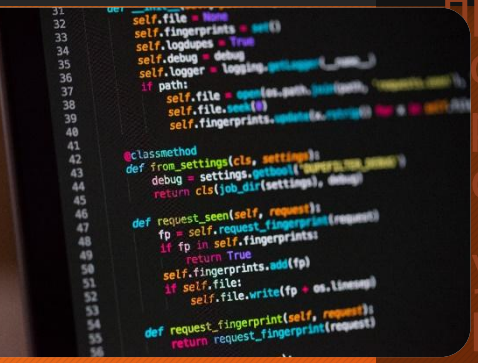
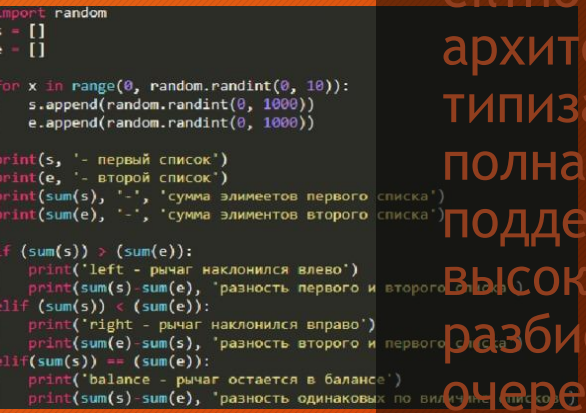
Язык Программирования

Python

Описание





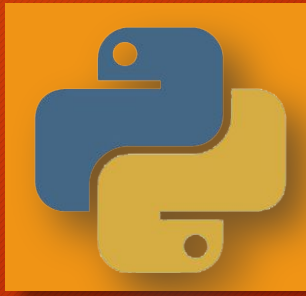
- Высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Преимущества Python

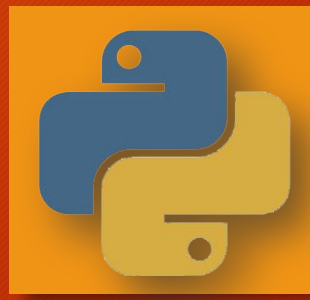
1. открытая разработка;
2. довольно прост в изучении, особенно на начальном этапе;
3. особенности синтаксиса стимулируют программиста писать хорошо читаемый код;
4. предоставляет средства быстрого прототипирования и динамической семантики;
5. имеет большое сообщество, позитивно настроенное по отношению к новичкам;
6. множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам;
7. механизмы модульности хорошо продуманы и могут быть легко использованы;
8. абсолютно всё в Python является объектами в смысле ООП, но при этом объектный подход не навязывается программисту.

Преимущества Python



Многие разработчики считают Python универсальным вариантом для программирования, чего только душе программиста будет угодно. Он достаточно простой и читабельный, а при всей своей простоте Python позволяет реализовать практически любую идею.

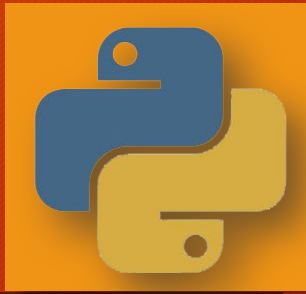
Преимущества Python



Пример кода, написанного на Python

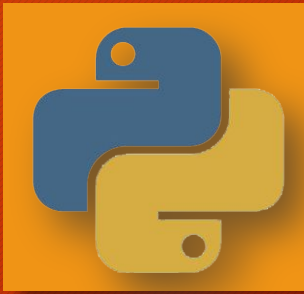
```python
from functools import partial, wraps
from inspect import getargspec

from functools import identity, _Q

class F(object):
    """Provide simple syntax for functions composition
    (through << and >> operators) and partial function
    application (through simple tuple syntax).
    Usage example:
    >>> func = F() << (_ + 10) << (_ + 5)
    >>> print(func(10))
    25
    >>> func = F() >> (filter, _ < 6) >> sum
    >>> print(func(range(10)))
    15
    """

    __slots__ = "f",

    def __init__(self, f = identity, *args, **kwargs):
        self.f = partial(f, *args, **kwargs) if any([args, kwargs]) else f

    @classmethod
    def __compose(cls, f, g):
        """Produces new class intance that will
        execute given functions one by one. Internal
        method that was added to avoid code duplication
        in other methods.
        """
        return cls(lambda *args, **kwargs: f(g(*args, **kwargs)))
```

Недостатки Python



- не слишком удачная поддержка многопоточности;
- на Python создано не так уж много качественных программных проектов по сравнению с другими универсальными языками программирования, например, с Java;
- отсутствие коммерческой поддержки средств разработки;
- изначальная ограниченность средств для работы с базами данных;
- бенчмарки показывают меньшую производительность Python по сравнению с основными Java VM, что создаёт этому языку репутацию медленного.

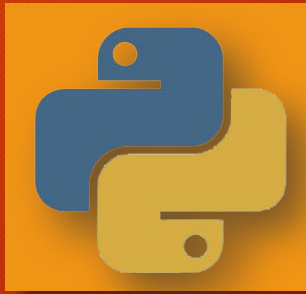
Недостатки Python



В сентябрьском рейтинге популярности языков программирования наблюдается рост популярности языка Python (7.653%), который переместился на третье место, вытеснив с него язык C++ (7.396%). Первое и второе места как и раньше удерживают языки Java (17.436%) и C (15.447%). По сравнению с сентябрём прошлого года.

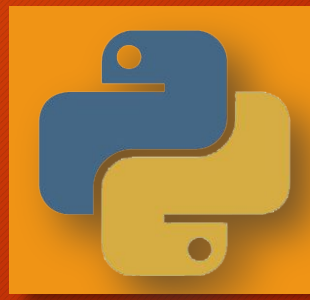
Сентябрь 2018	Сентябрь 2017				
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	^	Python	7.653%	+4.67%
4	3	v	C++	7.394%	+1.83%
5	8	^	Visual Basic .NET	5.308%	+3.33%
6	4	v	C#	3.295%	-1.48%
7	6	v	PHP	2.775%	+0.57%
8	7	v	JavaScript	2.131%	+0.11%
9	-	^^	SQL	2.062%	+2.06%
10	18	^^	Objective-C	1.509%	+0.00%

Структура данных



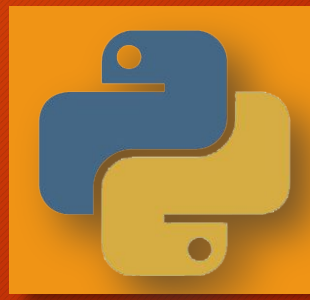
Python поддерживает динамическую типизацию, то есть тип переменной определяется только во время исполнения. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем». В Python имеются встроенные типы: булевый, строка, Unicode-строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие.
Из коллекций в Python
встроены: список, кортеж(*неизменяемый список*), словарь, множество и другие. Все значения являются объектами, в том числе функции, методы, модули, классы.

Структура данных

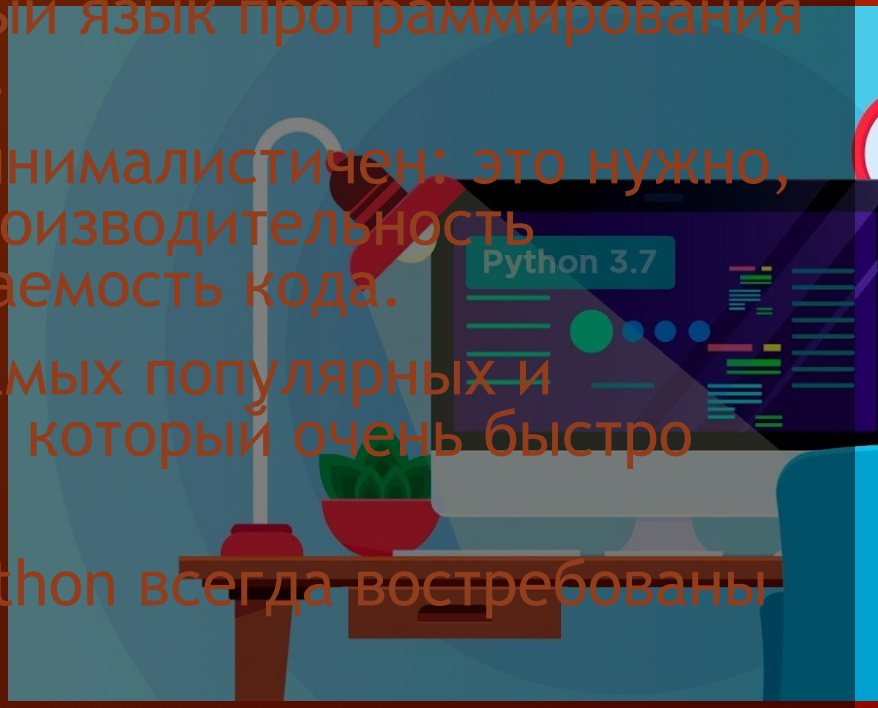


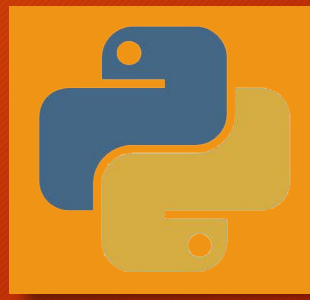
Все объекты делятся на ссылочные и атомарные. К атомарным относятся int, long (в версии 3 любое число int, так как в версии 3 нет ограничения на размер), complex и некоторые другие. При присваивании атомарных объектов копируется их значение, в то время как для ссылочных копируется только указатель на объект, таким образом, обе переменные после присваивания используют одно и то же значение. Ссылочные объекты бывают изменяемые и неизменяемые. Например, строки и кортежи являются неизменяемыми, а списки, словари и многие другие объекты — изменяемыми. Кортеж в Python является, по сути, неизменяемым списком. Во многих случаях кортежи работают быстрее списков, поэтому если не планируется изменять последовательность, то лучше использовать именно их.

Зачем учить Python?



- Это высокоуровневый язык программирования общего назначения.
- Синтаксис языка минималистичен: это нужно, чтобы увеличить производительность разработчика и читаемость кода.
- Python — один из самых популярных и адаптивных языков, который очень быстро развивается.
- Разработчики на Python всегда востребованы на рынке.





Презентация окончена

Спасибо за внимание