



МИЭТ

Национальный исследовательский университет «МИЭТ»

Кафедра ПКИМС

Компьютерные технологии в научных исследованиях

Семинар №7

Язык Python. Основные сведения

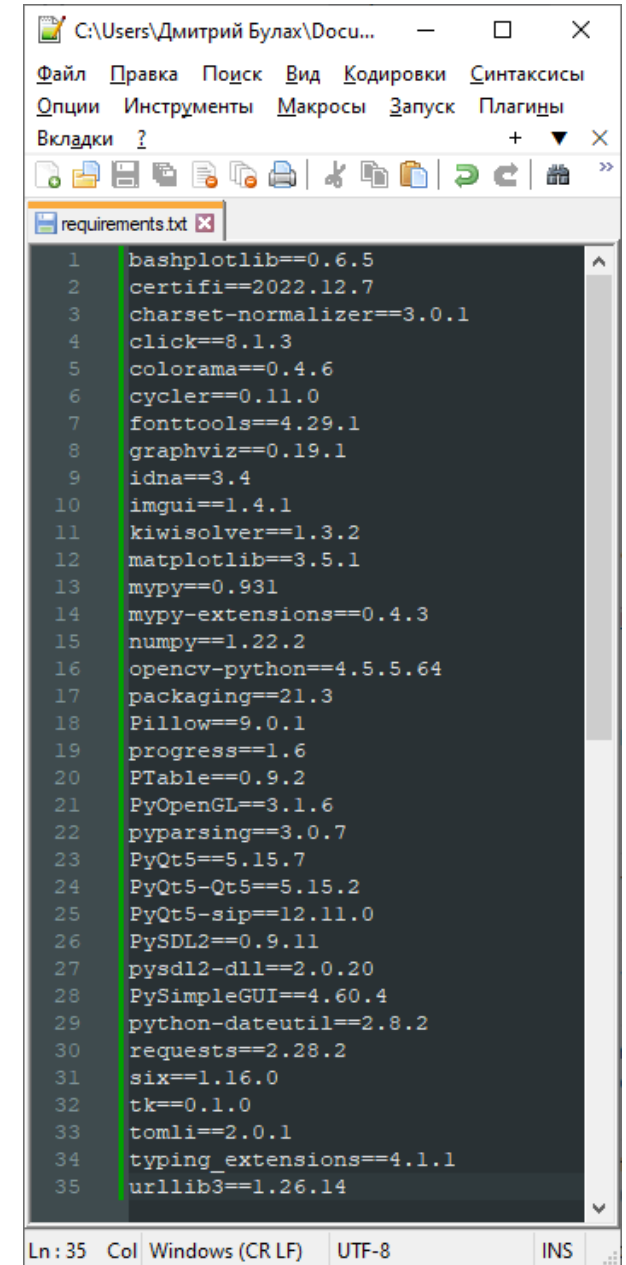
pip freeze: получение списка пакетов в формате «requirements»

```
bashplotlib==0.6.5
certifi==2022.12.7
charset-normalizer==3.0.1
click==8.1.3
colorama==0.4.6
cyclr==0.11.0
fonttools==4.29.1
graphviz==0.19.1
idna==3.4
imgui==1.4.1
kiwisolver==1.3.2
matplotlib==3.5.1
mypy==0.931
mypy-extensions==0.4.3
numpy==1.22.2
opencv-python==4.5.5.64
packaging==21.3
Pillow==9.0.1
progress==1.6
PTable==0.9.2
PyOpenGL==3.1.6
pyparsing==3.0.7
PyQt5==5.15.7
PyQt5-Qt5==5.15.2
PyQt5-sip==12.11.0
PySDL2==0.9.11
pysdl2-dll==2.0.20
PySimpleGUI==4.60.4
python-dateutil==2.8.2
requests==2.28.2
six==1.16.0
tk==0.1.0
tomli==2.0.1
typing_extensions==4.1.1
urllib3==1.26.14
```

pip install: установка заданного перечня пакетов

C:\Users\Дмитрий Булах>pip freeze > requirements.txt

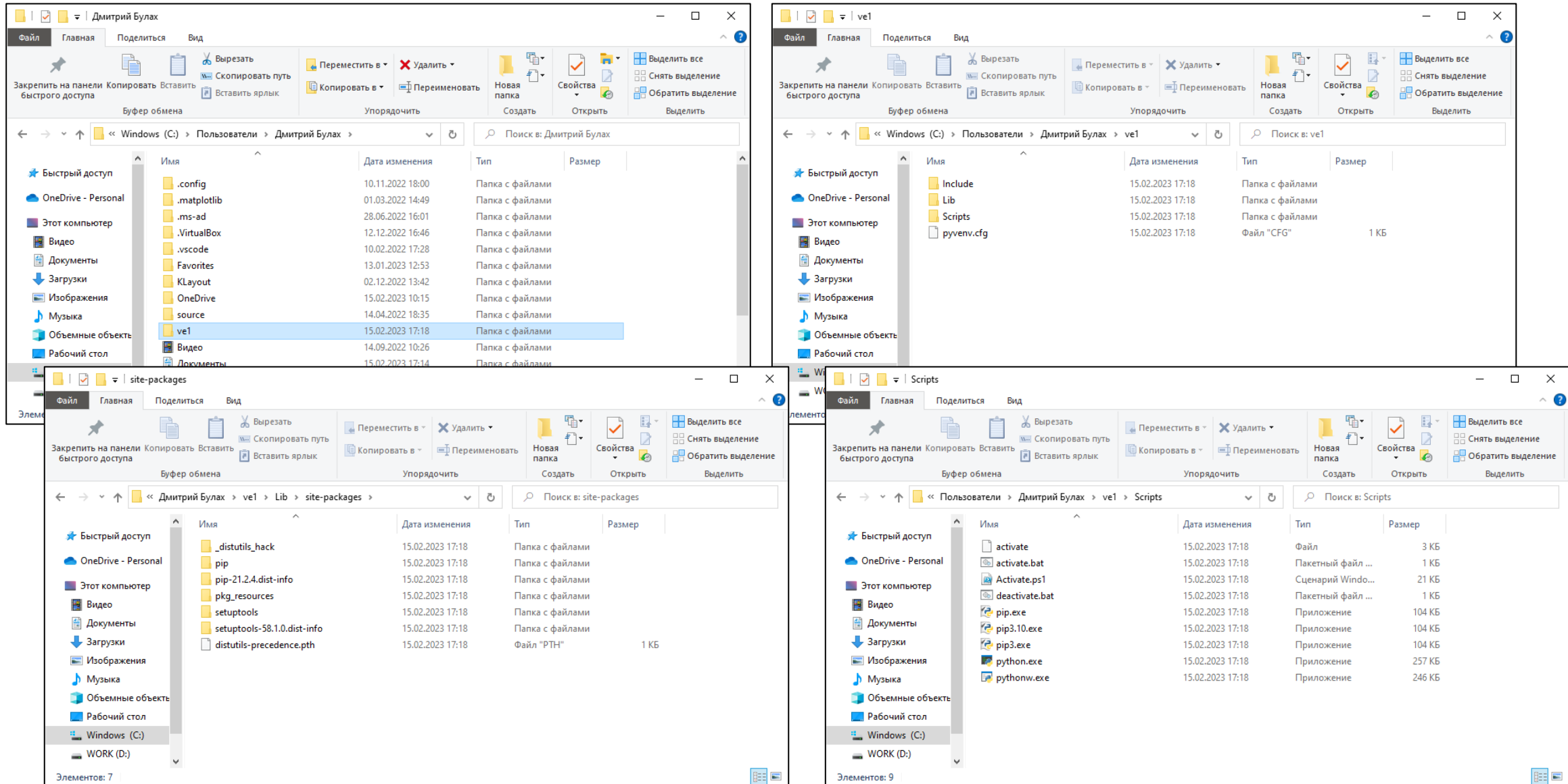
C:\Users\Дмитрий Булах>pip install -r requirements.txt



```
1 bashplotlib==0.6.5
2 certifi==2022.12.7
3 charset-normalizer==3.0.1
4 click==8.1.3
5 colorama==0.4.6
6 cyclер==0.11.0
7 fonttools==4.29.1
8 graphviz==0.19.1
9 idna==3.4
10 imgui==1.4.1
11 kiwisolver==1.3.2
12 matplotlib==3.5.1
13 mypy==0.931
14 mypy-extensions==0.4.3
15 numpy==1.22.2
16 opencv-python==4.5.5.64
17 packaging==21.3
18 Pillow==9.0.1
19 progress==1.6
20 PTable==0.9.2
21 PyOpenGL==3.1.6
22 pyparsing==3.0.7
23 PyQt5==5.15.7
24 PyQt5-Qt5==5.15.2
25 PyQt5-sip==12.11.0
26 PySDL2==0.9.11
27 pysdl2-dll==2.0.20
28 PySimpleGUI==4.60.4
29 python-dateutil==2.8.2
30 requests==2.28.2
31 six==1.16.0
32 tk==0.1.0
33 tomli==2.0.1
34 typing_extensions==4.1.1
35 urllib3==1.26.14
```

Виртуальное окружение в Python: создание

```
C:\Users\Дмитрий Булах>python -m venv ve1
```



Виртуальное окружение в Python: активация

C:\Users\Дмитрий Булах>ve1\Scripts\activate.bat

```
C:\windows\system32\cmd.exe

C:\Users\Дмитрий Булах>python -m venv ve1

C:\Users\Дмитрий Булах>ve1\Scripts\activate.bat

(ve1) C:\Users\Дмитрий Булах>_
```

```
C:\windows\system32\cmd.exe

C:\Users\Дмитрий Булах>python -m venv ve1

C:\Users\Дмитрий Булах>ve1\Scripts\activate.bat

(ve1) C:\Users\Дмитрий Булах>pip list
Package      Version
-----
pip          21.2.4
setuptools   58.1.0
WARNING: You are using pip version 21.2.4; however, version 23.0 is available.
You should consider upgrading via the 'C:\Users\Дмитрий Булах\ve1\Scripts\python.exe -m pip install --upgrade pip' command.

(ve1) C:\Users\Дмитрий Булах>
```

Виртуальное окружение в Python: работа с venv

```
C:\windows\system32\cmd.exe
C:\Users\Дмитрий Булах>ve1\Scripts\activate.bat

(ve1) C:\Users\Дмитрий Булах>pip list
Package      Version
-----
pip          21.2.4
setuptools   58.1.0
WARNING: You are using pip version 21.2.4; however, version 23.0 is available.
You should consider upgrading via the 'C:\Users\Дмитрий Булах\ve1\Scripts\python.exe -m pip install --upgrade pip' command.

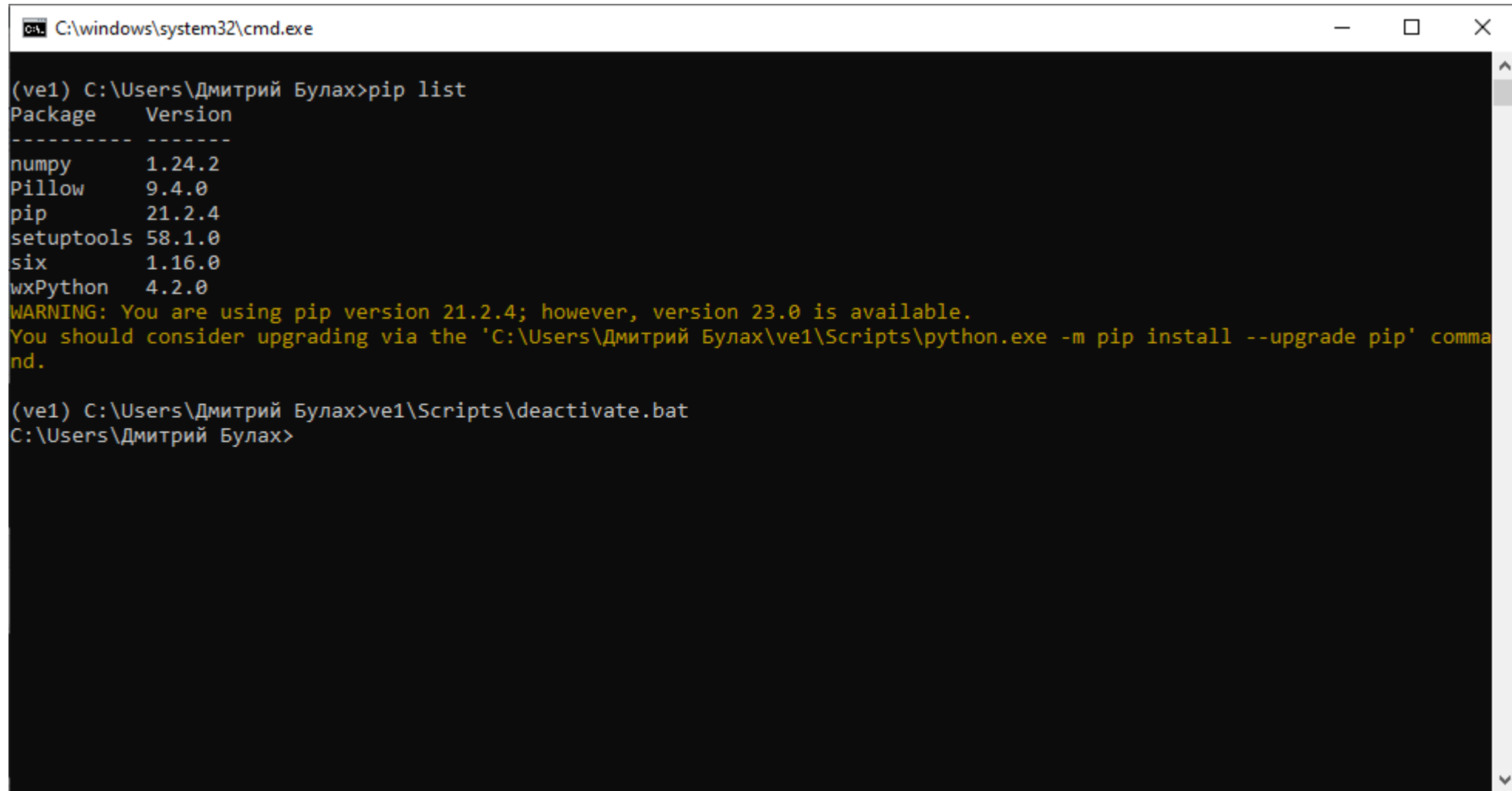
(ve1) C:\Users\Дмитрий Булах>pip install wxpython
```

```
C:\windows\system32\cmd.exe
Collecting wxpython
  Downloading wxPython-4.2.0-cp310-
Collecting pillow
  Downloading Pillow-9.4.0-cp310-
Collecting numpy
  Downloading numpy-1.24.2-cp310-
Collecting six
  Using cached six-1.16.0-py2.py3
Installing collected packages: six, pillow, numpy, wxpython
Successfully installed numpy-1.24.2 pillow-9.4.0 six-1.16.0 wxpython-4.2.0
WARNING: You are using pip version 21.2.4; however, version 23.0 is available.
You should consider upgrading via the 'C:\Users\Дмитрий Булах\ve1\Scripts\python.exe -m pip install --upgrade pip' command.

(ve1) C:\Users\Дмитрий Булах>
```

```
C:\windows\system32\cmd.exe
C:\Users\Дмитрий Булах>pip list
Package      Version
-----
bashplotlib  0.6.5
certifi      2022.12.7
charset-normalizer 3.0.1
click        8.1.3
colorama     0.4.6
cycler       0.11.0
fonttools    4.29.1
graphviz     0.19.1
idna         3.4
imgui        1.4.1
kiwisolver   1.3.2
matplotlib   3.5.1
mypy         0.931
mypy-extensions 0.4.3
numpy        1.22.2
opencv-python 4.5.5.64
packaging    21.3
Pillow       9.0.1
pip          23.0
progress     1.6
PTable       0.9.2
PyOpenGL     3.1.6
pyparsing    3.0.7
PyQt5        5.15.7
PyQt5-Qt5    5.15.2
PyQt5-sip    12.11.0
PySDL2       0.9.11
```

Виртуальное окружение в Python: выход



```
C:\windows\system32\cmd.exe

(ve1) C:\Users\Дмитрий Булах>pip list
Package      Version
-----
numpy        1.24.2
Pillow       9.4.0
pip          21.2.4
setuptools   58.1.0
six          1.16.0
wxPython     4.2.0
WARNING: You are using pip version 21.2.4; however, version 23.0 is available.
You should consider upgrading via the 'C:\Users\Дмитрий Булах\ve1\Scripts\python.exe -m pip install --upgrade pip' command.

(ve1) C:\Users\Дмитрий Булах>ve1\Scripts\deactivate.bat
C:\Users\Дмитрий Булах>
```

Типы данных в Python

В Python есть несколько стандартных типов данных:

- числа (Numbers)
 - целые (Integers)
 - вещественные (Real)
 - комплексные (Complex)
- строки (Strings)
- списки (Lists)
- множества (Sets)
- кортежи (Tuples)
- словари (Dictionaries)
- логический тип данных (Boolean)

Эти типы данных можно, в свою очередь,

классифицировать по нескольким признакам:

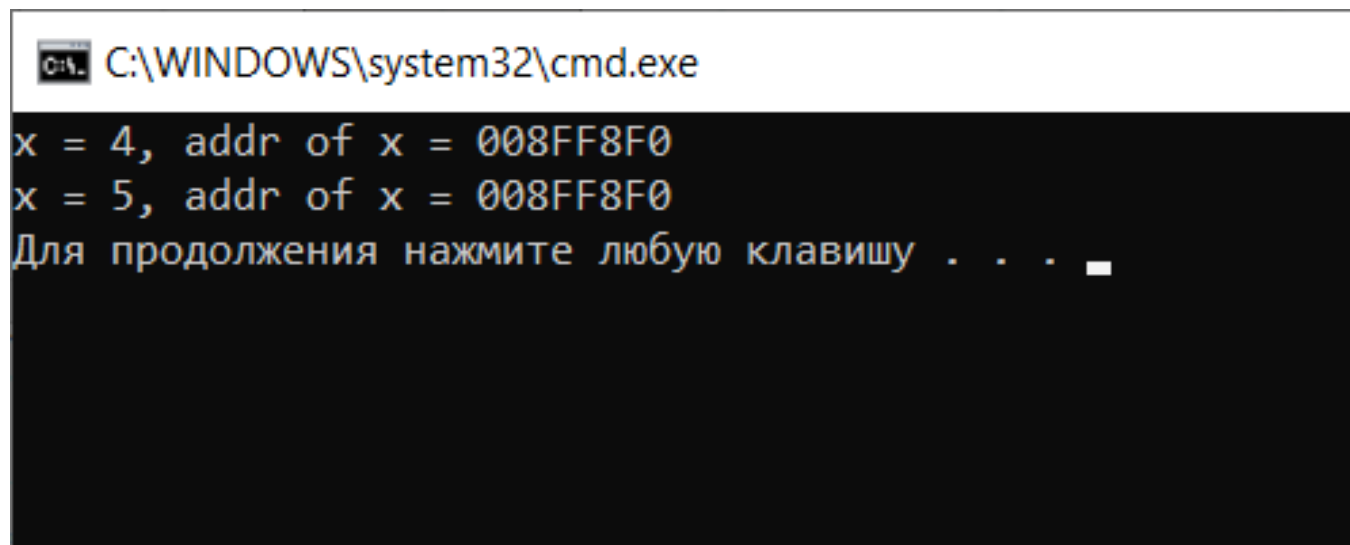
- изменяемые (списки, словари и множества)
- неизменяемые (числа, строки и кортежи)
- упорядоченные (списки, кортежи, строки и словари)
- неупорядоченные (множества)

```
x = 42  
print(f' value: {x} of type {type(x)}')
```

```
value: 42 of type <class 'int'>
```


Что значит, что тип int – неизменяемый? (1)

```
int main() {  
    int x = 4;  
  
    std::cout << "x = " << x << ", addr of x = " << &x << std::endl;  
  
    x = 5;  
  
    std::cout << "x = " << x << ", addr of x = " << &x << std::endl;  
  
    return 0;  
}
```



C:\WINDOWS\system32\cmd.exe

```
x = 4, addr of x = 008FF8F0  
x = 5, addr of x = 008FF8F0  
Для продолжения нажмите любую клавишу . . .
```

Что значит, что тип int – неизменяемый? (1)

C:\WINDOWS\system32\cmd.exe - python

```
C:\Users\Дмитрий Булах>python
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 4
>>> id(x)
2244390682960
>>>
```

C:\WINDOWS\system32\cmd.exe - python

```
C:\Users\Дмитрий Булах>python
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 4
>>> id(x)
2244390682960
>>> x = 5
>>> id(x)
2244390682992
>>>
```

C:\WINDOWS\system32\cmd.exe - python

```
C:\Users\Дмитрий Булах>python
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 4
>>> id(x)
2244390682960
>>> x = 5
>>> id(x)
2244390682992
>>> x = 4
>>> id(x)
2244390682960
>>>
```

Операции с целыми числами

```
#!/usr/bin/python
```

```
x = 5
```

```
y = 3
```

```
print(5 / 3)
```

```
print(5 // 3)
```

```
print(5 % 3)
```

```
print(5 ** 3)
```

```
1.6666666666666667
```

```
1
```

```
2
```

```
125
```

Операция	Результат
$-x$	Смена знака
$x + y$	Сумма двух чисел
$x - y$	Разность двух чисел
$x * y$	Произведение двух чисел
x / y	Частное чисел
$x // y$	Целая часть от деления
$x \% y$	Остаток от деления
$x ** y$	Возведение x в степень y , x^y

Декораторы Python (1)

```
class Rectangle:
    def __init__(self, a, b):
        self.a = a
        self.b = b
```

```
    def area(self):
        return self.a * self.b
```

```
rect = Rectangle(5, 6)
print(rect.area())
```

```
print(rect.area)
```

```
<bound method Rectangle.area of <__main__.Rectangle object at 0x0000022E074236D0>>
```

```
class Rectangle:
    def __init__(self, a, b):
        self.a = a
        self.b = b

    @property
    def area(self):
        return self.a * self.b
```

```
rect = Rectangle(5, 6)
print(rect.area)
```

Декораторы Python (2)

```
def decorator_function(func):  
    def wrapper():  
        print('Функция-обёртка!')  
        print(f'Оборачиваемая функция: {func}')  
        print('Выполняем обёрнутую функцию...')  
        func()  
        print('Выходим из обёртки')  
    return wrapper
```

```
def hello_world():  
    print('Hello world!')
```

hello_world()

Hello world!



```
@decorator_function  
def hello_world():  
    print('Hello world!')
```

hello_world()

Функция-обёртка!

Оборачиваемая функция: <function hello_world at 0x000001D7E1CC89A0>

Выполняем обёрнутую функцию...

Hello world!

Выходим из обёртки

Декораторы Python (3)

```
class Circle:
    def __init__(self, radius):
        self._radius = radius

    def get_radius(self):
        return self._radius

    def set_radius(self, value):
        self._radius = value

radius = property(
    fget = get_radius,
    fset = set_radius,
    doc = 'The radius property'
)

property([fget=None, fset=None, fdel=None, doc=None])

circle = Circle(42.0)
circle.radius = 15.6
print(f'Радиус = {circle.radius}')
```

Декораторы Python (4)

```
class Circle:
    def __init__(self, radius):
        self._radius = radius

    def get_radius(self):
        return self._radius

    def set_radius(self, value):
        self._radius = value

radius = property(
    fget = get_radius,
    fset = set_radius,
    doc = 'The radius property'
)
```

```
class Circle:
    def __init__(self, radius):
        self._radius = radius

    @property
    def radius(self):
        return self._radius

    @property.setter
    def radius(self, value):
        self._radius = value
```

Синтаксис языка Python: аннотирование типов данных (1)

```
def RequestGroundH(sensorId : int) -> float:  
    """
```

Функция посылает запрос типа GET для получения значения влажности почвы

URL для запроса: https://dt.miet.ru/ppo_it/api/hum/<number>

Вход : (int) номер датчика в диапазоне [1..6]

Выход: (float, float) значения температуры и влажности воздуха

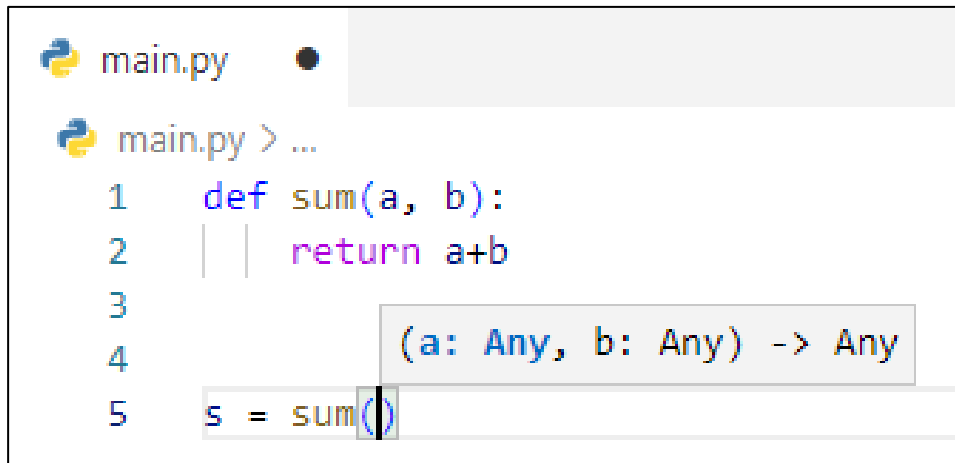
"""

```
ret = requests.get(f'https://dt.miet.ru/ppo_it/api/hum/{sensorId}')
```

```
print(f'Status: {ret.status_code} Answer: {ret.text}')
```

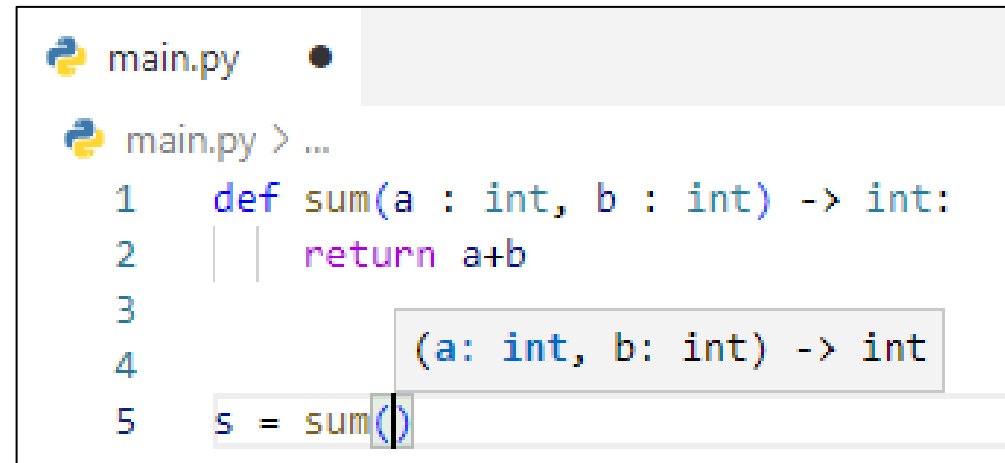
```
json_code = json.loads(ret.text)
```

```
return (float(json_code['humidity']))
```



```
main.py  
  
main.py > ...  
1 def sum(a, b):  
2     return a+b  
3  
4  
5 s = sum()
```

Tooltip: (a: Any, b: Any) -> Any



```
main.py  
  
main.py > ...  
1 def sum(a : int, b : int) -> int:  
2     return a+b  
3  
4  
5 s = sum()
```

Tooltip: (a: int, b: int) -> int

Синтаксис языка Python: аннотирование типов данных (2)

```
class Rectangle:
    def __init__(self, a : int, b : int):
        self.a = a
        self.b = b
```

```
@property
def area(self) -> int:
    return self.a * self.b
```

```
rect = Rectangle(4, 6)
print(rect.area)
```

```
rect = Rectangle(4, 6.8)
print(rect.area)
```

mypy code.py

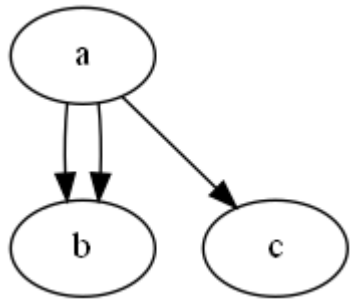
```
● ▶ Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main mypy code.py
  Success: no issues found in 1 source file
○ ▶ Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main █
```

```
○ ▶ Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main mypy code.py
  code.py:10: error: Argument 2 to "Rectangle" has incompatible type "float"; expected "int"
  Found 1 error in 1 file (checked 1 source file)
○ ▶ Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main ERROR █
```

Модули в Python: graphviz

Язык DOT

```
digraph {  
    a -> b  
    a -> b  
    a -> c  
}
```



Язык Python и библиотека graphviz

```
from graphviz import Digraph
```

```
graph = Digraph()
```

```
graph.edge('a', 'b')
```

```
graph.edge('a', 'b')
```

```
graph.edge('a', 'c')
```

```
graph.render('result.dot', format='png', view=True)
```

Модули в Python: numpy



```
import numpy as np
```

```
a = np.array([1, 2, 3])
```

Создание массива по диапазону:

```
a = np.arange(0, 10, 1)  
a = np.arange(0, 1, 0.1)
```

Создание массива по диапазону и числу разбиений:

```
a = np.linspace(0, 2, 7)
```

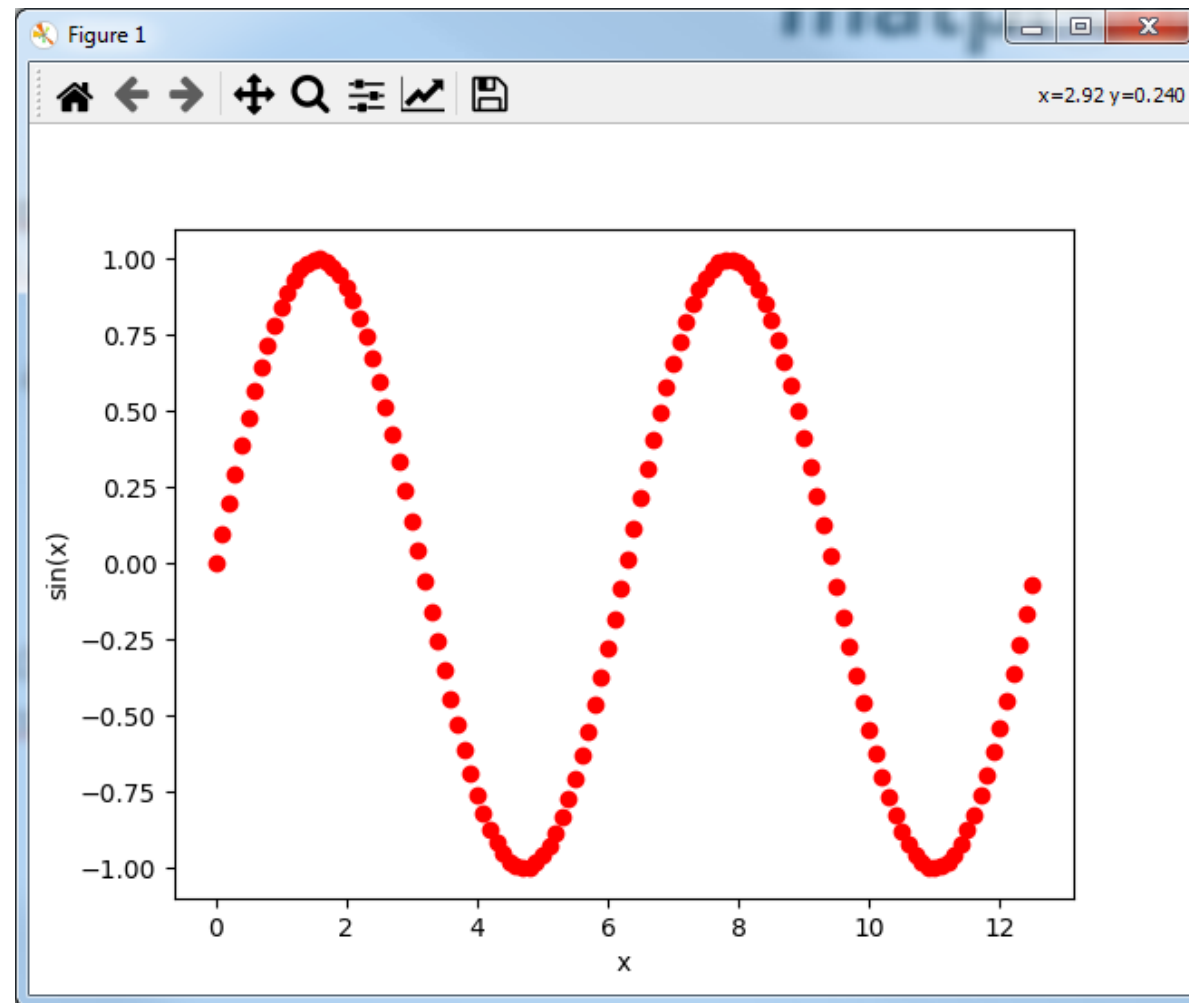
Модули в Python: matplotlib



```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.arange(0, 4*np.pi, 0.1)
y = np.sin(x)
```

```
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.plot(x, y, 'ro')
plt.show()
```



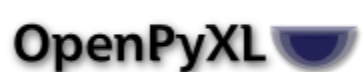
Модули в Python: работа с Excel



```
import pandas as pd

df = pd.read_excel('file.xlsx')

cell_value = df.iloc[0, 0]
print(f"The value in cell C9 is: {cell_value}")
```



```
import openpyxl

workbook = openpyxl.load_workbook(filename='file.xlsx')

sheet = workbook.active

cell_value = sheet['A1'].value
print(f"Value in A1: {cell_value}")
```

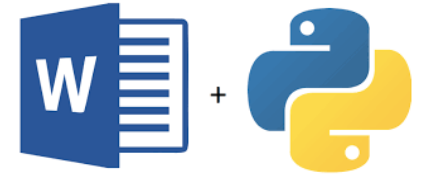
Модули в Python: работа с Word

```
from docx import Document

document = Document('document.docx')

print("Paragraphs:")
for para in document.paragraphs:
    print(para.text)

print("\nTables:")
for table in document.tables:
    for row in table.rows:
        print("|".join(cell.text for cell in row.cells))
```



Модули в Python: click

```
import click
```

```
@click.command()
```

```
@click.argument('num', type=int)
```

```
@click.option('--arg', type=int, help='Simple arg')
```

```
@click.option('--enable', is_flag=True)
```

```
def main(num, arg, enable):
```

```
    print(f'Num      : {num} of type {type(num)}')
```

```
    print(f'Arg      : {arg} of type {type(arg)}')
```

```
    print(f'Enable   : {enable}')
```

```
if __name__ == '__main__':
```

```
    main()
```

```
Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main python .\code.py 42 --arg=5 --enable
Num      : 42 of type <class 'int'>
Arg      : 5 of type <class 'int'>
Enable   : True
Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main python .\code.py 42 --enable
Num      : 42 of type <class 'int'>
Arg      : None of type <class 'NoneType'>
Enable   : True
Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main python .\code.py 42 --arg=5
Num      : 42 of type <class 'int'>
Arg      : 5 of type <class 'int'>
Enable   : False
Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main python .\code.py --arg=5 --enable
Usage: code.py [OPTIONS] NUM
Try 'code.py --help' for help.

Error: Missing argument 'NUM'.
Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main USAGE
```

Модули в Python: colorama

```
from colorama import Fore, Back
```

```
def main():  
    print(Fore.RED + 'Error!')  
    print(Back.GREEN + Fore.YELLOW + 'Hello World!')
```

```
if __name__ == '__main__':  
    main()
```

```
● ▶ Дмитрий Булах@DESKTOP-VNE4M0R C: > cd > htdocs > main python .\code.py  
Error!  
Hello World!  
○ ▶ Дмитрий Булах@DESKTOP-VNE4M0R C: > cd > htdocs > main
```


Модули в Python: PrettyTable

```
from prettytable import PrettyTable
```

```
def main():
```

```
    pt = PrettyTable()
```

```
    pt.field_names = ['No.', 'Student', 'Group', 'Rating']
```

```
    pt.add_row([1, 'Ivanov', 'EN-24', '87'])
```

```
    pt.add_row([2, 'Petron', 'EN-24', '74'])
```

```
    pt.add_row([3, 'Sidorov', 'EN-24', '92'])
```

```
    pt.add_row([3, 'DlinnayaFamiliya', 'EN-24', '92'])
```

```
    print(pt)
```

```
if __name__ == '__main__':
```

```
    main()
```

Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main python .\code.py

No.	Student	Group	Rating
1	Ivanov	EN-24	87
2	Petron	EN-24	74
3	Sidorov	EN-24	92
3	DlinnayaFamiliya	EN-24	92

Дмитрий Булах@DESKTOP-VNE4M0R C: > htdocs \main

Модули в Python: PyQt5

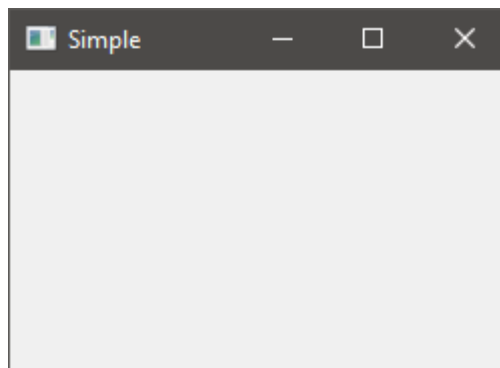


```
import sys
from PyQt5.QtWidgets import QApplication, QWidget

if __name__ == '__main__':
    app = QApplication(sys.argv)

    win = QWidget()
    win.resize(250, 150)
    win.move(300, 300)
    win.setWindowTitle('Simple')
    win.show()

    sys.exit(app.exec_())
```



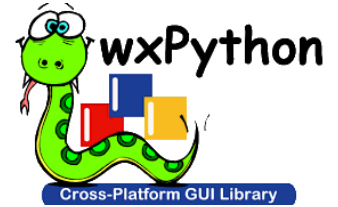
```
#include <QApplication>
#include <QWidget>
```

```
int main(int argc, char *argv[]) {
    QApplication app(argc, argv);

    QWidget *win = new QWidget;
    win->resize(250, 150);
    win->move(300, 300);
    win->setWindowTitle("Simple");
    win->show();

    return app.exec();
}
```

Модули в Python: wxPython



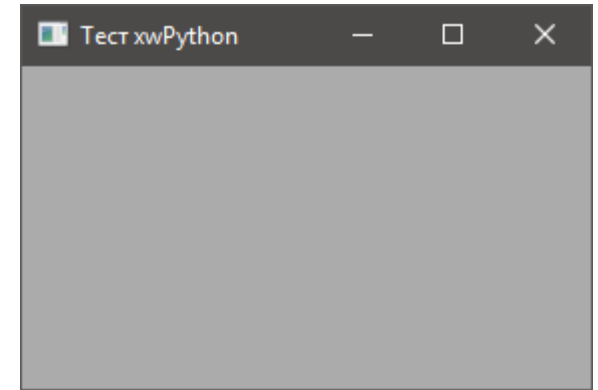
```
import wx

class Example(wx.Frame):
    def __init__(self, title):
        super(Example, self).__init__(None, title=title,
                                       size=(300, 200))

        self.Move((800, 250))

def main():
    app = wx.App()
    ex = Example(title='Тест wxPython')
    ex.Show()
    app.MainLoop()

if __name__ == '__main__':
    main()
```



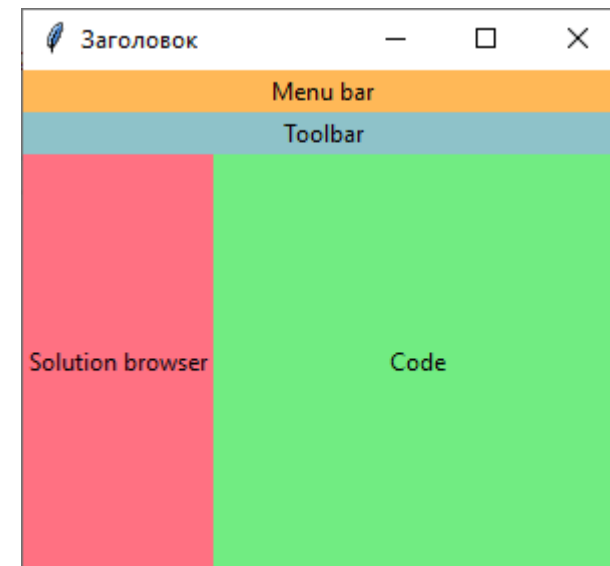
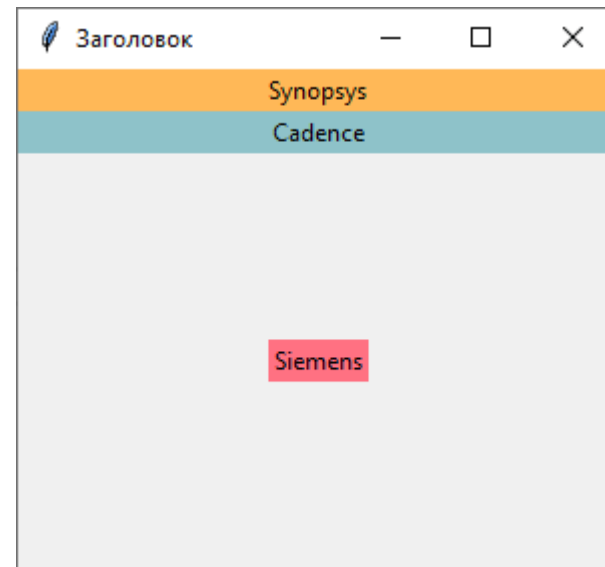
Модули в Python: tkinter

```
import tkinter as tk
```

```
window = tk.Tk()
window.title("Заголовок")
window.resizable(True, False)
window.geometry("300x250+10+10")
```

```
l1 = tk.Label(window, text="Synopsys", bg="#FFB857")
l1.pack(fill='x')
l2 = tk.Label(window, text="Cadence", bg="#8EC2C9")
l2.pack(fill='x')
l3 = tk.Label(window, text="Siemens", bg="#FF7182")
l3.pack(expand='yes')
```

```
l1 = tk.Label(window, text="Menu bar", bg="#FFB857")
l1.pack(fill='x')
l2 = tk.Label(window, text="Toolbar", bg="#8EC2C9")
l2.pack(fill='x')
l3 = tk.Label(window, text="Solution browser", bg="#FF7182")
l3.pack(side='left', fill='y')
l4 = tk.Label(window, text="Code", bg="#71EC82")
l4.pack(expand='yes', fill='both')
```



Модули в Python: pySimpleGui



```
import PySimpleGUI as sg

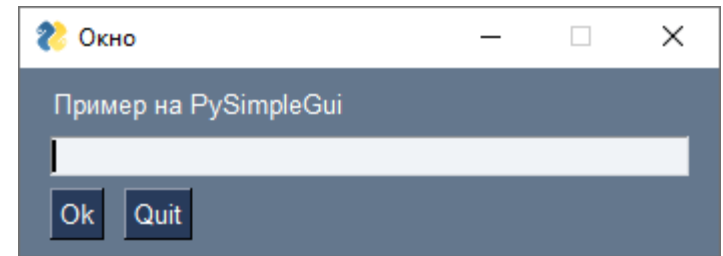
layout = [
    [sg.Text("Пример на PySimpleGui")],
    [sg.Input()],
    [sg.Button('Ok'), sg.Button('Quit')]
]

window = sg.Window('Окно', layout)

while True:
    event, values = window.read()

    if event == sg.WINDOW_CLOSED:
        break

window.close()
```



Модули в Python: MyHDL

```
module inv(x, y);
```

```
    input  x;
```

```
    output y;
```

```
    always@(x)
```

```
        y = ~x;
```

```
endmodule
```

```
entity inv is
```

```
    port(x: in  bit;
```

```
          y: out bit);
```

```
end inv;
```

```
architecture BEH of inv is
```

```
begin
```

```
    process(x)
```

```
    begin
```

```
        y <= not x;
```

```
    end process;
```

```
end RTL;
```

```
from myhdl import *
```

```
@block
```

```
def inv(x, y):
```

```
    @always(x)
```

```
    def logic():
```

```
        y.next = not x
```

```
    return logic
```

Модули в Python: OpenCV

