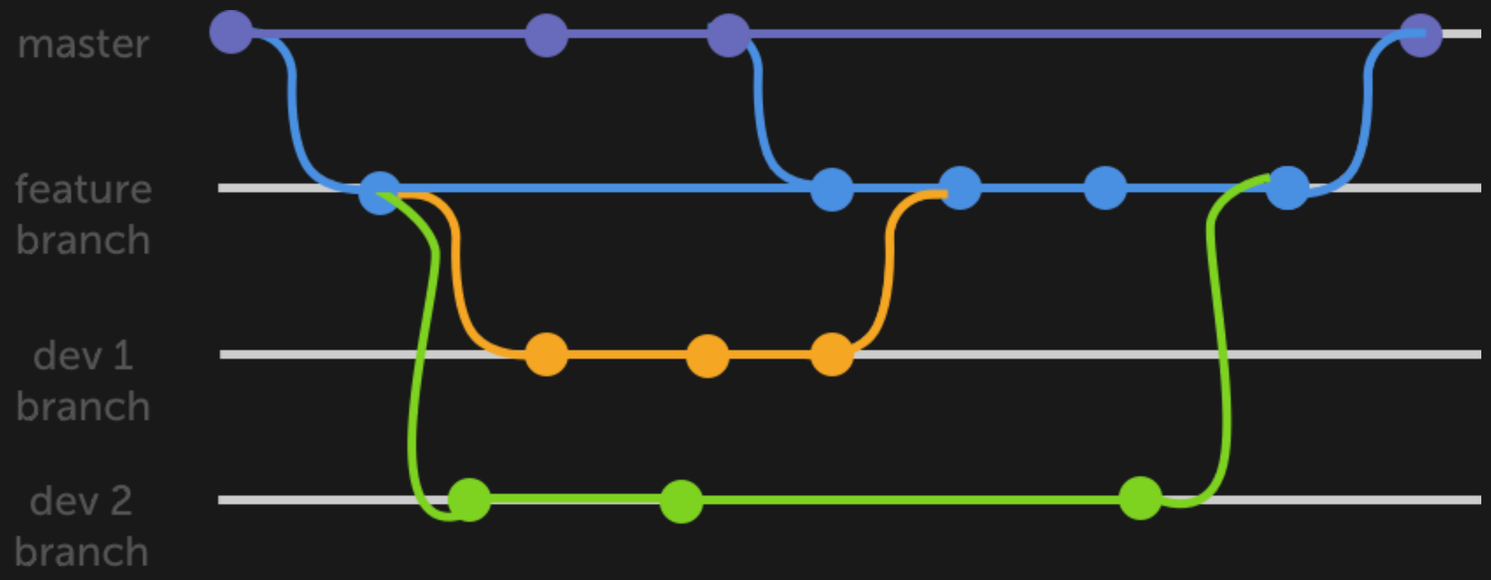
Git — це найпоширеніша система контролю версій. Git відстежує зміни, які ви вносите у файли, щоб у вас був запис про те, що було зроблено, і ви могли повернутися до певних версій, якщо вам це знадобиться.

Git також полегшує спільну роботу, дозволяючи об'єднувати зміни, внесені кількома людьми, в одне джерело.

Тож незалежно від того, пишете ви код, який бачитимете лише ви, чи працюєте в команді, Git буде вам корисним.

Why do we need it?

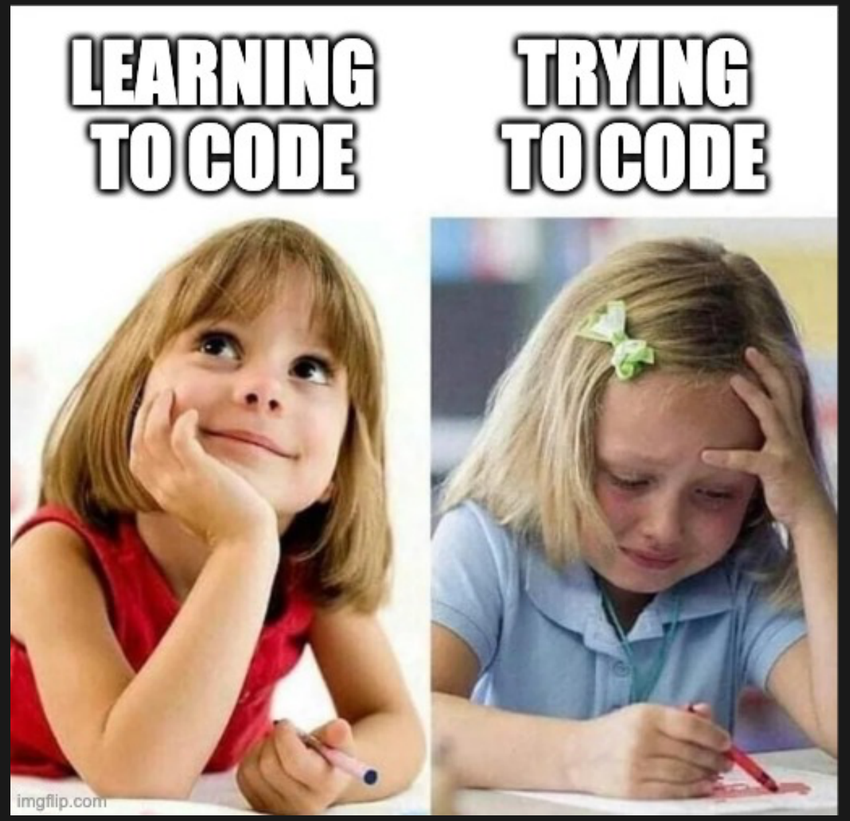
Git відкриває можливість працювати багатьом людям над одним проектом.
Ви можете створювати необмежену кількість розгалужень, які будуть відмінні від master гілки.Що дасть вам змогу не боятися зіпсувати код головної гілки та тестувати функціонал.



Github guide

https://docs.github.com/en/get-started/quickstart/hello-world

Thank you!


LEARNING
TO CODE
TRYING
TO CODE
imgflip.com