**Лабораторная работа №1**

**Тема Введение в Python**

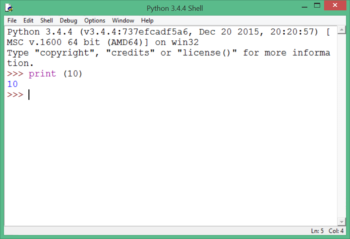
**Содержание:**

* Среда разработки
* Сравнение кода двух языков
* Оператор присваивания
* Ввод и вывод данных
* Арифметические действия и математические функции
* Преобразование типов
* Python для системы счисления
* Условный оператор
  + Сложные условия

**Среда разработки**

Для обучения используется IDLE python 3.X GUI (возможна и другая среда разработки).

Основным окном IDLE является окно интерпретатора, позволяющее программировать в интерактивном режиме, т.е. после ввода команды, она сразу же исполняется после нажатия [Enter]. Каждая строка начинается с символов >>>, результат выполнения отображается сразу под строкой оператора:



*IDLE python 3.4*

Для полноценной работы и сохранения результатов в файл следует выбрать в *меню File* пункт *NewFile*. После чего в открывшемся окне можно набирать код программы.

**Сравнение кода двух языков**

Для начала сравним фрагменты кода одной и той же программы, написанной на Python и на C++. Результат показывает, насколько меньше код, написанный на Python:  
    
**На PYTHON**

|  |  |
| --- | --- |
| 1 | **print**("Hello, World!") |

**На С++**

|  |  |
| --- | --- |
| 1  2  3  4  5 | #include  void main()  {  cout<<"Hello, World!"<<endl;  } |

**Оператор присваивания**

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | A =3.14  **print**(type(A))*# float*  A ='Hello'  **print**(type(A))*# str*  a = b = c =0  a +=1*# a = a + 1*  с=5//2*#int*  d =5/2*# float*  b = c\*\*2*# b = c^2 (степень)*  a, b = b, a *# обмензначениями a=b, b=a* |

**Ввод и вывод данных**

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

|  |  |
| --- | --- |
| 1  2  3  4  5 | a =1  b =2  **print**(a)  **print**(a + b)  **print**('сумма = ', a + b) |

* Существует возможность записи инструкций в одну строку, разделяя их через ;. Однако не следует часто использовать такой способ, это снижает удобочитаемость:

|  |  |
| --- | --- |
| 1  2  3 | a =1; b =2;**print**(a)  **print**(a + b)  **print**('сумма = ', a + b) |

* Для функции **print**может задаваться так называемый сепаратор — разделитель между элементами вывода:

|  |  |
| --- | --- |
| 1  2  3 | x=2  y=5  **print**( x,"+", y,"=",x+y,sep=" ") |

Результат отобразится с пробелами между элементами: 2 + 5 = 7

* Для форматированного вывода используется format:

|  |  |
| --- | --- |
| 1  2 | x =11  **print**("{:4d}".format(x)) |

В результате выведется число 11, а перед ним два пробела, так как указано использовать для вывода четыре знакоместа.

* Или с несколькими аргументами:

|  |  |
| --- | --- |
| 1  2 | x =2  **print**("{:4d}{:4d}{:4d}".format(x,x+x, x\*x)) |

В итоге каждое из значений выводится из расчета 4 знакоместа.

* Ввод данных осуществляется при помощи оператора input:

|  |  |
| --- | --- |
| 1  2 | a =input()  **print**(a) |

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

|  |
| --- |
| a =input("Введите количество: ") |

* Функция **input**воспринимает входные данные, как поток символов. Поэтому, чтобы принять целочисленное значение, следует воспользоваться функцией **int()**:

|  |
| --- |
| a =int(input()) |

**Задание Python 1\_1:**  
Используя операторы **input** и **print**, выведите значения целых переменных **x=5**и **y=3** в следующих форматах:

 3+5=?

 3 | 5 | 8

 Z(5)=F(3)

 x=5; y=3;

 Ответ: (5;3)

**Арифметические действия и математические функции**

**x + y** Сложение

**x - y** Вычитание

**x \* y** Произведение

**x / y** Деление

**x // y** Целочисленное деление

**x % y** Остаток от целочисленного деления

**x\*\*y** Возведение в степень

**-x** Смена знака числа

**Рассмотрим стандартные математические функции:**

* **abs(x)** модуль числа;
* **round(x)** – в Python округление числа x к ближайшему целому;
* **sqrt(x)** – квадратный корень числа;
* **exp(x)** – экспонента x;
* **log(x)** – натуральный логарифм x.

Перечисленные функции можно использовать в программе, только подключив специальный модуль:

|  |
| --- |
| **import**math  *# далее используем какую-либо функцию:*  **print**(math.sqrt(64))*# 8.0* |

**Преобразование типов**

Перечислим основные типы данных в Python, которые понадобятся на ближайших занятиях:

**int** – целочисленные значения;  
**float** – вещественные (дробные) значения;  
**bool** – логические значения — истина (True) или ложь (False);  
**str** – символьная строка или единичный символ.

Рассмотрим примеры того, как используется преобразование типов в Питон:

преобразование в символьную строку:

|  |  |
| --- | --- |
| 1  2  3 | a =1.7  a=str(a)  **print**(a)*# '1.7'* |

преобразование в целое:

|  |  |
| --- | --- |
| 1  2  3 | x =1.7  x=int(x)  **print**(x)*# 1* |

преобразование в вещественное:

|  |  |
| --- | --- |
| 1  2  3 | y=1  y=float(y)  **print**(y)*# 1.0* |

**Python для системы счисления**

* **bin(y)** — целое число преобразовывается в двоичную строку.
* **hex(y)** — целое число преобразовывается в шестнадцатеричную строку.
* **oct(y)** — целое число преобразовывается в восьмеричную строку.

Рассмотрим примеры работы с системами счисления:

|  |
| --- |
| bin(17)*# '0b10001'*  oct(17)*# '0o21'*  hex(17)*# '0x11'* |

**Задание Python 1\_2:** Даны две переменные. Запросить их значение. Выполнить основные арифметические действия с переменными, целочисленное деление, возведение в квадрат. Осуществить перевод в системы счисления. Вывести результат.

**Условный оператор**

В языке python синтаксис обладает интересной особенностью: дело в том, что в коде нет операторных скобок (begin..end или {...}); вместо них **отступы** указывают, какие операторы выполнять внутри той или иной конструкции.

**Зачем отступы и где их ставить?**

Стандартная запись условной инструкции Python ifelseвыглядит следующим образом:

|  |
| --- |
| **if** условие1:  оператор1  **elif** условие2:  оператор2  **else**:  оператор3 |
| **if** x >0:  **if** x <2:  **else**:  оператор |  |

 Следует обратить внимания, что знак сравнения в Питоне записывается, как два знака =:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **if** x <0:  блок1  **elif** x ==0: *# сравнение!*  блок2  **else**:  блок3 |

Другой пример использования условия:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **if** x <0:  **print**('мало')  **elif** x ==0:  **print**('средне')  **else**:  **print**('много') |

**СЛОЖНЫЕ УСЛОВИЯ**

* Использование двойных неравенств разрешено:

|  |  |
| --- | --- |
| 1  2  3  4 | **if**0< x<2:  **if**0< y<2:  **else**:  оператор |

* Пример использования условия c elif:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **if** x <0:  **print**('мало')  **elif** -0<= x <=0:  **print**('средне')  **else**:  **print**('много') |

* Кроме того, можно применять логический оператор AND (И):

|  |
| --- |
| **if**x >=30**and** x <=40:  ... |

**Задание Python 1\_3:** Запрашивается количество часов и заработок в час. Рассчитать зарплату. Если работник работал более *40* часов, — то зарплата умножается на коэффициент *1,5*. Оформить в формате дружелюбного интерфейса.

**Задание Python 1\_4:** Напишите программу, которая определяет, верно ли, что введённое число – четырёхзначное.