

Лабораторная работа № 1

Организация работы пользователя в системе программирования Python. Организация ввода-вывода данных

Цель работы: получение навыков в организации ввода/вывода данных и порядка работы в системе программирования Python.

Оборудование: ПЭВМ.

Общие положения

Python - интерпретируемый, объектно-ориентированный высокоуровневый язык программирования с динамической семантикой. Это высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Установка Python на компьютер зависит от используемой операционной системы. Нужно понимать, что следует различать базовую установку, включающую интерпретатор, среду разработки IDLE, а также стандартную библиотеку, и установку дополнительных модулей, которых для Python написано очень много.

Способы установки Python:

1) базовая установка, включающая интерпретатор, среду разработки IDLE, а также стандартную библиотеку

Python 3 по ссылке <https://www.python.org/downloads/>

2) установка специализированных сборок, при которой вы получаете готовую и настроенную среду программирования с множеством установленных модулей помимо стандартной библиотеки.

Wing IDE 101 по ссылке <http://wingware.com/downloads/wingide-101>

В среде Wing установить следующие настройки:

Зайдите в меню Edit → Preferences. Перейдите к категории Files. Для опции Default Encoding выберите значение Unicode (UTF-8) utf-8.

Зайдите в меню Edit → Preferences. Перейдите к категории Debugger → I/O. Для опции Debug I/O Encoding выберите значение Unicode (UTF-8) utf-8.

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

Программа на Python, с точки зрения интерпретатора, состоит из логических строк. Одна логическая строка, как правило, располагается в одной физической, но длинные логические строки можно явно (с помощью обратной косой черты) или неявно (внутри скобок) разбить на несколько физических:

```
print (a, " - очень длинная строка, которая не помещается в", \
80, "знакоместах")
```

Конец строки является концом инструкции (точка с запятой не требуется).

В Python отступ очень важен. Вложенные инструкции объединяются в блоки по величине отступов.

Когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, Python использует отступ для указания блока кода. Используют **отступ - 4 пробела**.

Основная инструкция:

Вложенный блок инструкций

Python предоставляет возможность комментирования документации внутри кода. **Комментарии** следует начинать с символа #, а Python отобразит остальную часть строки в виде комментария:

Это комментарий

Основными типами данных в Python являются:

- **int** – целочисленные значения – положительные и отрицательные целые числа, а также 0 (например, 4, 687, -45, 0);
- **float** – вещественные (дробные) значения (например, 1.45, -3.789654, 0.00453). Для разделения целой и дробной частей здесь используется точка, а не запятая;
- **bool** – логические значения — значения могут принимать одно из двух значений: True (истина) или False (ложь);
- **str** – символьная строка или единичный символ - набор символов, заключенных в кавычки (например, "ball", "Whatisyourname?", 'd', '6589'). Кавычки в Python могут быть одинарными или двойными.

В программе на языке Python связь между данными и переменными устанавливается с помощью **оператора присваивания**, который обозначается знаком (=).

<имя переменной> = <значение переменной>

Выполняется оператор стандартным образом: сначала вычисляется выражения справа от знака равенства, а затем полученное значение записывается в переменную, указанную слева от знака равенства.

Ввод данных осуществляется при помощи оператора **input()**:

a = input()

В скобках функции можно указать сообщение-комментарий к вводимым данным:

a = input ("Введите количество: ")

Вывод данных осуществляется при помощи оператора **print()**:

a = 1

b = 2

print(a)

print(a + b)

print('сумма = ', a + b)

Внутри круглых скобок через запятую пишется то, что хотим вывести.

По умолчанию функция **print()** принимает несколько аргументов, выводит их через пробел, после чего ставит перевод строки. Это поведение можно изменить, используя именованные параметры **sep** (разделитель) и **end** (окончание).

Порядок выполнения работы

Задание 1. Напишите программу, которая выводит на первой строчке «Привет, Python!», а на второй – «Приятно познакомиться.»

Задание 2. Напишите программу, которая считывает с клавиатуры последовательно три строки: название фильма, название кинотеатра и время, после чего выводит на экран «Билет на " [название фильма] " в " [название кинотеатра] " на [время] забронирован.» .

Задание 3. Ввести с клавиатуры три целых числа, найти их сумму, произведение и среднее арифметическое.

Задание 4. Получить случайное трехзначное число и вывести через запятую его отдельные цифры.

Задание 5. Ввести три числа: цену пирожка (два числа: рубли, потом – копейки) и количество пирожков. Найти сумму, которую нужно заплатить (рубли и копейки)

Пример:

Стоимость пирожка:

12

50

Сколько пирожков:

5

К оплате: 62 руб. 50 коп.

Содержание отчета

1. Постановка задачи.
2. Текст программы.
3. Результаты выполнения программы.