Основы программирования на языке Python

- Одним из самых популярных современных языков программирования является Python (произносится «па́йтон» или просто «питон»). Его разработал в 1991 году нидерландский программист Гвидо ван Россум. Этот язык непрерывно совершенствуется, сейчас используется версия Python 3 (Python 3.9)
- Свое имя – Пайтон (или Питон) – получил от названия телесериала, а не пресмыкающегося
- Язык Python применяется для обработки различных данных, математических вычислений, создания изображений, работы с базами данных, разработки веб-сайтов.

Особенности Python

- Простой
- Лёгкий в освоении
- Свободный и открытый
- Язык высокого уровня
- Портируемый
- Интерпретируемый
- Объектно-ориентированный
- Расширяемый
- Встраиваемый
- Обширные библиотеки

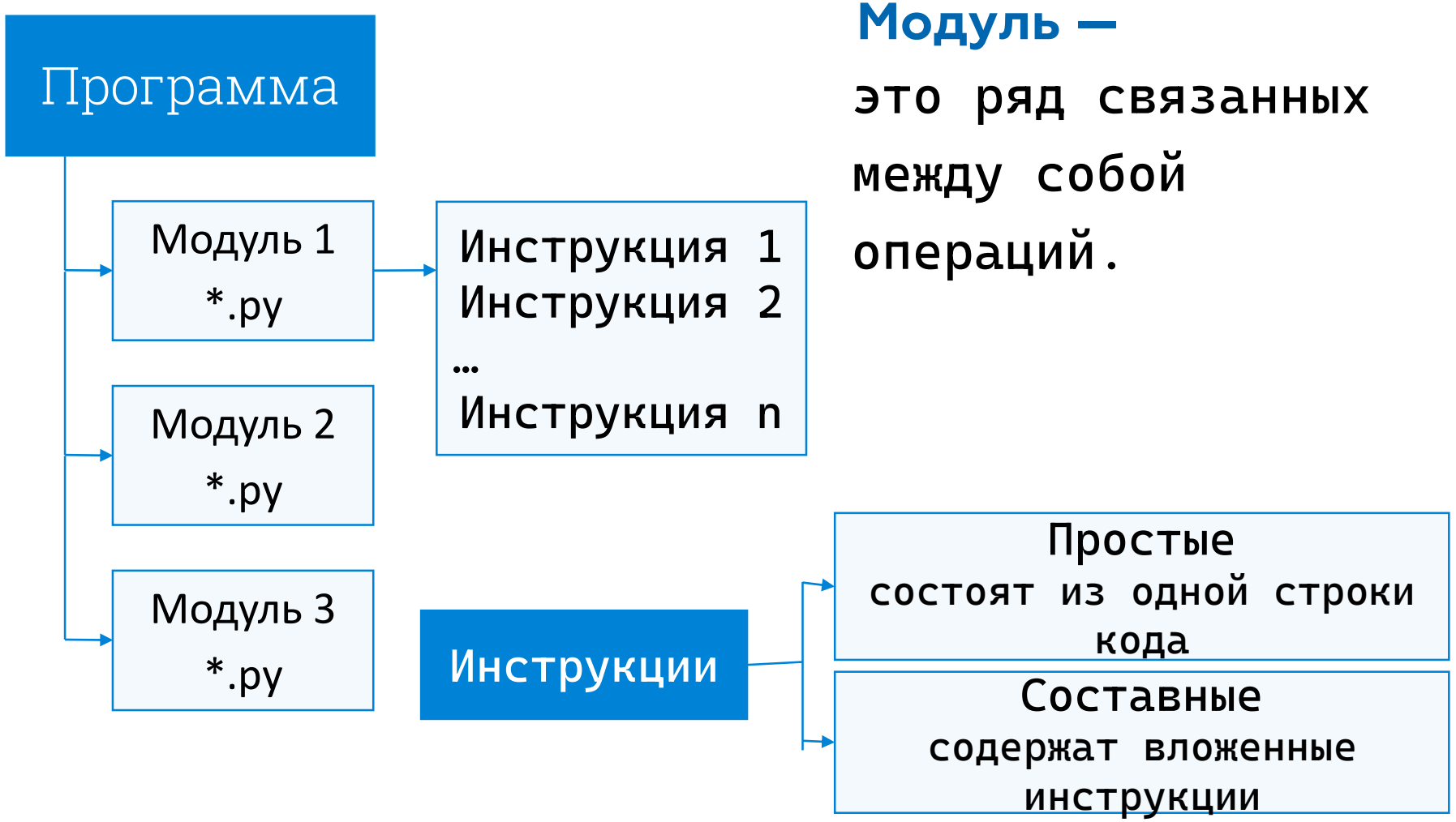
Python

Интерпретируемый высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью

Модульность

- принцип, согласно которому программа разделяется на отдельные именованные сущности, называемые модулями.
- Модульность часто является средством упрощения задачи проектирования программы и распределения процесса разработки между группами разработчиков.
- При разбиении программы на модули для каждого из них указывается реализуемая им функциональность, а также связи с другими модулями

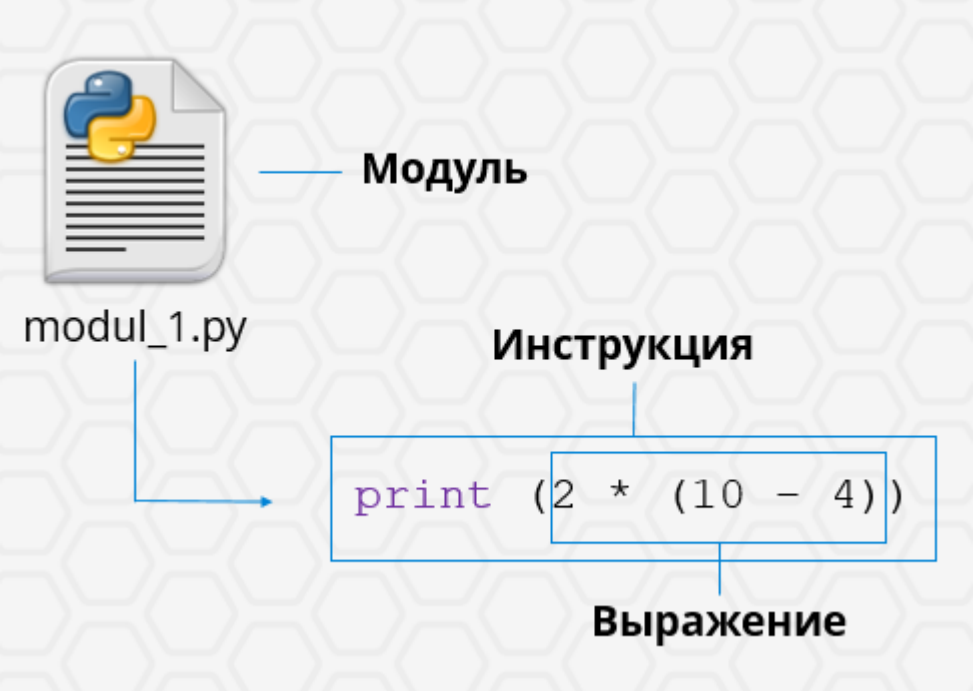
Структура программы на языке Python:



Модуль –

это ряд связанных между собой операций.

Структура программы на языке Python:



Инструкции — это указания компьютеру, определяющие, какие операции выполнит компьютер над данными.

Выражения в составе инструкций определяют, над какими именно данными будут выполнены действия, описанные в инструкции.

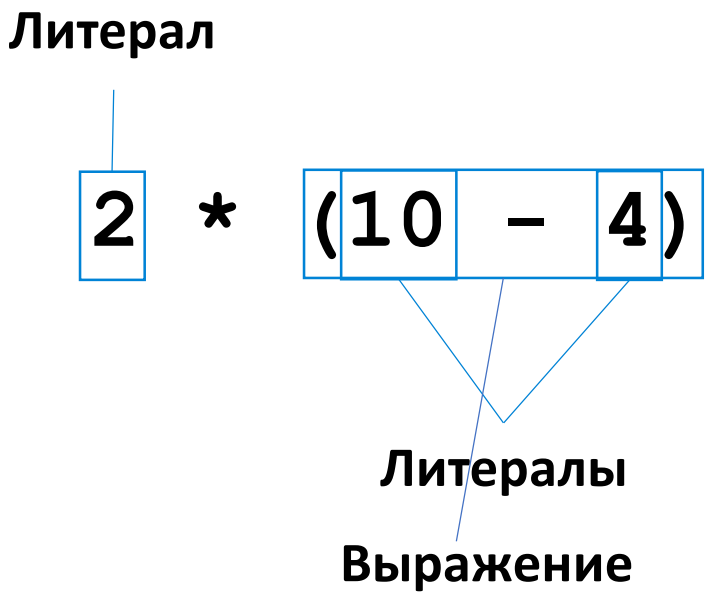
Структура программы на языке Python:

Операции - это любые действия над операндами.

Операнды -

это некоторые данные.

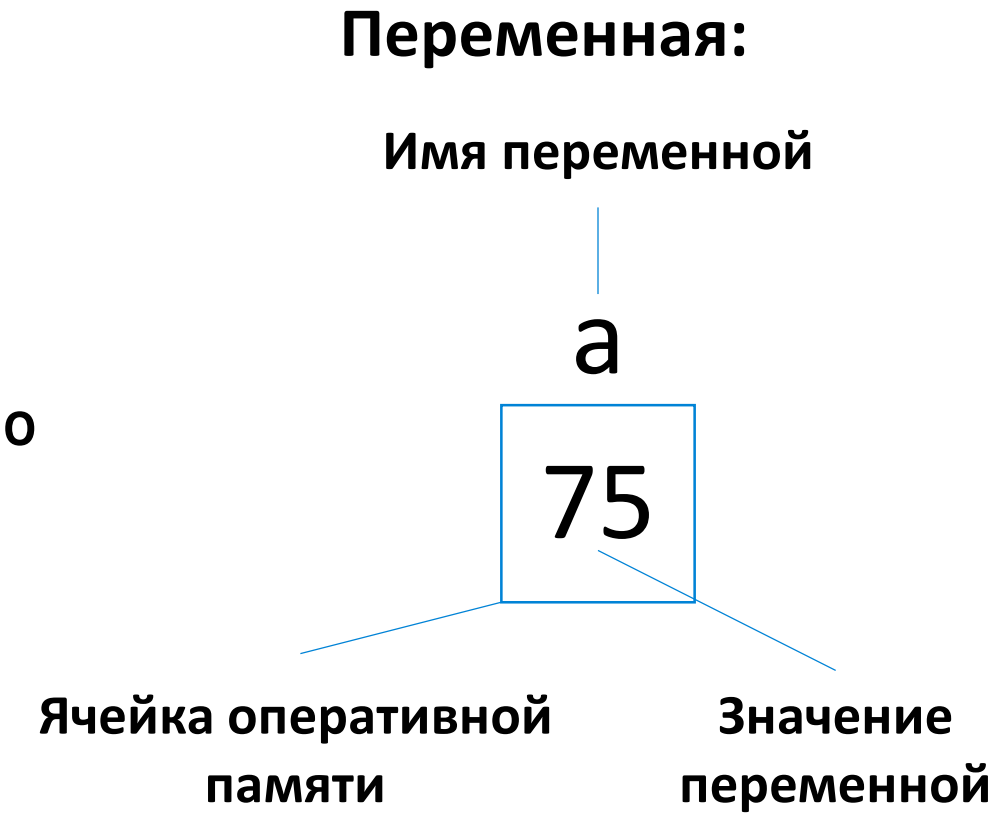
- ✓ литералы;
- ✓ выражения;
- ✓ переменные.



Приоритет выполнения операций - соответствует принятому в математике.

Структура программы на языке Python:

Переменная – это именованная область оперативной памяти, в которой хранятся некоторые данные определённого типа.



- Имя переменной может состоять только из цифр, букв и символов подчеркивания
- Имя переменной не может начинаться с цифр
- Имя должно описывать суть , т.е. нужно давать имена, говорящие о назначении данных, на которые они ссылаются
- Имя переменной не должно совпадать с командами языка (зарезервированными ключевыми словами)
- Имя переменной принято начинать со строчной буквы
- Не следует создавать имена длиннее 15 символов
- Чтобы узнать значение, на которое ссылается переменная, находясь в режиме интерпретатора, достаточно ее вызвать (написать имя и нажать Enter).

В пайтоне применяется два типа наименования переменных: camel case и underscore notation

Camel case подразумевает, что каждое новое подслово в наименовании переменной начинается с большой буквы. Например:

```
1 userName = "Tom"
```

Underscore notation подразумевает, что подслова в наименовании переменной разделяются знаком подчеркивания. Например:

```
1 user_name = "Tom"
```

КОПИРОВАТЬ

```
>>> sun_to_earth = 149597970
>>> sun_to_earth = sun_to_earth + 1
>>> print(sun_to_earth)
149597971
```

Инструкция присваивания

Инструкция присваивания

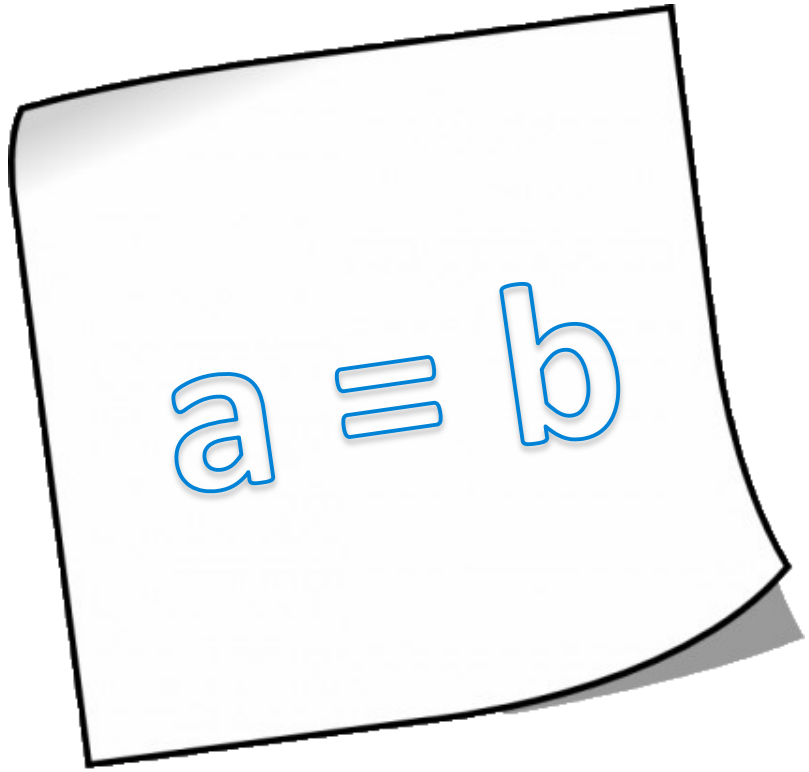
определяет данные, соответствующие переменной.

Запись инструкции:

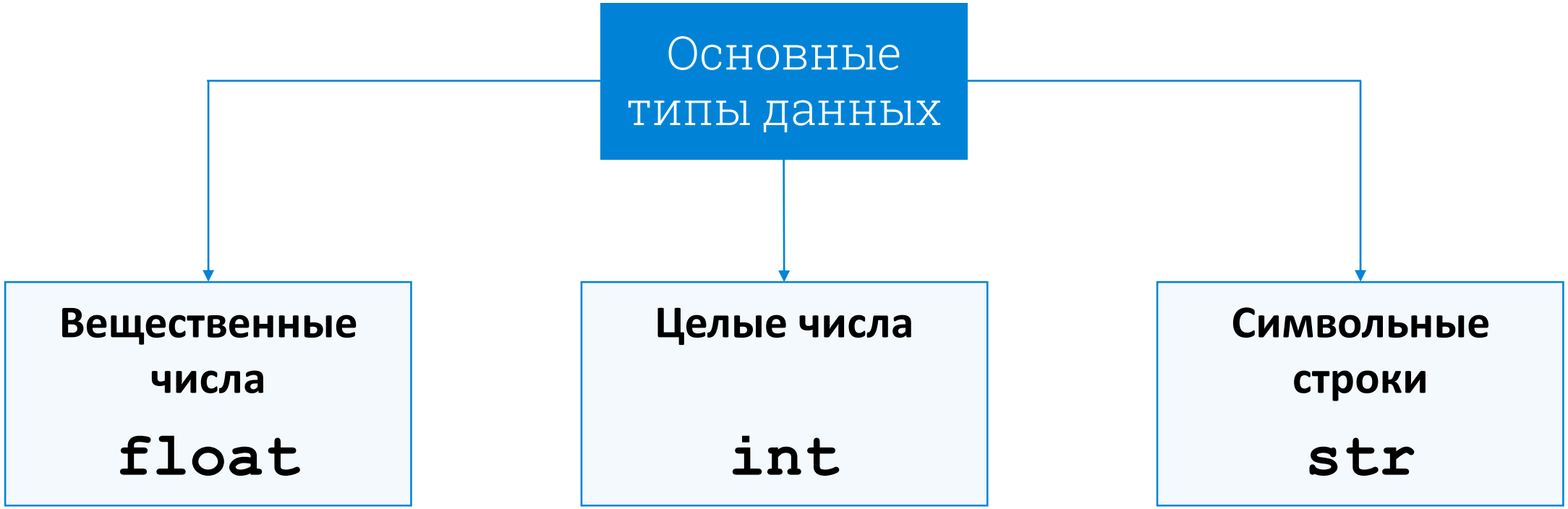
<имя переменной> = <значение>

Пример:

a = 4



Типы данных в языке Python



Синтаксис

- Не содержит операторных скобок (begin..end в pascal или {..}в Си), вместо этого блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием.
- Однострочные комментарии начинаются со знака фунта «#», многострочные – начинаются и заканчиваются тремя двойными кавычками «"""».

Особенности синтаксиса и терминологии

Строка текста программы это инструкция
Отступы это часть синтаксиса
Составная конструкция языка состоит из заголовка и блока
Заголовок это строка, завершающаяся двоеточием
Блок это последовательность строк, отступ которых на единицу
больше, чем у строки-заголовка

```
if apples > pears:
    eat_apples(apples - pears)
    print('We ate extra apples')
get_more_fruits()
```

Функция print()

Начиная с версии 3 в языке Python инструкция print заменена функцией print()
Большинство объектов имеют свойство "быть напечатанным", то есть быть преобразованным в текстовую строку
Функция print() воспринимает любое количество аргументов и "печатает" их последовательно в стандартный вывод разделяя пробелом и завершая символом новой строки
Именованные аргументы позволяют настроить поведение функции print()
аргумент sep позволяет заменить разделитель
аргумент end позволяет заменить конец строки
аргумент file позволяет направить вывод в файл
Пример замены разделителя:

```
print(1, 2, 3) # => 1 2 3
print(1, 2, 3, sep='->') # => 1->2->3
```

Комментарии

Символ комментария #. Сам символ и все символы справа от него и до конца строки игнорируются

Для комментирования большого фрагмента текста можно использовать строки в тройных кавычках

Строка это элементарное выражение, а выражение это инструкция, то есть синтаксически корректная конструкция

Примеры:

```
a = 1
a = a * 2 # здесь a увеличивается вдвое
a = a * 2 # здесь a увеличивается вдвое еще раз
a = a * 2 # и еще раз
"""
a = a * 2 # и еще раз
a = a * 2 # и еще раз
a = a * 2 # и еще раз
"""
# После всех вычислений переменная a примет значение 8
```

Пример синтаксически корректной программы
Одиночное выражение, записанное в строке программы является допустимой инструкцией

```
a = 3 + 2 # инструкция присваивания
3 + 2 # одиночное выражение записанное в строке
"A string" # элементарное выражение
3 # элементарное выражение
a # элементарное выражение
a == 12 # одиночное выражение записанное в строке
```

Вызов функции без использования ее результата также является одиночным выражением, записанным в строке текста программы

```
exit(1)
```

Отступы составлены из пробелов или горизонтальных табуляций.
Рекомендуемый единичный отступ 4 пробела.
В отступах нельзя смешивать пробелы и табуляции; в Питоне версии 2 это допустимо но не рекомендуется
За пределами отступов:

- пробел и табуляция эквивалентны
- пробел это разделитель, но он необходим только там, где нет другого разделителя
- там где есть один пробел можно добавить любое количество пробелов

Синтаксис

- Чтобы присвоить значение переменной используется знак «=», а для сравнения – «==». Для увеличения значения переменной, или добавления к строке используется оператор «+=», а для уменьшения – «-=». Все эти операции могут взаимодействовать с большинством типов, в том числе со строками.

Синтаксис

- На PYTHON

```
print("Hello, World!")
```

- На C++

```
#include <iostream.h>
void main()
 {
     cout << "Hello, World!" << endl;
 }
```

Преимущества Python

- Скорость выполнения программ написанных на Python очень высока. Это связанно с тем, что основные библиотеки Python написаны на C++ и выполнение задач занимает меньше времени, чем на других языках высокого уровня.
- В стандартных библиотеках Python вы можете найти средства для работы с электронной почтой, протоколами Интернета, FTP, HTTP, базами данных, и пр.

Преимущества Python

- Скрипты, написанные при помощи Python выполняются на большинстве современных ОС. Такая переносимость обеспечивает Python применение в самых различных областях.
- Python подходит для любых решений в области программирования, будь то офисные программы, вэб-приложения, GUI-приложения и т.д.

Преимущества Python

Интерактивный режим

- В основном интерпретатор выполняет команды построчно: пишешь строку, нажимаешь Enter, интерпретатор выполняет ее, наблюдаешь результат.

- Возможности языка позволяют использовать его как калькулятор, не зная команд программирования.

```
2 + 5
3 * (5 – 8)
2.4 + 3.0 / 2
```

Недостатки Python

- Python, как и другие интерпретируемые языки, имеет сравнительно невысокую скорость выполнения программ. Однако, в случае с Python этот недостаток компенсируется уменьшением времени разработки программы. В среднем, программа на Python в 2–4 раза компактнее, чем её аналог на C++ или Java