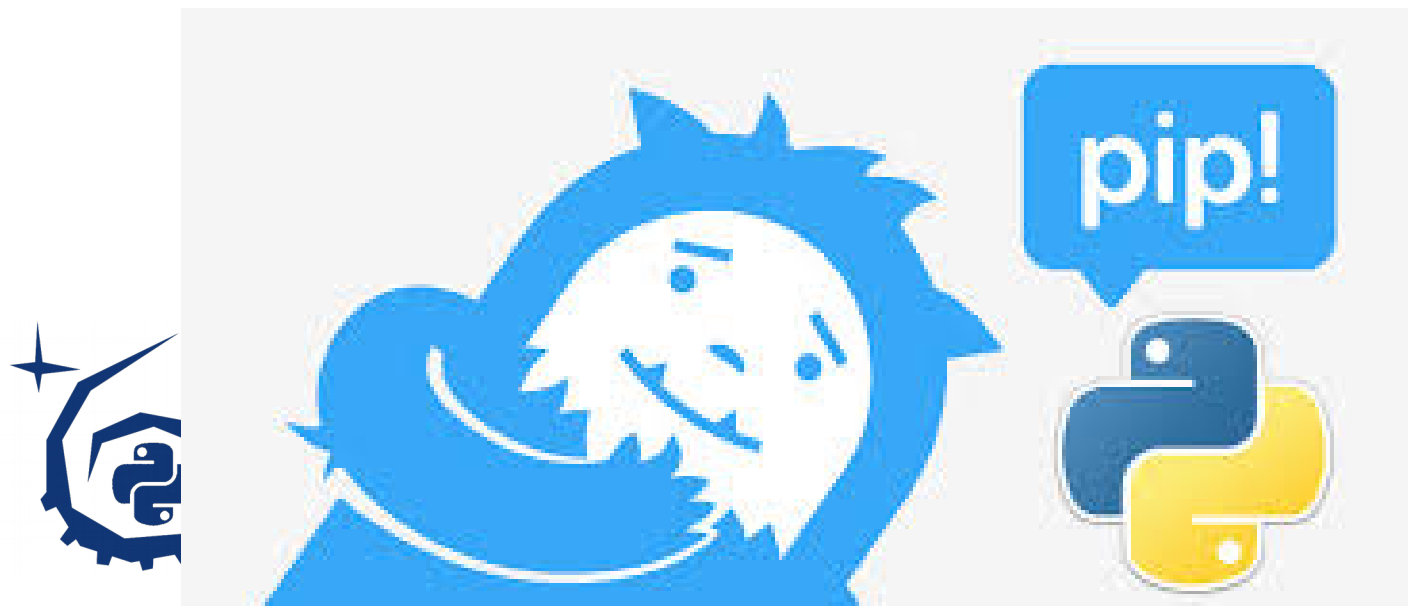


# ПАКЕТЫ МОДУЛИ СНИППЕТЫ



# ИМПОРТ

```
import math # math.sin
```

```
import math as m # m.sin
```

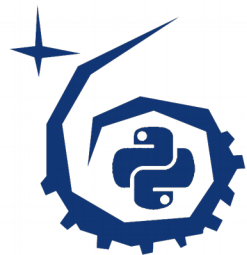
```
from math import * # sin
```

```
from random import randint
```

```
from math import sin as sinus
```

```
from os import path # path.basename
```

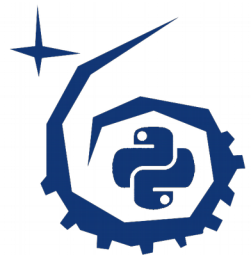
```
From os.path import basename
```



**PEP8: Используйте 4 пробела на каждый уровень отступа.**

# МОДУЛИ

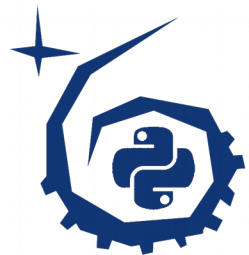
```
import Tkinter  
  
Tkinter._test()
```



**PEP8: Пробелы - самый предпочтительный метод отступов.**

# ВРЕМЕННЫЕ ФАЙЛЫ

```
>>> import tempfile
>>> f = tempfile.mkstemp("_suffix", "prefix_")
>>> print(f)
(3, '/tmp/prefix_0uqa6ydi_suffix')
>>> fh = open(f[1], 'w')
>>> fh.write('Hello!')
6
>>> fh.close()
>>> fh = open(f[1])
>>> print(fh.read())
Hello!
```



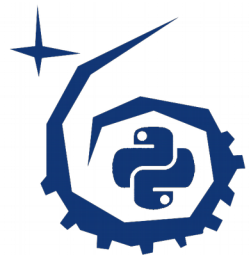
# МОДУЛИ

**встроенные**

```
print (dir (__builtins__))
```

**стандартные**

```
sys.modules
```



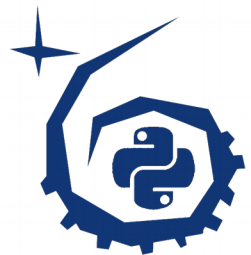
**PEP8: Ограничьте длину строки максимум 79 символами.**

# PYPI

```
pip
```

```
pip install
```

```
pip freeze
```

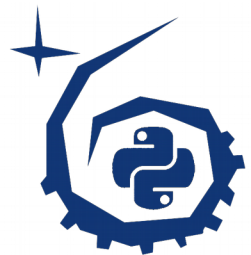


**PEP8: Отделяйте функции двумя пустыми строками.**

# PEP8

```
pip install pep8
```

```
pep8 main.py
```



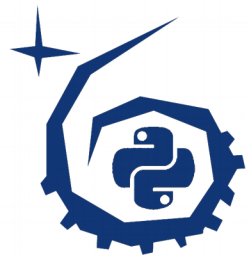
**PEP8: Кодировка Python должна быть UTF-8.**

# PrettyTable

```
from prettytable import PrettyTable

table = PrettyTable(["животное", "свирепость"])

table.add_row(["Оборотень", 100])
table.add_row(["Гризли", 87])
table.add_row(["Кролик из Кэрбенног", 110])
```

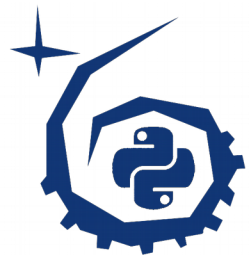


**PEP8: Все идентификаторы обязаны содержать только ASCII символы, и означать английские слова везде, где это возможно.**



**fuzzywuzzy**

```
fuzz.ratio('Этот год был очень хорош!', \  
          'Тот год был не очень плох!')
```



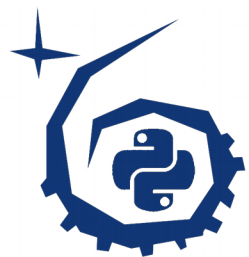
**PEP8: Каждый импорт, как правило, должен быть на отдельной строке.**

# ProgressBar

```
from progressbar import ProgressBar
import time
```

```
pbar = ProgressBar(maxval=10)
pbar.start()
for i in range(1, 11):
    pbar.update(i)
    time.sleep(1)
pbar.finish()
```

```
# 60% |#####
```



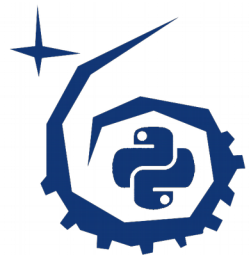
**PEP8:** Импорты всегда помещаются в начале файла, сразу после комментариев к модулю и строк документации, и перед объявлением констант.

# better-exceptions

```
pip install better_exceptions
```

```
export BETTER_EXCEPTIONS=1
```

```
import better_exceptions
```

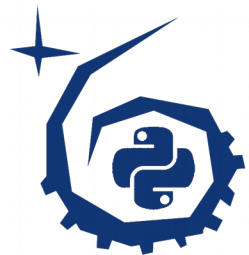


**PEP8: Импорты должны быть сгруппированы в следующем порядке:**

- импорты из стандартной библиотеки
- импорты сторонних библиотек
- импорты модулей текущего проекта.

**fuzzywuzzy**

```
fuzz.ratio('Этот год был очень хорош!', \  
          'Тот год был не очень плох!')
```



**PEP8: Вставляйте пустую строку между каждой группой импортов.**

# GUI

<https://habr.com/post/133337/>

```
import tkinter
```

```
tkinter._test()
```

```
from tkinter import *  
root = Tk()
```

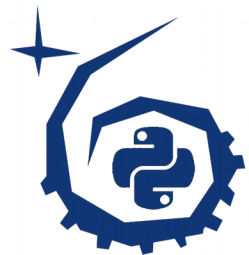
```
def Hello(event):  
    print("Yet another hello world")
```

```
btn = Button(root,                                #родительское окно  
              text="Click me",                    #надпись на кнопке  
              width=30,height=5,                  #ширина и высота  
              bg="white",fg="black")              #цвет фона и надписи
```

```
btn.bind("<Button-1>", Hello)                      #при нажатии ЛКМ на кнопку вызывается функция Hello
```

```
btn.pack()                                         #расположить кнопку на главном окне
```

```
root.mainloop()
```



**PEP8:** Рекомендуется абсолютное импортирование, так как оно обычно более читаемо и ведет себя лучше.

# ГРАФИКИ

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

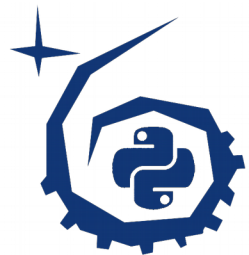
```
x = np.arange(0, 10, 0.2)
```

```
y = np.sin(x)
```

```
fig, ax = plt.subplots()
```

```
ax.plot(x, y)
```

```
plt.show()
```



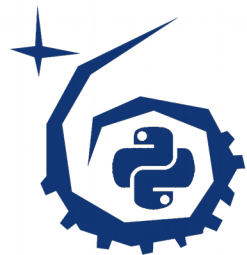
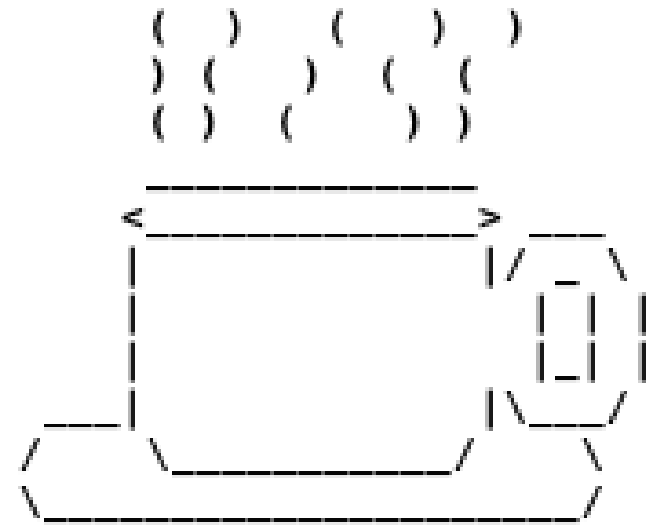
РЕР8: Шаблоны импортов (from import \*) следует избегать, так как они делают неясным то, какие имена присутствуют в глобальном пространстве имён.

# ASCII\_ART

```
from ascii_art import Chart
```

```
def example_one():  
    data = [1, 2, 3, 4, 3, 12, 4, 16, 18, 15, 5, 5, 10]  
    c = Chart(data)  
    print(c.render())
```

```
if __name__ == "__main__":  
    example_one()
```



**PEP8: Избегайте использования пробелов непосредственно перед запятой, точкой с запятой или двоеточием.**



# mimesis

```
pip install mimesis
```

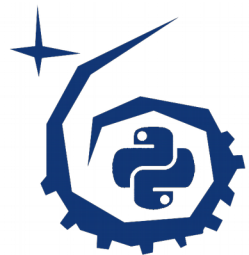
```
>>> from mimesis import Person
```

```
>>> from mimesis.enums import Gender
```

```
>>> def get_card(gender):
```

```
...     person = Person('ru')
```

```
>>> pprint(get_card(Gender.FEMALE))
```



PEP8: Избегайте использования пробелов Использование более <sup>16</sup>  
одного пробела вокруг оператора присваивания (или любого  
другого) для того, чтобы выровнять его с другим.