
QA
מע

אלכס גורבצ'וב
АЛЕКСЕЙ ГОРБАЧЁВ



PYTHON
PYTHON



Python



Введение

Python представляет популярный высокоуровневый язык программирования, который предназначен для создания приложений различных типов.

Это и веб-приложения, и игры, и настольные программы, и работа с базами данных. Довольно большое распространение Python получил в области машинного обучения и исследований искусственного интеллекта.

Впервые язык Python был анонсирован в 1991 году голландским разработчиком Гвидо Ван Россумом. С тех пор данный язык проделал большой путь развития. В 2000 году была издана версия 2.0, а в 2008 году - версия 3.0. Несмотря на вроде такие большие промежутки между версиями постоянно выходят подверсии.

Python



Введение

MAJOR COMPANIES
THAT USE



python

Google NETFLIX facebook. Instagram

amazon Quora # slack intel NASA

Dropbox ebay Spotify CapitalOne

Python



Введение

Почему тогда Python не №1?

Почему же при столь обширном спектре плюсов люди пишут еще и на других языках? Тут дело в нескольких причинах. Python удобный, но в сравнении с тем же языком C он куда более медленный. Вы этого не заметите в мелких проектах, но если решите писать крупный проект или вовсе свою операционную систему, то вам понадобится максимальная скорость выполнения кода, что Python не может обеспечить.

По схожим причинам Python не является лидером во всех доступных ему сферах. Для игр нужно быстроедействие, для программ также, поэтому там выбор в основном падает на другие языки, а вот веб сайты могут уже потесниться в плане скорости и поэтому +- 100 миллисекунд на выполнение особой роли там не сыграют.

Python



Введение

Основные особенности языка программирования Python:

- Скриптовый язык. Код программ определяется в виде скриптов.
- Поддержка различных аспектов программирования, в том числе объектно-ориентированного.
- Интерпретация программ. Для работы со скриптами необходим интерпретатор, который запускает и выполняет скрипт.

Выполнение программы на Python выглядит следующим образом:

Сначала мы пишем в текстовом редакторе скрипт с набором выражений на данном языке программирования -> Передаем этот скрипт на выполнение интерпретатору -> Интерпретатор транслирует код в промежуточный байткод, а затем виртуальная машина переводит полученный байткод в набор инструкций, которые выполняются операционной системой.

Python



Введение



Python



Введение

Вот лишь некоторые вещи, которые умеет делать Python:

- Работа с xml/html файлами
- Работа с http запросами
- GUI (графический интерфейс)
- Создание веб-сценариев
- Работа с FTP
- Работа с изображениями, аудио и видео файлами
- Робототехника
- Программирование математических и научных вычислений

Таким образом, Python подходит для решения львиной доли повседневных задач, будь то резервное копирование, чтение электронной почты, либо же какая-нибудь игрушка. Язык программирования Python практически ничем не ограничен, поэтому также может использоваться в крупных проектах. Python интенсивно применяется IT-гигантами, такими как, например, Google и Yandex.

Python



Установка

- Установим самую последнюю версию языка 3.11.4. Скачать можно [тут](#).
- После скачивания – устанавливаем как простую программу (пару кликов Next).
- Запустим Visual Code.
- Напишем код:

```
print("Hello world!")
```
- Сохраним как Class1.py

Python



Синтаксис

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

- 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 (лучше всего - знак табуляции).
- 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.

Python



Синтаксис

Несколько специальных случаев:

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

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

- 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:

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

Python



Синтаксис



Python



Типы переменных в Python.

Всего в Python есть 4 базовых типа переменных:

- Целочисленные (int)
- С плавающей точкой (float, complex)
- Строки (String)
- Логические (boolean)

Полная таблица всех типов:

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview

Python



Типы переменных в Python.

Типы переменных в языке Python **не объявляются очевидно**, тем не менее они присутствуют. Интерпретатор понимает что записывается в переменную и на основании этого добавляет тип к этой переменной.

Во время выполнения программы есть возможность перезаписывать переменные, а также менять их тип. Если вначале переменная была с типом `float`, то потом её можно преобразовать в другой тип, к примеру, в `String`.

В одной строке можно создать сразу несколько переменных:

```
first = sec = third = 1 # Всем трём переменным будет присвоено значение 1
```

```
first, sec, third = "Hi", 75, 23.1 # Поочередное присвоение значений
```

Python



Как оставлять комментарии в коде?

В Python для написания комментария используется #
Python не поддерживает комментарии на несколько строк

```
print(k); # выводит на консоль значение переменной k
```

Python



Основные операции с переменными в Python

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Python



Основные операции с переменными в Python

Operator	Example	Same As
=	x = 5	x = 5
=+	x += 3	x = x + 3
=-	x -= 3	x = x - 3
=*	x *= 3	x = x * 3
=/	x /= 3	x = x / 3
=%	x %= 3	x = x % 3
=//	x //= 3	x = x // 3
=**	x **= 3	x = x ** 3
=&	x &= 3	x = x & 3
=	x = 3	x = x 3
=^	x ^= 3	x = x ^ 3
=<<	x >>= 3	x = x >> 3
=>>	x <<= 3	x = x << 3

Python



Ввод данных в Python

В Python для получения данных от пользователя используется функция `input()`:

```
переменная = input("Сообщение")
```

В Python нету различия между типами получаемых данных. Тип по определению будет - String. Если мы хотим получить числовое значение, используем:

```
переменная = int(input("Сообщение")) # для целых
```

```
переменная = float(input("Сообщение")) # для дробных чисел
```

Если в строке вводится не одно число, а сразу несколько (например, четыре), придется воспользоваться функцией `map`:

```
a, b, c, d = map(int, input().split())
```

Java



Задание.

- Запросите у пользователя ввести имя и, отдельно год рождения
- Посчитайте возраст и выведите сообщение на консоль, в котором будет его имя и подсчитанный возраст

Python



Генерация случайных чисел в Python

В Python для получения случайных чисел используется функция `random.random()`:

```
random.random()
```

Для использования данной функции обязательно добавить библиотеку `random`:

```
import random
```

Удобнее всего использовать:

```
random.randrange(n,m) # n - нижняя граница, m - верхняя граница
```

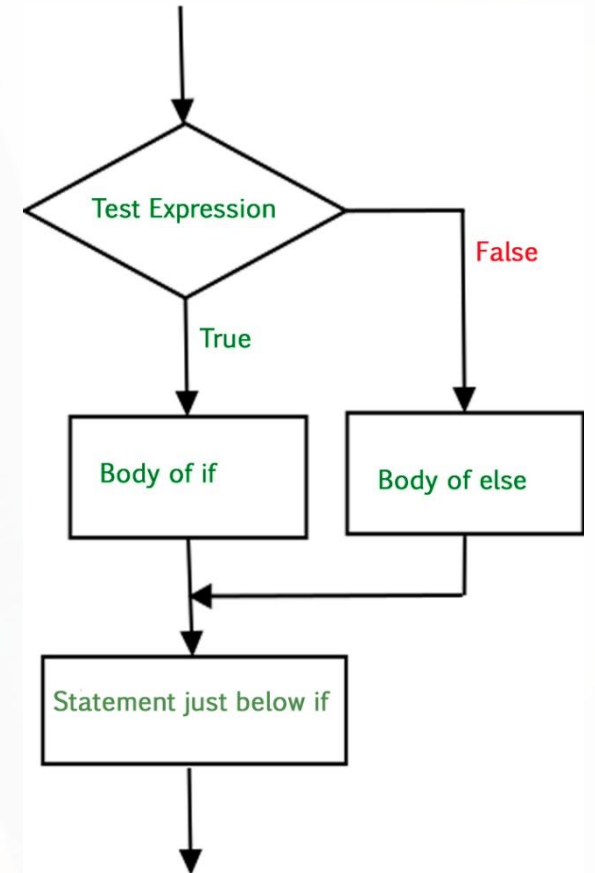
Python



Условные операторы в Python

Общая форма условного оператора if в Python такая:

```
if условие:  
    инструкции  
elif условие:  
    инструкции  
else:  
    инструкции
```



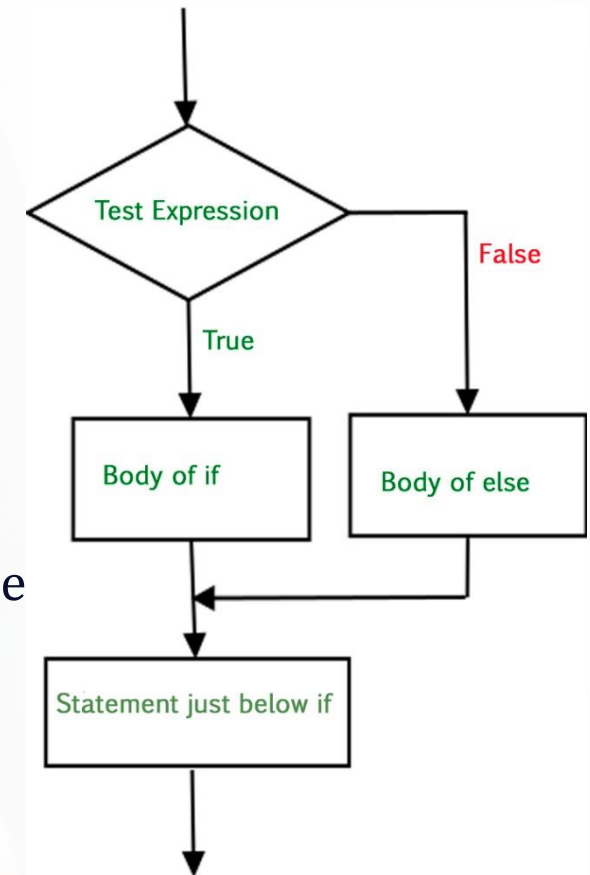
Python



Условные операторы в Python

Обратите внимание, что:

- Оператор всегда начинается со слова `if`, за которым идёт сразу условие.
- После условия ставится двоеточие (`:`).
- Для того, чтобы указать альтернативный вариант (если не выполняется, тогда...) используется слово `else`.
- Если условий несколько, то каждое из них будет записываться через оператор `elif`, после которого записывается альтернативное условие и ставится двоеточие (`:`).
- Последний вариант (если не то, не то и не то, то...) записывается через `else` без условия.



Задание.

- Необходимо написать программу, где бы пользователю предлагалось ввести любое число. Программа выполнит проверку на чётность.
Если чётное выведите: "Число " + number + " чётное."
Если нет то выведите: "Число " + number + " нечётное.»
- Необходимо написать программу определяющую, является ли данный год високосным. (Напомним, что год является високосным, если его номер кратен 4, но не кратен 100, а также если он кратен 400.)



Thanks for your time 😊