



Язык программирования PYTHON был создан примерно в 1991 году голландцем Гвидо ван Россумом.

Свое имя – Пайтон (или Питон) – получил от названия Телесериала.

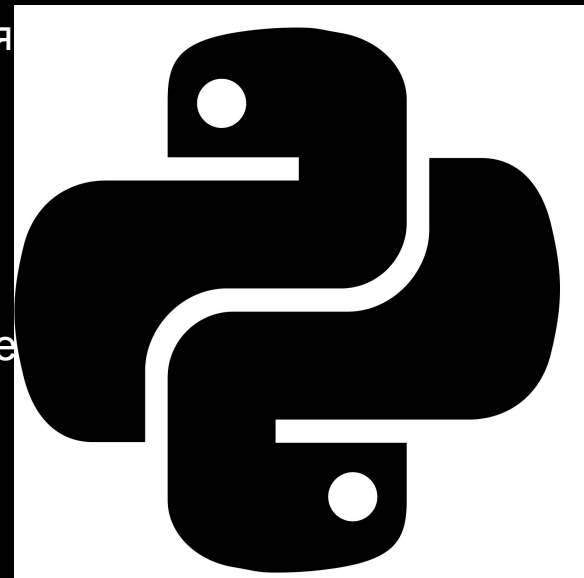
После того, как Россум разработал язык, он выложил его

в Интернет, где уже целое сообщество программистов присоединилось к его улучшению.

- Python – высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода

- Python и подавляющее большинство библиотек к нему бесплатны и поставляются в исходных кодах. Более того, в отличие от многих открытых систем, лицензия никак не ограничивает использование Python в коммерческих разработках

- Python имеет ясный синтаксис. В нем сведены к минимуму такие вспомогательные конструкции как скобки, слова-организаторы_блоков. Взамен программист обязан четко соблюдать отступы, которые и являются организаторами блоков. В результате код получается незагруженным лишними элементами и легко читаемым.





Python_interface.py

Project

- Python_interface.py
- External Libraries
- Scratches and Consoles

Python_interface.py

```
1 import serial          #import serial to enable sending and receiving data using the serial port
2 import tkinter         #tkinter is the GUI for python
3 data= serial.Serial('com4',9600)    #define the com of the arduino and the baud rate which is set to 9600
4 def turn_led_on():     #This is to turn the LED on, we send one over the serial port and in the arduino code we turn the
5     data.write(b'1')
6 def turn_led_off():    #This is to turn the LED off, we send one over the serial port and in the arduino code we turn the
7     data.write(b'0')
8 def blink():
9     data.write(b'2')
10 led_ctrl_win=tkinter.Tk()
11 Button=tkinter.Button
12 text=tkinter.Text
13 on=Button(led_ctrl_win, text="ON ", command=turn_led_on, height=6, width=10)
14 off=Button(led_ctrl_win, text="OFF", command=turn_led_off,height=6, width=10)
15 blinking=Button(led_ctrl_win, text="Blink", command=blink,height=6, width=20)
16 #T= text(led_ctrl_win, height=2, width=30)
17 #T.insert(data.read())
18 on.pack(side=tkinter.LEFT)
19 off.pack(side=tkinter.RIGHT)
20 blinking.pack(side=tkinter.LEFT)
21 led_ctrl_win.mainloop()
```

Преимущества Python

- Скорость выполнения программ написанных на Python очень высока. Это связано с тем, что основные библиотеки Python написаны на C++ и выполнение задач занимает меньше времени, чем на других языках высокого уровня.
- В связи с этим вы можете писать свои собственные модули для Python на C или C++
- В стандартных библиотеках Python вы можете найти средства для работы с электронной почтой, протоколами Интернета, FTP, HTTP, базами данных, и пр.
- 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, веб-приложения, GUI-приложения и т.д.

Недостатки PYTHON

- Python медлительный. Он способен на многое, но это многое он делает медленнее конкурентов
- Python не особо адекватно распоряжается памятью и часто «съедает» больше ресурсов, чем для решения тех же задач использовали бы JavaScript или C
- Python строго привязан к системным библиотеками. Отсюда возникают сложности при попытке использовать язык на новых программных платформах

```
503         message =
504         if not hasattr(self, '_headers_buffer'):
505             self._headers_buffer = []
506         self._headers_buffer.append((" %s %d %s\r\n" %
507                                     (self.protocol_version, code, message)).encode(
508                                         'latin-1', 'strict'))
509
510     def send_header(self, keyword, value):
511         """Send a MIME header to the headers buffer."""
512         if self.request_version != 'HTTP/0.9':
513             if not hasattr(self, '_headers_buffer'):
514                 self._headers_buffer = []
515             self._headers_buffer.append(
516                 ("%s: %s\r\n" % (keyword, value)).encode('latin-1'
517
518
519
520
521
522
523         if keyword.lower() == 'connection':
524             if value.lower() == 'close':
525                 self.close_connection = True
526             elif value.lower() == 'keep-alive':
527                 self.close_connection = False
```



1. В каком году был создан Python?

2. Кто создал Python?

3. Почему Python является
простым языком
программирования?

Заключение

Благодаря простоте и гибкости языка Python, его можно рекомендовать пользователям (математикам, физикам, экономистам и т.д.) не являющимся программистами, но использующими вычислительную технику и программирование в своей работе. Программы на Python разрабатываются в среднем в полтора-два (а порой и в два-три) раза быстрее, нежели на компилируемых языках (C, C++, Pascal).

Поэтому, язык может представлять не малый интерес и для профессиональных программистов, разрабатывающих приложения, не критичные к скорости выполнения, а также программы, использующие сложные структуры данных. В частности, Python хорошо зарекомендовал себя при разработке программ работы с графами, генерации деревьев.

Спасибо за внимание!

