

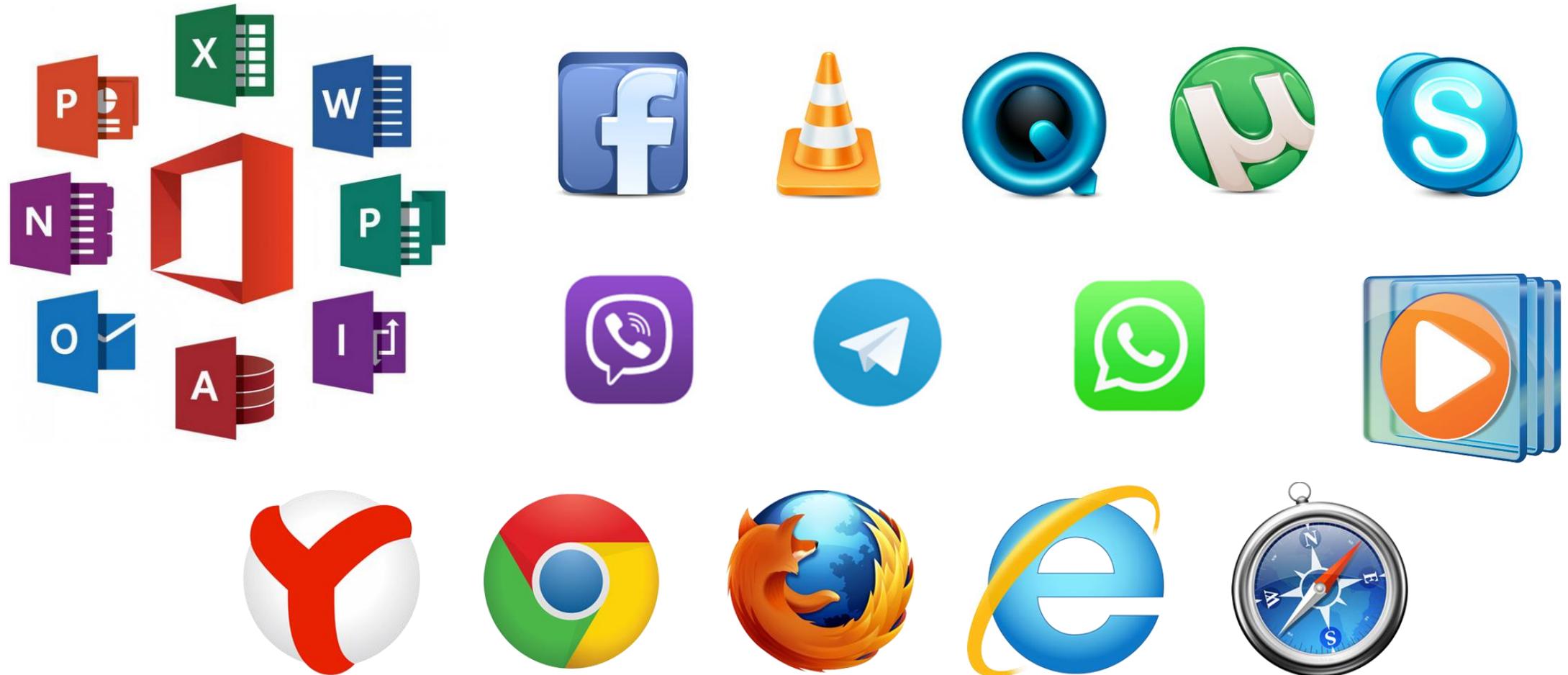
**Яндекс**

**Яндекс**

**Знакомство с программой и с LMS.  
Установка и настройка среды  
программирования  
PER8. Ввод-вывод. Переменные**

Введение

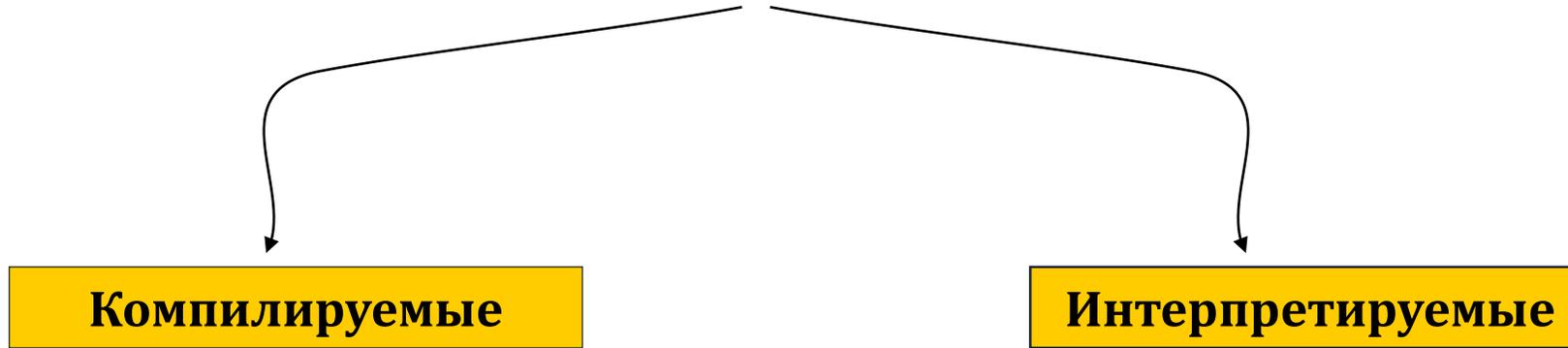
# Программа – это список команд, то есть инструкций для компьютера



**Язык программирования – это набор строгих правил, согласно которым компьютер МОЖЕТ «ПОНИМАТЬ» команды и ВЫПОЛНЯТЬ ИХ.**



# Языки программирования



Если программа написана на компилируемом языке (например, **Pascal** или **C++**), то перед выполнением её нужно сначала полностью проверить на некоторые ошибки и перевести в более «понятную» компьютеру форму.

Если программа написана на интерпретируемом языке (например, **Java** или **Python**), то она выполняется по каждой отдельной команде. Такой подход придаёт языку особую гибкость, и простоту в написании программ.

# Интерпретатор Python



Программа на  
языке Python

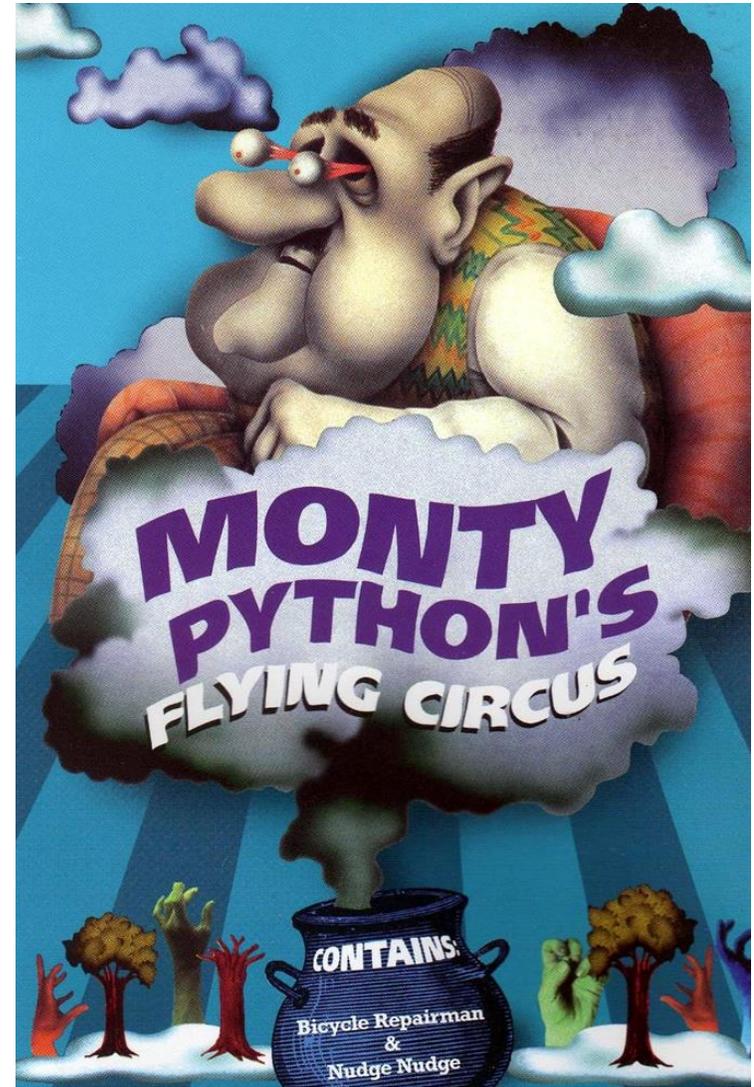


Интерпретатор идет по тексту  
программы, анализирует и переводит  
каждую строку в форму, понятную  
компьютеру



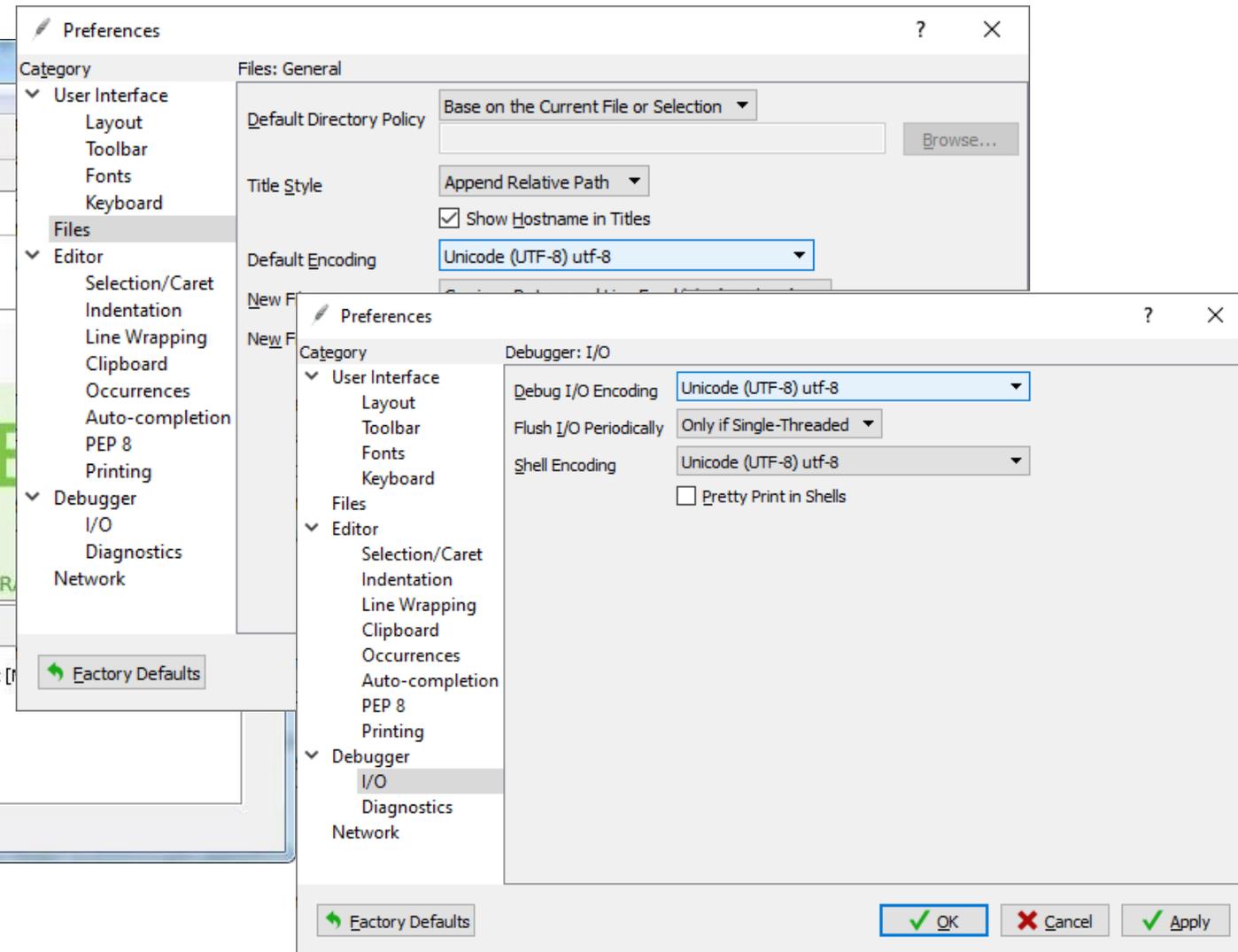
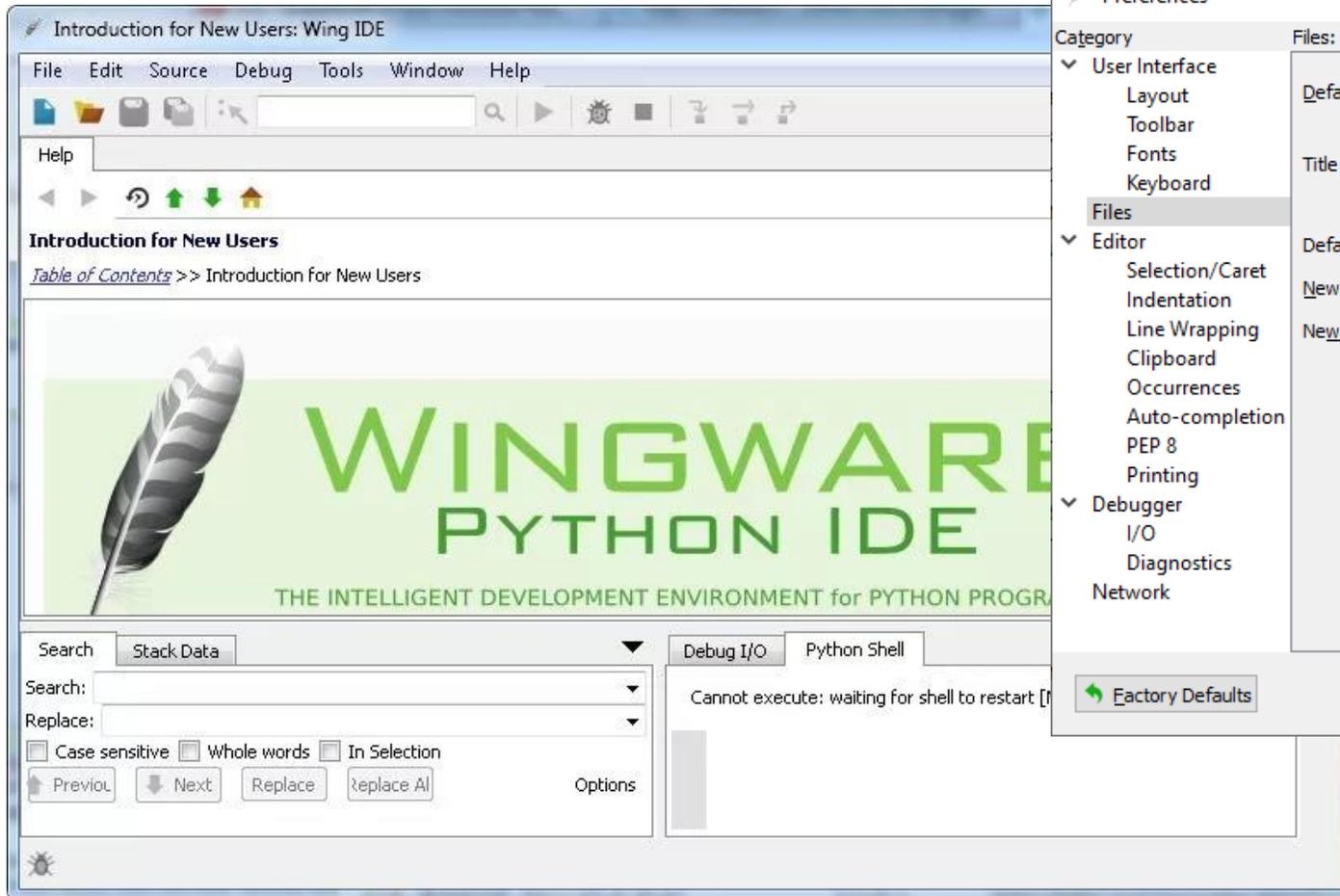
Кажется, мне  
нужно  
напечатать  
приветствие...

# Гвидо Ван Россум



Знакомство с IDE

# Среда разработки (англ. IDE, Integrated Development Environment)





PEP 8

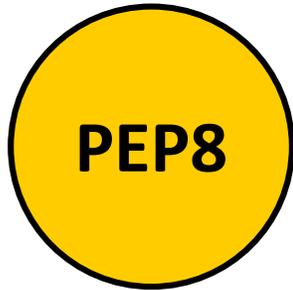
Coding style in Python

*Coding style in Python*

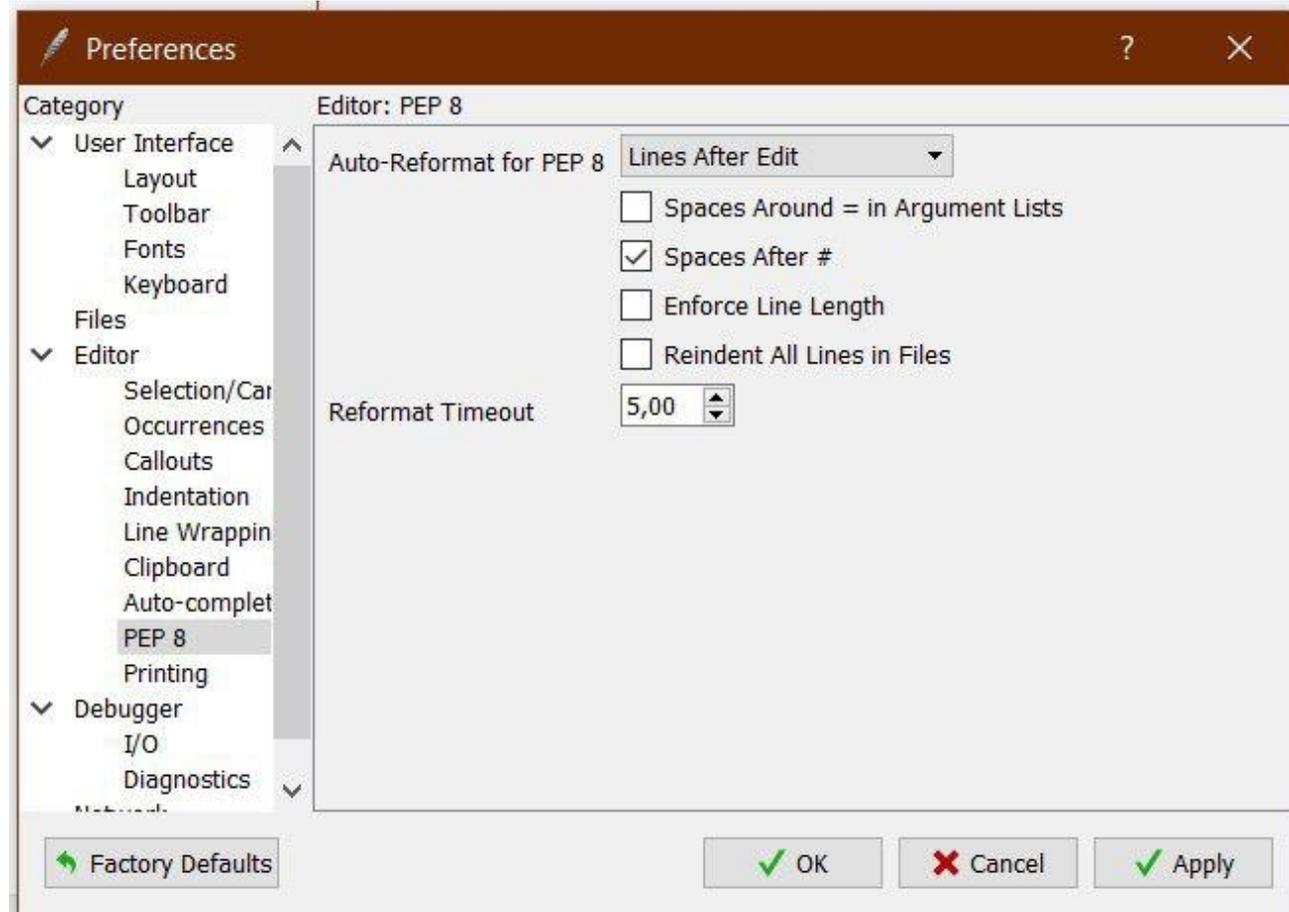


Mecanismo  
Complesso

При оформлении программ, мы будем  
пользоваться PEP8 –  
**(Python Enhanced Proposal)** – это  
документ, описывающий общепринятый  
(рекомендуемый) стиль написания  
программ на языке Python.



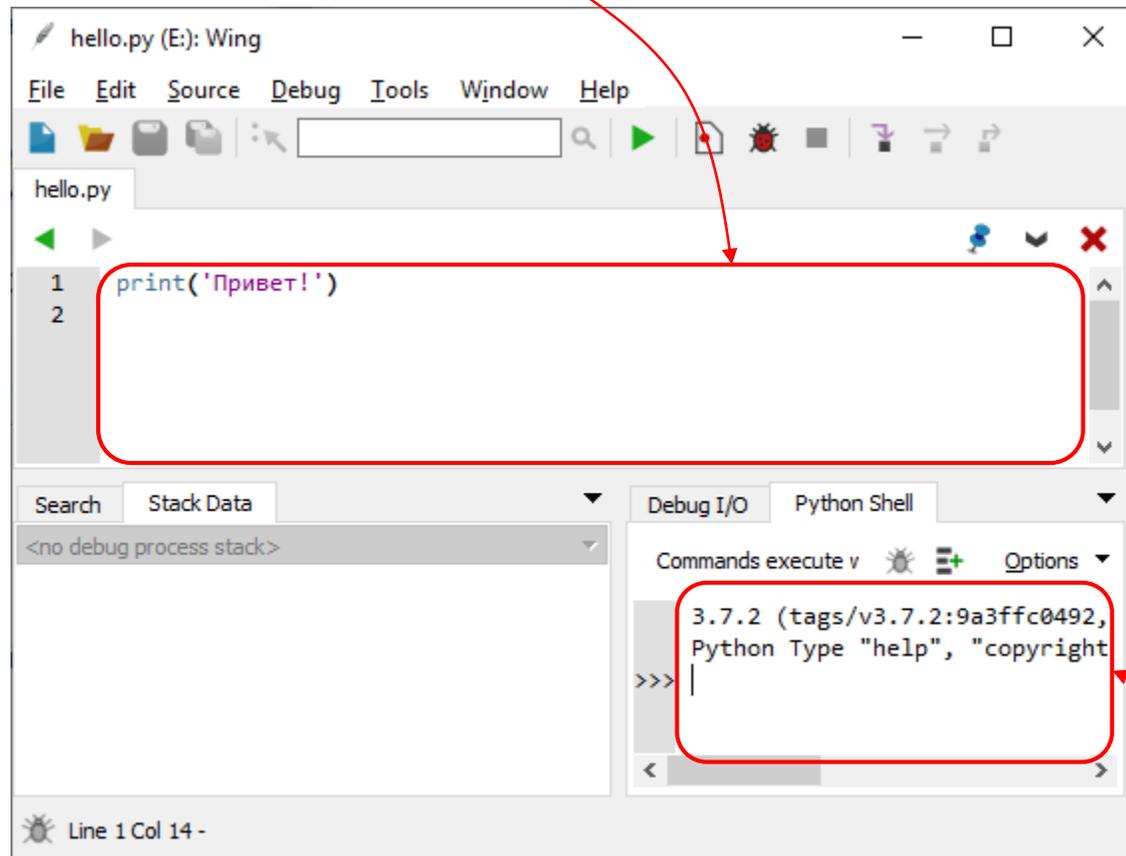
Среды разработки могут помогать с форматированием по **PEP8**. Для включения такой возможности в **Wing IDE** зайдите в меню «**Edit**»→«**Preferences**». Перейдите к категории «**Editor**»→«**PEP8**». Для опции «**Auto-Reformat for PEP 8**» выберите значение «**Lines After Edit**».



Команда print()

# Команда вывода на экран слова "Привет!"

Код – взгляд программиста



Правильно: `print ("Привет! ")`

Неправильно: `print ("Привет!")`

Консоль (ввод, вывод, сообщения об ошибках) –  
взгляд пользователя

Тестирующая система



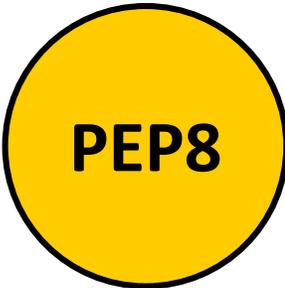
## Задача «Приветствие»

Напишите программу, которая выводит «Привет, Яндекс!»  
(без кавычек) и больше ничего не делает.



## Задача «Знакомство»

Напишите программу, которая выводит на первой строке «Привет, Яндекс!», а на второй – «Приятно познакомиться.»



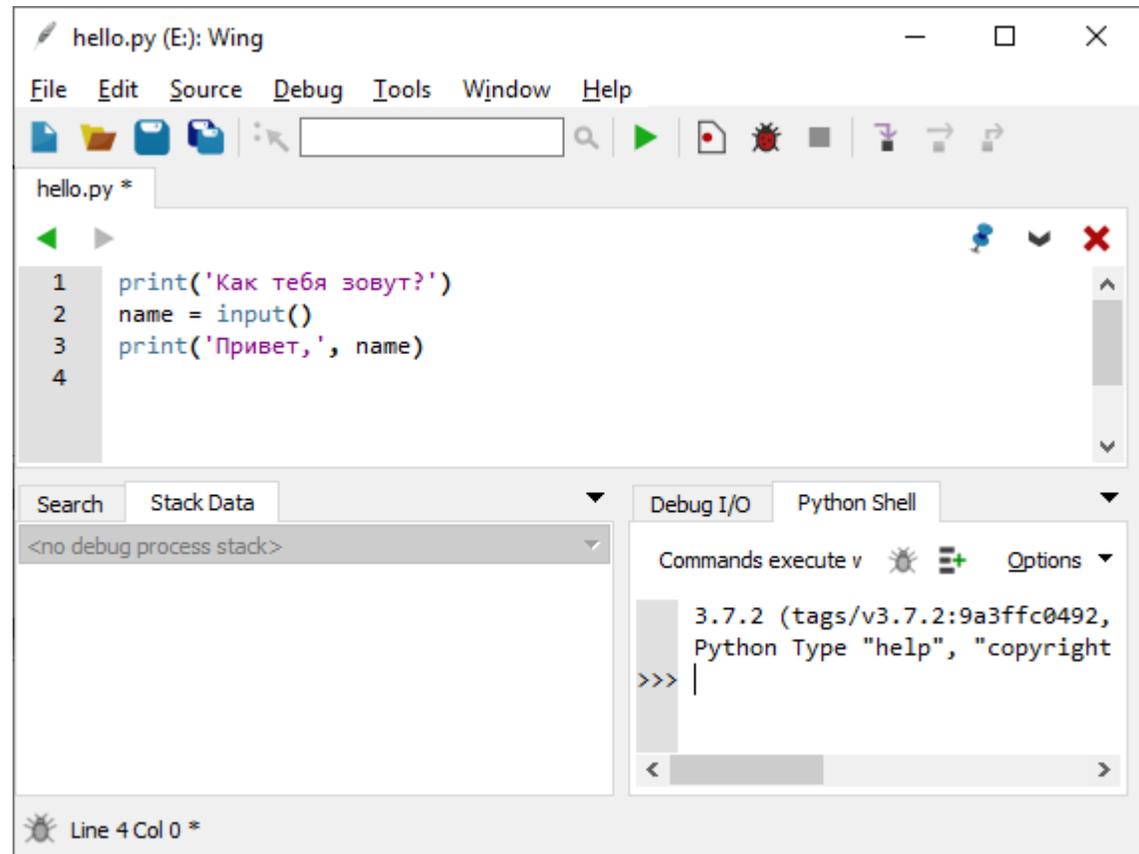
**Правильно:** `print ('Привет', 'Яндекс!')`

**Неправильно:** `print ('Привет', 'Яндекс!')`

Команда input()

# Команда `input()`

```
print('Как тебя зовут?')  
name = input()  
print('Привет,', name)
```



The screenshot shows the Wing IDE interface. The main editor window displays a Python script named 'hello.py' with the following code:

```
1 print('Как тебя зовут?')  
2 name = input()  
3 print('Привет,', name)  
4
```

Below the editor, the 'Python Shell' tab is active, showing the output of the script:

```
3.7.2 (tags/v3.7.2:9a3ffc0492,  
Python Type "help", "copyright  
>>> |
```

The status bar at the bottom indicates 'Line 4 Col 0 \*'.

# **input()** и **print()**

Команда **input()** всегда пишется с круглыми скобками.

Команда работает так:

когда программа доходит до места, где есть **input()**, она ждёт, пока пользователь введёт строку с клавиатуры (ввод завершается нажатием клавиши **Enter**). Введённая строка подставляется на место **input()**.

Если нужно, чтобы программа что-то печатала на экране (и это увидел пользователь) — **print()**.

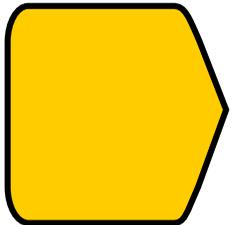
Если нужно, чтобы пользователь что-то напечатал с клавиатуры (чтобы программа могла использовать эти данные) — **input()**.

Переменные

# Переменные

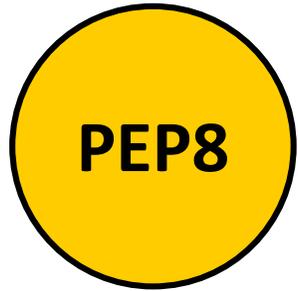
Команду `name = input()` можно считать так –

"подожди, пока пользователь введет какую-то строку и помести введенную строку в переменную `name`".



**Имя переменной** должно отражать ее назначение и может состоять из латинских букв, цифр и символа подчеркивания. Имя не может начинаться с цифры.

# Переменные

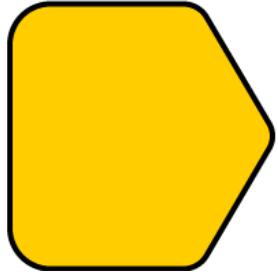


Для именования переменных принято использовать стиль **lower\_case\_with\_underscores** (слова из маленьких букв с подчеркиваниями).

Избегайте использовать такие символы, которые могут не однозначно трактоваться в различных шрифтах, это буквы **o** - большая и маленькая и цифра **0**, буквы **l** маленькая, **l** большая и цифра **1**.

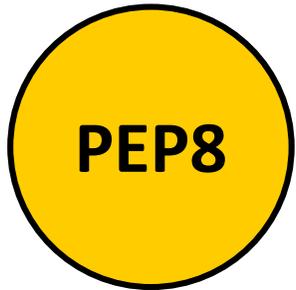
Также нельзя использовать в качестве имени переменной ключевые слова, которые существуют в языке.

# Переменные



**Значение переменной** - это то, что сохраняет в себе переменная. Знак "=" обозначает команду под названием «оператор присваивания». Оператор присваивания присваивает значение, которое находится справа от знака равно, переменной, которая находится слева от знака равно.

```
name = input()
```



Всегда окружайте оператор присваивания одним пробелом с каждой стороны:

**Правильно:** bird = "Тук-тук"

**Неправильно:** bird="Тук-тук"

# Пример

```
print('Какая твоя любимая еда?')  
meal = input()  
print('Да.', meal, '- это вкусно.')
```

Обратите внимание: интерпретатор ждет, что пользователь что-то введёт с клавиатуры ровно столько раз, сколько команд `input()` встречается в программе. Каждый `input()` завершается нажатием `Enter` на клавиатуре.

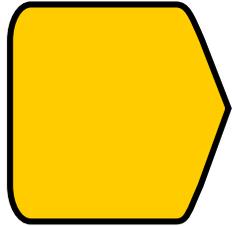
# Пример

```
print('Как тебя зовут?')
name = input()
print('Привет,', name)
print('Какая твоя любимая еда?')
meal = input()
print('Да.', meal, '- это вкусно.')
```

```
hello = 'Здравствуйте.'
hello2 = hello
print(hello2)
```

Но могут ли в пределах одной программы под одним именем быть разные значения? Да!

# Переменные



Итак, если вы хотите, чтобы у вас была **переменная с каким-то именем и каким-то значением**, нужно написать на отдельной строчке:

**<имя переменной> = <значение переменной> .**

Как только эта команда выполнится, в программе появится указанная переменная с таким значением.

**Помните:** команды выполняются последовательно, в том же порядке, в котором они написаны.

**Обратите внимание:** для того чтобы вывести кавычки внутри текстовой строки, необходимо вкладывать кавычки другого типа.

Взлом программы



# Задача «Загадка»

```
import random

months = ["январь", "февраль", "март", "апрель",
          "май", "июнь", "июль", "август", "сентябрь",
          "октябрь", "ноябрь", "декабрь"]

secret = random.choice(months)
# ВЫШЕ НЕПОНЯТНЫЙ КОД!
# сейчас в переменной secret лежит загаданный месяц
### В следующую строку нужно записать решение!

### В предыдущую строку нужно записать решение!
print("Какой месяц я загадал?")
solution = input()
# НИЖЕ НЕПОНЯТНЫЙ КОД!
# тут программа проверяет, совпал ли ответ с загаданным словом
if secret == solution:
    print("Точно!")
elif solution not in months:
    print("Это слово вообще не месяц!")
else:
    print("Не угадал!")
```

Трассировка

# Задача

Предположим, что у нас есть программа, которая входит в интерфейс сайта «Госуслуги» и служит для смены имени. Как будет работать эта программа, что она выведет при каком-либо пользовательском вводе?

```
print('Введите фамилию:')
surname = input()

print('Введите имя:')
name = input()

print(name, surname)

print('Введите новое имя:')
new_name = input()

print(name, surname)
print(new_name, name)
name = new_name
print(new_name, name)
print(name, surname)
```

# Задача

Предположим, что у нас есть программа, которая входит в интерфейс сайта «Госуслуги» и служит для смены имени. Как будет работать эта программа, что она выведет при каком-либо пользовательском вводе?

```
print('Введите фамилию:') # Введите фамилию
surname = input() # ожидание ввода -> surname = 'Пупкин'
print('Введите имя:') # Введите имя
name = input() # ожидание ввода -> name = 'Василий'
print(name, surname) # Василий Пупкин
print('Введите новое имя:') # Введите новое имя
new_name = input() # ожидание ввода -> new_name = 'Васисуалий'
print(name, surname) # Василий Пупкин
print(new_name, surname) # Васисуалий Пупкин
name = new_name # присваивание значения name = 'Васисуалий'
print(new_name, name) # Васисуалий Васисуалий
print(name, surname) # Васисуалий Пупкин
```

# Задача

Немного изменим программу. Как будет работать эта программа, что она выведет при каком-либо пользовательском вводе?

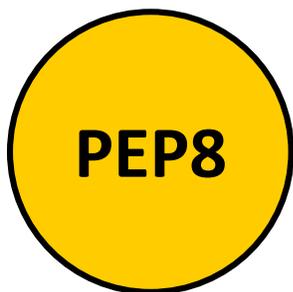
```
print('Введите фамилию:') #
surname = input() #
print('Введите имя:') #
name = input() #
print(name, surname) #
print('Введите новое имя:') #
new_name = input() #
old_name = name #
name = new_name #
print(new_name, old_name) #
print(name, surname) #
```

# Комментарии

Если вы начнёте строку со знака решётки #, то интерпретатор Python будет игнорировать всю эту строку. Программа будет выполняться так, как будто строки нет. Такая строка называется **комментарием**.

Комментарии нужны в двух случаях:

1. Когда нужно добавить в программу какую-то пометку для человека, который будет читать эту программу (например, см. третью строку ***guessing\_game***).
2. Когда нужно убрать какую-то строку кода, но удалять её не хочется (например, потом её, возможно, понадобится вернуть). Это называется «закомментировать» строку.



"Встрочные" комментарии находятся в той же строке, что и инструкция. Они должны **отделяться по крайней мере двумя пробелами** от инструкции и начинаться с символа # и одного пробела.

Комментарии в строке с кодом не нужны и только отвлекают от чтения, если они объясняют очевидное.

**Правильно:** `x = x + 1` *# компенсация границы*

**Неправильно:** `x = x + 1` *# увеличение на единицу*

**Яндекс**