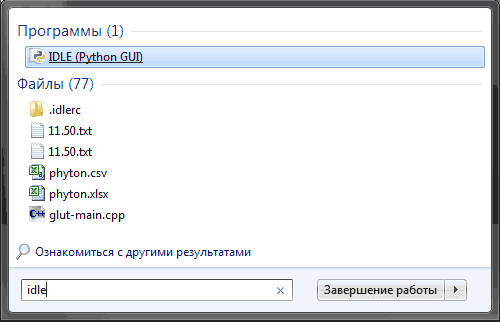
**Лабораторна робота №2. Перша програма. Среда разработки IDLE. Синтаксис. Чому моя програма не працює?**

**1. Первая программа. Среда разработки IDLE**

Сегодня мы напишем свою первую программу в среде разработки IDLE.

После загрузки и установки python открываем IDLE (среда разработки на языке Python, поставляемая вместе с дистрибутивом).

Здесь и далее буду приводить примеры под ОС Windows, так как именно она у меня сейчас под рукой.

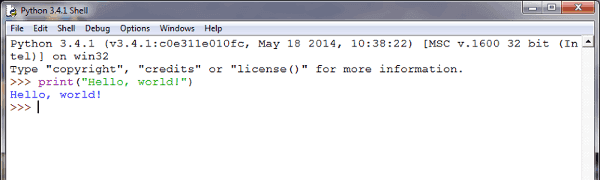


Запускаем IDLE (изначально запускается в интерактивном режиме), после чего уже можно начинать писать первую программу. Традиционно, первой программой у нас будет "hello world".

Чтобы написать "hello world" на python, достаточно всего одной строки:

print("Hello world!")

Вводим этот код в IDLE и нажимаем Enter. Результат виден на картинке:



Поздравляю! Вы написали свою первую программу на python!

С интерактивным режимом мы немного познакомились, можете с ним ещё поиграться, например, написать

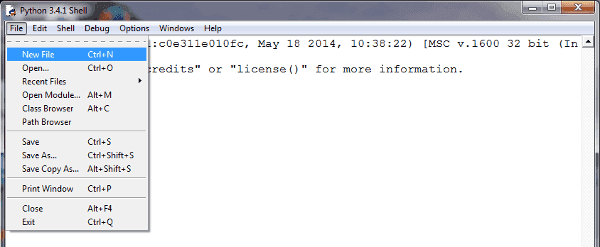
print(3 + 4)

print(3 \* 5)

print(3 \*\* 2)

Но, всё-таки, интерактивный режим не будет являться основным. В основном, вы будете сохранять программный код в файл и запускать уже файл.

Для того, чтобы создать новое окно, в интерактивном режиме IDLE выберите File → New File (или нажмите Ctrl + N).



В открывшемся окне введите следующий код:

name = input("Как Вас зовут? ")

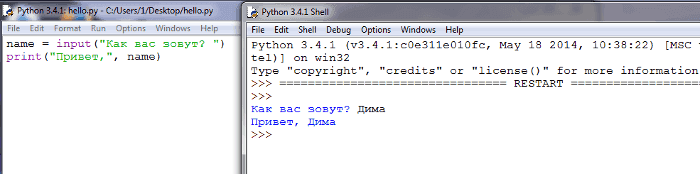
print("Привет,", name)

Первая строка печатает вопрос ("Как Вас зовут? "), ожидает, пока вы не напечатаете что-нибудь и не нажмёте Enter и сохраняет введённое значение в переменной name.

Во второй строке мы используем функцию print для вывода текста на экран, в данном случае для вывода "Привет, " и того, что хранится в переменной "name".

Теперь нажмём F5 (или выберем в меню IDLE Run → Run Module) и убедимся, что то, что мы написали, работает. Перед запуском IDLE предложит нам сохранить файл. Сохраним туда, куда вам будет удобно, после чего программа запустится.

Вы должны увидеть что-то наподобие этого (на скриншоте слева - файл с написанной вами программой, справа - результат её работы):



Поздравляю! Вы научились писать простейшие программы, а также познакомились со средой разработки IDLE.

**2. Синтаксис языка Python**

Синтаксис языка Python, как и сам язык, очень прост.

***Синтаксис***

* Конец строки является концом инструкции (точка с запятой не требуется).
* Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков. И про читаемость кода не забывайте. Отступ в 1 пробел, к примеру, не лучшее решение. Используйте 4 пробела (или знак табуляции, на худой конец).
* Вложенные инструкции в Python записываются в соответствии с одним и тем же шаблоном, когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

Основная инструкция:

Вложенный блок инструкций

***Несколько специальных случаев***

* Иногда возможно записать несколько инструкций в одной строке, разделяя их точкой с запятой:

a = 1; b = 2; print(a, b)

Но не делайте это слишком часто! Помните об удобочитаемости. А лучше вообще так не делайте.

* Допустимо записывать одну инструкцию в нескольких строках. Достаточно ее заключить в пару круглых, квадратных или фигурных скобок:

**if** (a == 1 **and** b == 2 **and**

c == 3 **and** d == 4): *# Не забываем про двоеточие*

print('spam' \* 3)

Тело составной инструкции может располагаться в той же строке, что и тело основной, если тело составной инструкции не содержит составных инструкций. Ну я думаю, вы поняли :). Давайте лучше пример приведу:

**if** x > y: print(x)

Полное понимание синтаксиса, конечно, приходит с опытом.

**3. Программа не работает. Что делать?**

Моя программа не работает! Что делать? В данной статье я постараюсь собрать наиболее частые ошибки начинающих программировать на python 3, а также расскажу, как их исправлять.

**Проблема**: Моя программа не запускается. На доли секунды появляется чёрное окошко, а затем исчезает.

**Причина**: после окончания выполнения программы (после выполнения всего кода или при возникновении исключения программа закрывается. И если вы её вызвали двойным кликом по иконке (а вы, скорее всего, вызвали её именно так), то она закроется вместе с окошком, в котором находится вывод программы.

**Решение**: запускать программу через IDLE или через консоль.

**Проблема**: Не работает функция input. Пишет SyntaxError.

**Пример кода**:

>>>

**>>>** a = input()

hello world

File "<string>", line 1

hello world

^

SyntaxError: unexpected EOF while parsing

**Причина**: Вы запустили Python 2.

**Решение**: Установить Python 3.

**Проблема**: TypeError: Can't convert 'int' object to str implicitly.

**Пример кода**:

>>> a = input() + 5

8

Traceback (most recent call last):

File "<stdin>", line 1, **in** <module>

TypeError: Can't convert 'int' object to str implicitly

**Причина**: Нельзя складывать строку с числом.

**Решение**: Привести строку к числу с помощью функции int(). Кстати, заметьте, что функция input() всегда возвращает строку!

>>>

**>>>** a = int(input()) + 5

8

**>>>** a

13

**Проблема**: SyntaxError: invalid syntax.

**Пример кода**:

a = 5

**if** a == 5

print('Ура!')

**Ошибка**:

File "a.py", line 3

**if** a == 5

^

SyntaxError: invalid syntax

**Причина**: Забыто двоеточие.

**Решение**:

a = 5

**if** a == 5:

print('Ура!')

**Проблема**: SyntaxError: invalid syntax.

**Пример кода**:

a = 5

**if** a = 5:

print('Ура!')

**Ошибка**:

File "a.py", line 3

**if** a = 5

^

SyntaxError: invalid syntax

**Причина**: Забыто равно.

**Решение**:

a = 5

**if** a == 5:

print('Ура!')

**Проблема**: NameError: name 'a' is not defined.

**Пример кода**:

print(a)

**Причина**: Переменная "a" не существует. Возможно, вы опечатались в названии или забыли инициализировать её.

**Решение**: Исправить опечатку.

a = 10

print(a)

**Проблема**: IndentationError: expected an indented block.

**Пример кода**:

a = 10

**if** a > 0:

print(a)

**Причина**: Нужен отступ.

**Решение**:

a = 10

**if** a > 0:

print(a)

**Проблема**: TabError: inconsistent use of tabs and spaces in indentation.

**Пример кода**:

a = 10

**if** a > 0:

print(a)

print('Ура!')

**Ошибка**:

File "a.py", line 5

print('Ура!')

^

TabError: inconsistent use of tabs **and** spaces **in** indentation

**Причина**: Смешение пробелов и табуляции в отступах.

**Решение**: Исправить отступы.

a = 10

**if** a > 0:

print(a)

print('Ура!')

**Проблема**: UnboundLocalError: local variable 'a' referenced before assignment.

**Пример кода**:

**def** f():

a += 1

print(a)

a = 10

f()

**Ошибка**:

Traceback (most recent call last):

File "a.py", line 7, **in** <module>

f()

File "a.py", line 3, **in** f

a += 1

UnboundLocalError: local variable 'a' referenced before assignment

**Причина**: Попытка обратиться к локальной переменной, которая ещё не создана.

**Решение**:

**def** f():

**global** a

a += 1

print(a)

a = 10

f()

**Проблема**: Программа выполнилась, но в файл ничего не записалось / записалось не всё.

**Пример кода**:

>>>

**>>>** f = open('output.txt', 'w', encoding='utf-8')

**>>>** f.write('bla')

3

>>>

**Причина**: Не закрыт файл, часть данных могла остаться в буфере.

**Решение**:

>>>

**>>>** f = open('output.txt', 'w', encoding='utf-8')

**>>>** f.write('bla')

3

**>>>** f.close()

>>>

[](https://pythonworld.ru/uploads/python-error.png)Также вам может быть полезно это описание: