

*К.Ю. Поляков,
В.М. Гуровиц*

Язык Python в школьном курсе информатики

Популярность

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

декабрь 2015:

1. Java
2. C
3. C++
- 4. Python**
5. C#
6. PHP
7. Visual Basic
8. JavaScript

в восьмёрке с
2008 года!

Использование

Компании: Скрипты в программах: Скрипты в играх:


1. Google
2. Яндекс
3. CERN
4. NASA
5. ...

1. GIMP
2. Blender
3. Cinema 4D
4. Maya
5. Inkscape
6. Scribus
7. ...

1. Eve Online,
2. Civilization IV
3. Battlefield 2
4. Vampire
5. ...

Linux, Windows,
FreeBSD, Mac OS X,
Android, iOS...

 Кроссплатформенный!

 Свободно распространяемые (*open-source*) реализации!

Для обучения

<https://wiki.python.org/moin/SchoolsUsingPython>

Университеты и колледжи:

- США – 33
 - *University of California*
 - *University of Florida*
 - *University of Iowa*
 - *Massachusetts Institute of Technology (MIT)*
- Канада – 8
 - *University of Toronto*
 - *University of Alberta*
- Франция – 6
- Великобритания – 5
 - *University of Oxford*
- Австралия – 3
- Испания – 3
- ...

Вытесняет С и Java!

Москва:

- СУНЦ МГУ
- школа № 179 (МИОО)
- школа № 2007
- гимназия №1543
- гимназия «Вторая школа»
- школа № 57
- ...

Учебник К.Ю. Полякова и Е.А. Еремина

К **kpolyakov.spb.ru**
Преподавание, наука и жизнь.

Поиск Google™
карта сайта поиск по тэгам

главная школа вуз наука delphi программы походы автор Новости Блог

Программа
Учебник
Купить
Программа
Публикации
Тесты

Учебник информатики К.Ю. Полякова и Е.А. Еремина
Язык Python

Теоретический материал
В связи с растущей популярностью языка Python, авторами учебника разработаны варианты глав по...

Презентации
Презентации предназначены для проведения уроков информатики по учебнику К.Ю. Полякова и Е.А. Еремина (10-11 классы, углубленный уровень) на базе языка Python.

Онлайн-тесты для 10 класса
Глава 8. Алгоритмизация и программирование. Язык Python:

Программы на языке Python
По следующим ссылкам можно скачать готовые примеры программ и проектов, рассмотренных в учебнике. Отладка проводилась в среде [Wing IDE 101](#) для операционной системы *Windows*.

- Программы на языке Python к учебнику для 10 класса (архив ZIP, 37 Кб) 25.03.2014
- Программы на языке Python к учебнику для 11 класса (архив ZIP, 100 Кб) 26.03.2014
- Проекты на языке Python к учебнику для 11 класса (архив ZIP, 155 Кб) 29.03.2014
- Модуль **simpletk** для работы с графическим интерфейсом (архив ZIP, 16 Кб) 29.03.2014

Олимпиады

- Всероссийская олимпиада школьников
- Всероссийская командная олимпиада школьников
- Московская командная олимпиада школьников
- Московская олимпиада школьников
- Командная олимпиада школьников Санкт-Петербурга
- Интернет-олимпиады ИТМО
- Russian Code Cup

Сайты с автоматической проверкой решений:

- informatics.mccme.ru
- codeforces.com
- acm.timus.ru
- acmp.ru

Почему Python?



Динамическая типизация



Переменные не надо объявлять!

```
A = 100           # целое
A = 4.5           # вещественное
A = "Привет!"    # строка
A = [1, 2, 3, 4, 5] # список (массив)
A = (1, "Вася", 3) # кортеж
A = {"Вася": 1, "Петя": 23} # словарь
```


Динамическая типизация

! Функция может вернуть любое значение!

```
def solve ( a, b ):      #  $a \cdot x = b$   
    if a == 0:  
        if b == 0: return True  
        else: return None  
    else:  
        return b / a
```

ЛОГИЧЕСКОЕ
ЗНАЧЕНИЕ

пустое
значение

ЧИСЛО

Структура = отступы

! Отступы обязательны!

Условный оператор:

```
if x > 0:
    a = x
    if x > 1:
        b = x
        c = a + x
    else:
        b = x - 1
        c = x*x
    d = b + c
```

Цикл:

```
s = 0
for x in range(100):
    a = x
    c = 0
    while a > 0:
        c += a % 8
        a = a // 8
    s = s + c
```

! Нет begin-end, {}!

Компактность

Обмен значений двух переменных:

```
c := a  
a := b  
b := c
```

```
a, b = b, a
```

Алгоритм Евклида:

```
while b <> 0 do begin  
  c := a mod b;  
  a := b;  
  b := c  
end;  
writeln ( a );
```

```
a ← b0  
b ← a0 mod b0
```

```
while b:  
  a, b = b, a % b  
print ( a )
```

Списки (массивы)



Нумерация элементов с нуля!

Заполнение

```
for i:=0 to N-1 do  
  A[i] := 0;
```

```
A = [0]*N
```

```
for i:=0 to N-1 do  
  A[i] := i*i;
```

```
A = [i*i  
      for i in range(N)]
```

Максимум

```
M := A[0];  
for i:=1 to N-1 do  
  if A[i] > M then  
    M := A[i];
```

```
M = max(A)
```

Списки (массивы)

Выборка элементов по условию

```
count := 0;
for i:=0 to N-1 do
  if A[i] > 0 then begin
    count := count + 1;
    B[count] := A[i]
  end;
```

```
B = [x for x in A if x > 0]
```

```
B = [x
      for x in A
      if x > 0]
```

Сортировка

```
for i:=0 to N-1 do
  for j:=N-2 downto i do
    if A[j] > A[j+1] then begin
      c:=A[j]; A[j]:=A[j+1];
      A[j+1]:=c
    end;
```

```
A.sort()
```

Списки (массивы)

Реверс массива

```
for i:=1 to N div 2 do begin
  c := A[i];
  A[i] := A[N-i+1]
  A[N-i+1] := c;
end;
```

```
A = A[::-1]
```

Списки (массивы)

Удалить повторяющиеся элементы массива

```
count := 0;
for i:=1 to N do begin
  j := 1;
  while (j <= count) and (A[i] <> B[j]) do
    j := j + 1;
  if j > count then begin
    count := count + 1;
    B[count] := A[i]
  end
end;
```

```
B = list( set(A) )
```

Длинная арифметика

Задача: вычислить $100! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot 100$

```
const N = 33;
      d = 1000000;
var A: array[0..N] of longint;
    i, k, s, r: integer;
begin
  A[0] := 1;
  for k := 2 to 100 do begin
    r := 0;
    for i := 0 to N do begin
      s := A[i]*k + r;
      A[i] := s mod d;
      r := s div d
    end
  end
  { вывод длинного числа из
    массива A }
end.
```

```
A = 1
for i in range(2,101):
    A = A*i
print ( A )
```

или так:

```
import math
print ( math.factorial(100) )
```


Алфавитно-частотный словарь

Задача: построить алфавитно-частотный словарь

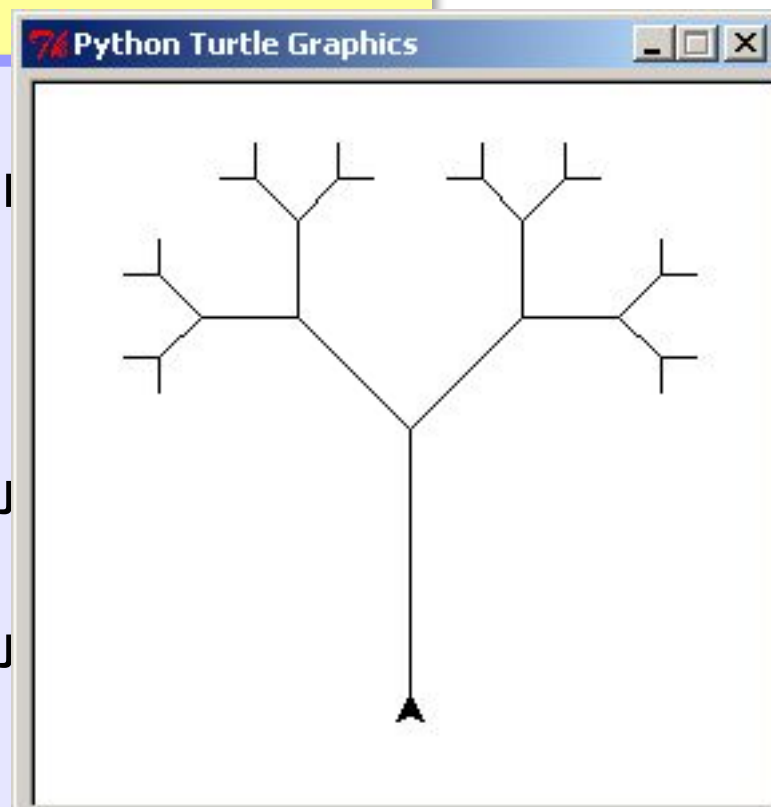
```
uses WordList;
var F: text; s: string;
    L: TWordList; p: integer;
begin
  Assign(F, 'input.txt'); Reset(F);
  SetLength(L.data, 0);
  L.size := 0;
  while not eof(F) do begin
    readln(F, s);
    p := Find ( L, s );
    if p >= 0 then
      L.data[p].count := L.data[p].count + 1;
    else begin
      p := FindPlace ( L, s );
      InsertWord ( L, p, s );
    end
  end;
  Close(F);
  ...
end.
```

Модуль ≈ 50 строк!

```
D = {}
for line in open("input_list.txt"):
  word = line.strip()
  if word:
    D[word] = D.get(word, 0) + 1
```

Черепашня графика

```
from turtle import *  
  
def tree ( levels , length ) :  
    i- from turtle_rus import *  
    def дерево ( уровни , длина ) :  
        if уровни > 0 :  
            вперед ( длина )  
            влево ( 45 )  
            дерево ( уровни-1 , дл  
            вправо ( 90 )  
            дерево ( уровни-1 , дл  
            влево ( 45 )  
            назад ( длина )  
    seth ( новый_курс ( 90 ) )  
    дерево ( 5 , 100 )
```



Ввод с клавиатуры

```
var N, NN: integer;
```

```
...
```

```
write ( "Введите число " );
```

```
read ( N );
```

```
NN := N*2;
```

```
write ( NN );
```

тип неизвестен!

```
N = input ( "Введите число " )
```

```
NN = N * 2
```

```
print ( NN )
```



Строка!

Введите число **12**

1212

```
N = int(input("Введите число "))
```

```
NN = N * 2
```

```
print ( NN )
```

Введите число **12**

24

Ввод с клавиатуры

```
var a, b, c: integer;  
...  
write ( "Введите три числа " );  
read ( a, b, c );
```

```
s = input ( "Введите три числа " ) # "1 2 3"  
      s.split()
```

применить к каждой
части функцию `int`

разбить строку
по пробелам

```
q = s.split() # список ["1", "2", "3"]  
a = int(q[0]) # 1  
b = int(q[1]) # 2  
c = int(q[2]) # 3
```

Ввод массива с клавиатуры

```
const MAX = 1000;  
var i, N: integer;  
    A: array[0..MAX-1] of integer;  
...  
readln ( N );  
for i:=0 to N-1 do  
    read ( A[i] );
```

```
N = int(input())
```

```
input()
```

N не нужно!

C2 (демо-2014)

Задача: найти и вывести максимальное значение среди трёхзначных элементов массива, не делящихся на 9.

```
max := 0;
for i := 0 to N-1 do
  if (100<=a[i]) and (a[i]<=998) and
    (a[i] mod 9<>0) and (a[i]>max) then
    max := a[i];
```

```
if max > 0 then
  writeln(max)
else writeln("Не найдено")
```

```
a = [x for x in a
      if 100 <= x and x <= 999
      and x % 9 != 0]
if len(a) > 0:
  print(max(a))
else:
  print("Не найдено")
```

C4

Задача: В первой строке вводится количество учащихся

N, далее идут N строк в формате:

<Фамилия> <Инициалы> <номер школы>

Номер школы < 100.

Из какой школы было меньше всего участников (таких школ может быть несколько)?

C4 (решение)

```
const LIM = 99;
var C:array[1..LIM] of integer;
  i, p, N = int(input())
  s:stri
begin
  schCount = {}
  for k:
    readln
    for i:
      x, x, school = input().split()
      read
      p :=
      Dele
      Val(
      C[k]
    minCount = min(schCount.values())
  end;
  Min :=
  for k:
    if (
      Mi
  for k:
    if C
    wr
end.
```

словарь
«НОМЕР-СЧЁТЧИК»

C4-2014

Задача: В первой строке вводится длина N последовательности, далее идут N положительных чисел, по одному в строке, а затем – **контрольная сумма R** – наибольшее произведение двух различных элементов последовательности, которое **делится на 21**.

Проверить правильность контрольной суммы.

C4-2014 (решение)

```
N = int(input())
M3 = M7 = M21 = M = 0

for i in range(N):
    x = int(input())
    if x % 21 != 0:
        if x % 3 == 0: M3 = max(M3, x)
        if x % 7 == 0: M7 = max(M7, x)
    if x % 21 == 0 and x > M21:
        M = max(M21, M)
        M21 = x
    else: M = max(M, x)

R0 = int(input())
R = max(M3*M7, M21*M)

if R == R0: print("Контроль пройден")
else:      print("Контроль не пройден")
```

Функциональное программирование

Функция – объект, который можно передавать в другие функции как аргумент и возвращать как результат других функций.

Языки LISP, Scheme, Haskell, Scala, Erlang, F#, ...

Функциональное программирование

Генератор списков

```
for i:=0 to N do  
  A[i] := i*i;
```

```
A = [i*i  
      for i in range(N)]
```

```
count := 0;  
for i:=0 to N do  
  if f(i) = 0 then begin  
    A[count] := i*i;  
    count := count + 1;  
  end;
```

```
A = [i*i  
      for i in range(N)  
      if f(i) == 0 ]
```

Самая длинная строка в файле

```
s = max ( open("test.txt"), key=len )
```

КЛЮЧ
сортировки

Отбор элементов по условию

Задача: Отобратить в массив В все элементы массива А, кубы которых больше 100.



```
count := 0;
for i:=0 to N-1 do
  if A[i]*A[i]*A[i] > 100 then begin
    count := count + 1;
    B[count] := A[i]
  end;
```



```
B = [x
      for x in A
      if x**3 > 100]
```

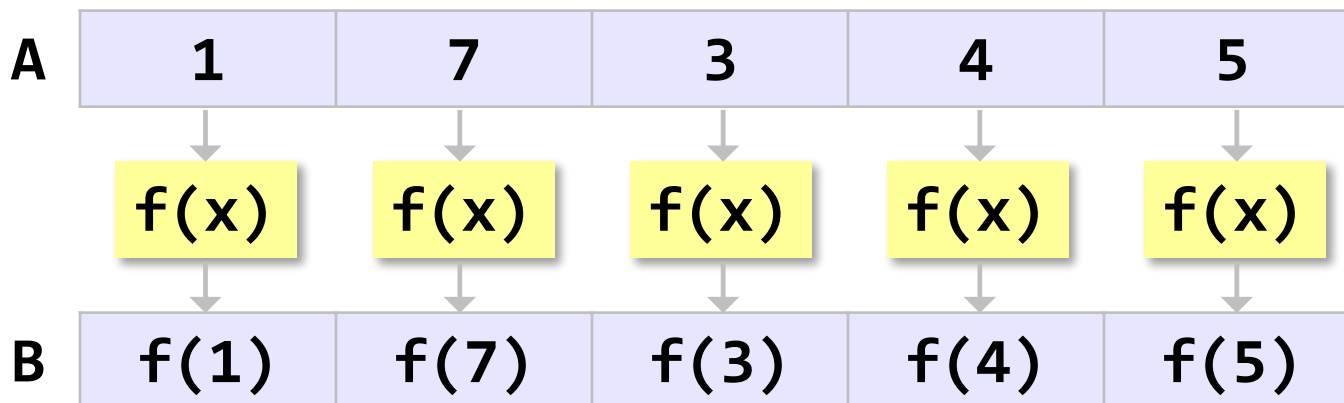


Меньше шансов на случайную ошибку!

Обработка всех элементов списка (**map**)

функция передаётся как
параметр функции

```
for i:=0 to N-1 do B = list( map(math.sin, A) )  
    B[i] := sin(A[i]);
```



Обработка всех элементов списка (**map**)

```
function x5(x: integer): integer;  
begin  
    x5 := x*x*x*x*x;  
end;  
...  
for i:=0 to N-1 do  
    B[i] := x5(A[i])
```

```
def x5 ( x ):  
    return x**5  
  
B = list( map(x5, A) )
```

Функции без имени (**λ-функции**)

```
B = list( map( lambda x: x**5 , A) )
```

Переменные-функции

```
sqr = lambda x: x**2
```

```
prod = lambda p, x: p * x
```

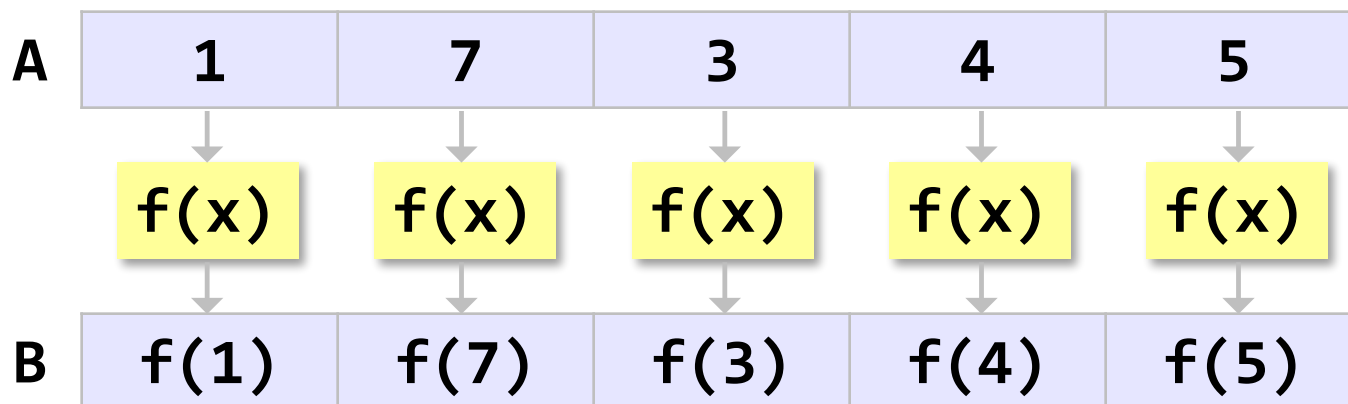
Map – Reduce

Задача: Определить, сколько страниц со словом «Python» есть на всех серверах компании.

```
for i:=0 to N-1 do  
  B[i] := f(A[i]);
```

Map

каждый сервер
отдельно!



```
result := B[0];  
for i:=1 to N-1 do  
  result := G(result, B[i]);
```

Reduce

Map – Reduce

```
from functools import reduce
```

функция
обработки

```
B = list( map( f, A ) )
```

```
p = reduce( G, B )
```

функция
накопления

Пример:

```
B = list( map( lambda x: x*x , A ) )
```

```
p = reduce( lambda p, x: p+x , B )
```

Функциональное программирование

Замена цикла рекурсией

```
function sumDigits(x: integer): integer;  
var s: integer;  
begin  
    s := 0;  
    while x > 0 do  
        s := s + x % 10;  
        x := x // 10;  
    end;  
    sumDigits := s;  
end;
```



Меньше шансов на случайную ошибку!

```
def sumDigits ( x ):  
    if x > 0:  
        return ( x % 10 +  
                sumDigits(x//10) )  
    else:  
        return 0
```

```
def sumDigits ( x ):  
    return sum( map(int, str(n)) )
```

Функциональное программирование

Замена цикла рекурсией

```
def isPalindrome ( s ):  
    for i in range(len(s) // 2):  
        if s[i] != s[-1-i]:  
            return False  
    return True
```

```
def isPalindrome ( s ):  
    if len(s) > 1:  
        return s[0] == s[-1] and \  
            isPalindrome(s[1:-1])  
    else:  
        return True
```

Функциональное программирование

Замена цикла рекурсией

(if(z)x(/)) ✓

():-) ✗

()())(()) ✗

```
def balance ( s ):
```

```
    opened = 0
```

```
    for c in s:
```

```
        if c == '(':
```

```
            opened
```

```
        elif c == ')':
```

```
            if ope
```

```
                retu
```

```
            opened
```

```
    return ope
```

```
def balance ( s, opened = 0 ):
```

```
    if len(s) == 0:
```

```
        return opened == 0
```

```
    elif s[0] == '(':
```

```
        return balance(s[1:], opened+1)
```

```
    elif s[0] == ')':
```

```
        return opened > 0 and \
```

```
            balance(s[1:], opened-1)
```

```
    else:
```

```
        return balance(s[1:], opened)
```

Функциональное программирование

Функция, возвращающая функцию

```
def createFunc ( valid ):  
    def f ( s ):  
        return s == valid  
    return f
```

запомнили!

получили функцию

```
check = createFunc ( "pass" )  
print ( check("pass") )           # True  
print ( check("bla-bla-bla") )# False
```

Графический интерфейс

Стандартный модуль **tkinter**.



```
from simpletk import *

app = TApplication("Шестнадцатеричная система")
app.size = (250, 36)
app.position = (200, 200)

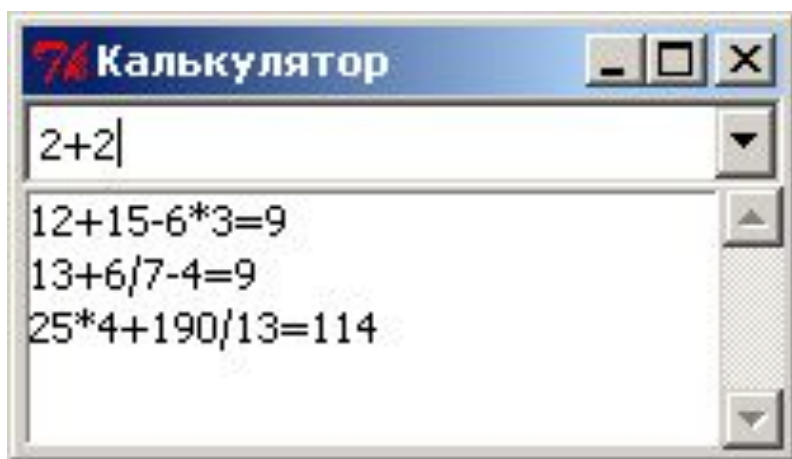
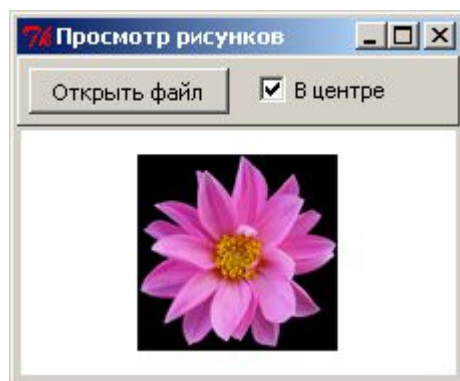
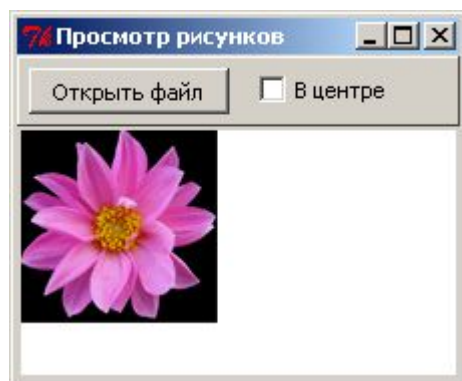
f = ("Courier New", 14, "bold")
hexLabel = TLabel(app, text="?", font=f, fg="navy" )
hexLabel.position = (155, 5)

def onNumChange(sender):
    hexLabel.text = "{:X}".format(sender.value)

decEdit = TIntEdit(app, width=12, font=f)
decEdit.position = (5, 5)
decEdit.text = "1001"
decEdit.onChange = onNumChange

app.Run()
```

Графический интерфейс



Альтернативы:
wxPython, PyGTK, PyQt.

Библиотеки

- **math** – математические функции
- **fractions** – рациональные дроби
- **decimal** – десятичная арифметика
- **random** – случайные числа, случайный выбор, случайная перестановка элементов
- **re** – регулярные выражения
- **itertools** – перестановки, сочетания
- **webbrowser**, **urllib**, **http**, **ftplib**, ... – работа с сетью
- **tkinter** – графический интерфейс
- **pyGame** – программирование игр
<http://www.pygame.org/news.html>
- **simplegui** – модуль для программирования игр на сайте
<http://www.codeskulptor.org>

Достоинства

- низкий порог входа
`print ("Привет!")`
- применяется в профессиональных разработках
- понятный синтаксис, отступы
- компактные решения (за счёт встроенных средств)
- язык **более высокого уровня**, чем С и Паскаль (списки, словари, ...)
- поддержка любых языков (utf-8)
- большая библиотека
- возможность разработки программ с графическим интерфейсом
- различные подходы к программированию (императивный, ООП, функциональный, ...)

Грабли



Габбли

! Нумерация элементов строк и массивов с нуля!

! Последний элемент среза не включается!

0	1	2	3	4	5	6	7
7	12	5	8	18	34	40	23

$A[1:3] \rightarrow [12, 5]$

$A[2:3] \rightarrow [5]$

$A[:3] \rightarrow A[0:3] \rightarrow [7, 12, 5]$

$A[5:] \rightarrow A[5:8] \rightarrow [34, 40, 23]$

$A[3:-2] \rightarrow A[3:6] \rightarrow [8, 18, 34]$

$len(A) - 2$

Габбли

! Строки нельзя изменять!

```
var s: string;  
...  
for i:=1 to Length(s) do  
  if s[i] = 'a' then s[i] := 'b';
```

~~s[i] := "b"~~

```
for i in range(len(s)):  
  if s[i] == "a":  
    s = s[:i] + "b" + s[i+1:]
```

с начала
до s[i-1]

от s[i+1]
до конца

Габбли

! Переменные не нужно объявлять!

```
if a > b:  
    print("OK")  
else:  
    this is spam
```

«We are all consenting adults here».

**нет ошибки, пока
не пришли сюда!**

```
x1 = 0  
if a > b:  
    x1 = 1
```

нет ошибки!

! Требуется тщательное тестирование!

Габбли

! Опасные опечатки!

```
s = "1"  
...  
s + s + "2"  
print( s )
```

1

должно быть
s = s + "2"

Габбли



Нет контроля типов параметров!

```
def trimLeft ( s ):  
    while len(s) and s[0] == ' ':  
        s = s[1:]  
    return s
```

```
print ( trimLeft(" 123  ") )  
print ( trimLeft(123) )
```

**нет ошибки, пока
не пришли сюда!**

Габбли

! Нет контроля возвращаемого значения!

```
def trimLeft ( s ):  
    while len(s) and s[0] == ' ':  
        s = s[1:]  
    return s
```

```
print ( trimLeft(" 123 ") )
```

None

```
A = [2, 1, 3, 7, 5, 4]  
B = A.sort()  
print ( B )
```

None

Гrabли

! Список – это ссылка!

```
A = [1, 2, 3]
B = A
```

A → [1, 2, 3]
B → [1, 2, 3]

A[0] = 0

A → [0, 2, 3]
B → [0, 2, 3]

```
A = [1, 2, 3]
B = A[:]
```

копия массива A

A → [1, 2, 3]
B → [1, 2, 3]

A[0] = 0

A → [0, 2, 3]
B → [1, 2, 3]

Грабли

Глобальные и локальные переменные

```
state = 0

def changeState(a, b):
    if a > b:
        state = 1

changeState ( 2, 1 )
print ( state )
```

**Новая
локальная
переменная!**

0

```
state = 0

def changeState(a, b):
    global state
    if a > b:
        state = 1

changeState ( 2, 1 )
print ( state )
```

Грабли

Глобальные и локальные переменные

```
x = 0
def f():
    print ( x )
f()
```

ОК!

**ИСПОЛЬЗОВАНИЕ
ЛОКАЛЬНОЙ
ПЕРЕМЕННОЙ ДО
ОПРЕДЕЛЕНИЯ!**

```
x = 0
def f():
    x += 3
    print ( x )
f()
```

```
x = 0
def f():
    global x
    x += 3
    print ( x )
f()
```

Габбли



Все данные объектов – открытые (*public*)!

```
class dog:  
    def __init__(self, _age):  
        self.age = _age
```

```
def spam():  
    tuzik.age = 100
```

```
tuzik = dog(5)
```

```
spam()
```

```
print(tuzik.age)
```

нет private!

Недостатки

- две несовместимых версии: **2.x** и **3.x**
- нужен **интерпретатор** для выполнения
- низкая **скорость** (в 100 раз ниже, чем на C)
- увеличенный расход **памяти**
- нет проверки **типов** данных
- нет доступа к низкоуровневым средствам (размещение данных в памяти, **аппаратура**)
- неклассическая объектная модель (нет **private** и **protected**)
- нет надёжных **RAD-систем** для программ с графическим интерфейсом

Дистанционное образование

Введение в *computer science* и программирование на языке Python (англ.)

<https://www.edx.org/course/mitx/mitx-6-00-1x-introduction-computer-1498>

Введение в интерактивное программирование на языке Python (англ.)

<https://www.coursera.org/course/interactivepython>

Онлайн-среда для интерактивного программирования на языке Python

<http://www.codeskulptor.org/>

Языке Python на сайте Codecademy (англ.)

<http://www.codecademy.com/ru/tracks/python>

Дистанционное образование

Материалы по программированию на языке Python на сайте школы 179 г. Москва

<http://server.179.ru/~dk/python.html>

Курс Д. Кириенко на сайте «Дистанционная подготовка по информатике»

<http://informatics.mccme.ru/course/view.php?id=156>

Интерактивный учебник языка Python
(автор курса – Д. Кириенко)

<http://pythontutor.ru>

Онлайн-среды:

<http://www.codeskulptor.org/>

<http://pythontutor.com/>

<http://ideone.com/>

http://www.compileonline.com/execute_python_online.php

<http://www.skulpt.org/>

визуализация
выполнения!

Конец фильма

ПОЛЯКОВ Константин Юрьевич

д.т.н., учитель информатики

ГБОУ СОШ № 163, г. Санкт-Петербург

kpolyakov@mail.ru

ГУРОВИЦ Владимир Михайлович

учитель информатики ФМШ № 2007, г. Москва,

координатор проекта дистанционной подготовки по

информатике (informatics.mccme.ru)