Python для школьников

Занятие 1. «Знакомство с Python: ввод/вывод, целочисленная арифметика»

Паньгина Н.Н. 2018

История языков программирования

Машинные коды → Ассемблеры → Языки высокого уровня → Объектно-ориентированные

История языка Python

Первая версия языка – 1994 год

Вторая версия языка – 2000 год

Третья версия языка – 2008 год

Гвидо ван Россум

Историческая справка

Гвидо ван Россум известен среди разработчиков как «Великодушный пожизненный диктатор Python».

Это означает, что он продолжает наблюдать за разработкой языка и принимает окончательные решения по его развитию, когда это необходимо.

Возможности языка Python

Язык программирования **Python 3 —** это мощный инструмент для создания программ самого разнообразного назначения, доступный даже для новичков.

- **Python -** интерпретируемый язык программирования
- Кроссплатформенность и бесплатность
- Понятный и лаконичный синтаксис
- Динамическая типизация
- Хорошая поддержка модульности
- Поддержка объектно-ориентированного программирования
- И др.

Установка языка Python и среды разработки

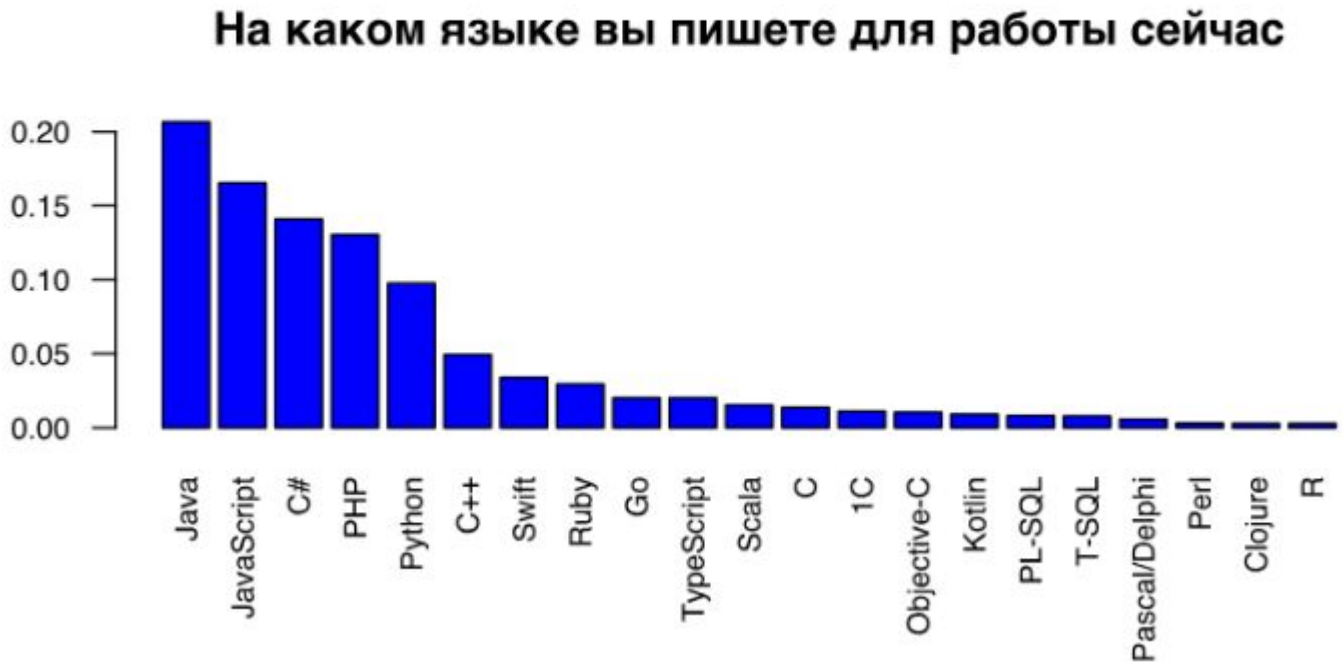
- Скачать дистрибутив с официального сайта http://www.python.org
- Рекомендуется использование версии 3..., как более современные
- Прямая ссылка на дистрибутив 3.6.0 для Windows: 32 бита, 64 бита. При установке рекомендуется выбирать вариант "Install for all users"
- Можно использовать IDLE – стандартную среду разработки для Python
- Также есть среда JetBrains PyCharm, более богатая возможностями, но и более сложная в использовании для начинающих
- Простая кроссплатформенная бесплатная среда для обучения – Wing IDE 101 (http://wingware.com/downloads/wingide-101)

2017г. Популярность Python

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.5
4. C++		97.1
5. C#		87.7
6. R		87.7
7. JavaScript		85.6
8. PHP		81.2
9. Go		75.1
10. Swift		73.7

2018г. Популярность Python

Коммерческое использование



2019г. Популярность Python

Language Types (click to hide)

Web | Mobile | Enterprise | Embedded

Language Rank	Types	Spectrum Ranking
1. Python	Web, Enterprise, Embedded	100.0
2. C++	Mobile, Enterprise, Embedded	99.7
3. Java	Web, Mobile, Enterprise	97.5
4. C	Mobile, Enterprise, Embedded	96.7
5. C#	Web, Mobile, Enterprise	89.4
6. PHP	Web	84.9
7. R	Enterprise	82.9
8. JavaScript	Web, Mobile	82.6
9. Go	Web, Enterprise	76.4
10. Assembly	Embedded	74.1

Оболочка IDLE для Python

Главное меню

Пункт меню «Настройки»

Встроенный калькулятор

Ввод/вывод данных

Окно настроек

Python 3.4.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015,
19:28:18) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for
more information.
>>> (2+8)*(54-14)/40
10.0
>>> ((2+8)*(54-14)/40))**3
SyntaxError: invalid syntax
>>> ((2+8)*(54-14)/40)**3
1000.0
>>>
```

Settings

Fonts/Tabs | Highlighting | Keys | General | Extensions

Base Editor Font

Font Face :

Courier New
Courier New Baltic
Courier New CE
Courier New CYR
Courier New Cyr

Size : 14

Bold

AaBbCcDdEe
FfGgHhIiJjK
1234567890
#:+=(){}[]

Indentation Width

Python Standard: 4 Spaces!

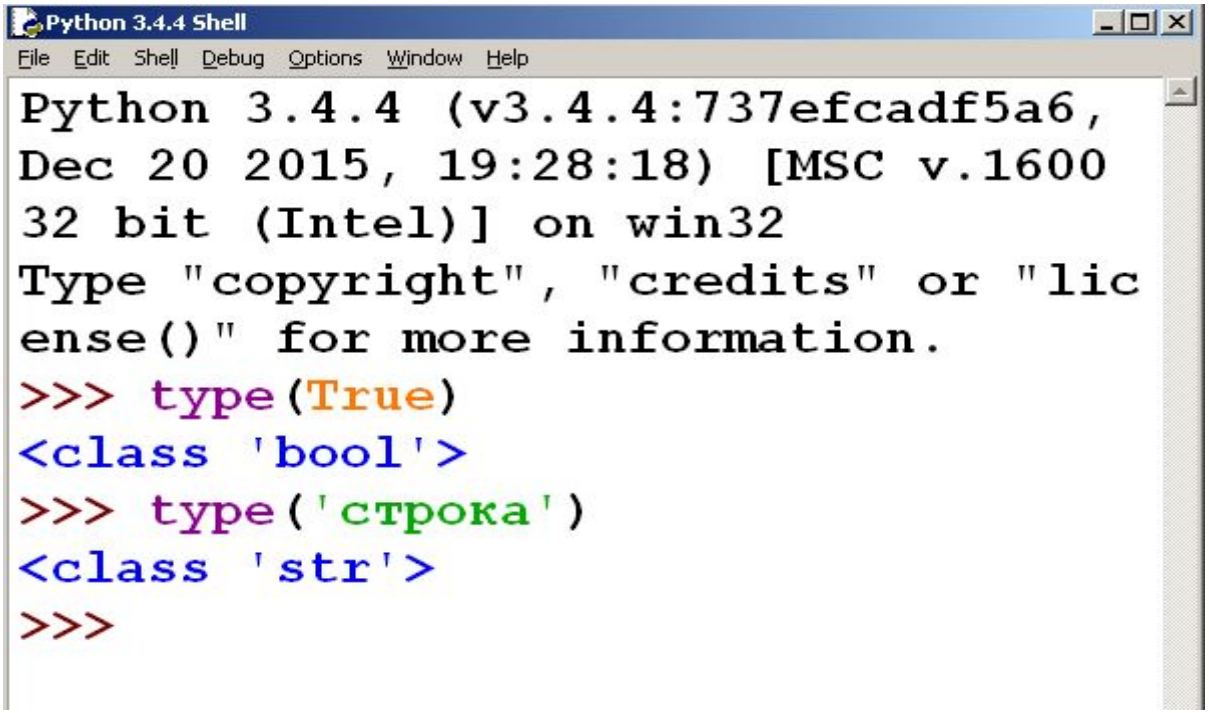
4

2 4 6 8 10 12 14 16

Ok | Apply | Cancel | Help

Типы данных в Python

- Числовые (целые – **int** (43), вещественные – **float** (3.1415))
- Строковые – **str** (‘Строка’, “Текст”)
- Булевские – **bool** (True, False)



Основные операции в Python

Для чисел:

A + B — сумма;
A - B — разность;
A * B — произведение;
A / B — частное;
A // B — деление нацело;
A % B — остаток от деления;
A ** B — возведение в степень.

Для строк:

A + B — конкатенация;
A * n — повторение n раз, значение n должно быть целого типа.

Приоритеты операций в Python

1. Выполняются операции возведения в степень **справа налево.**
2. Выполняются унарные минусы (отрицания).
3. Выполняются умножения и деления слева направо. Операции умножения и деления имеют одинаковый приоритет.
4. Выполняются сложения и вычитания слева направо. Операции сложения и вычитания имеют одинаковый приоритет.

<u>Пример 1</u>: чему будет равно 1 + 2 * 3 ** 1 + 1? **8**

<u>Пример 2</u>: чему будет равно 3 ** 3 ** 3 ? **7625597484987**

Преобразование типов в Python

Для этого используются функции, одноименные с именем типа, то есть **int, float, str**.

Например, int ('123') вернет целое число 123, а str (123)вернет строку '123'.

<u>Пример</u>:

Что будет в результате **str(2 + 2) * int (‘2’ + ‘2’)?**

‘4444444444444444444444’

Работа со встроенным калькулятором

Упражнения (для самостоятельного выполнения):

- **Вычислить выражение (16386 - 396) : 78 + 402 x 306** — **123217**
- **Вычислить факториал числа 17!** — **355687428096000**
- **Какой цифрой оканчивается сумма $29^{2019} + 5$** — **4**
- **Найти частное и остаток от деления числа 23^{13} на целую часть от числа [2019:67]** — **1680121206454891**, **2**, **23**

Окно написания программы на Python

Главное меню

Запуск программы

Код программы

Game_Orel-Reshka.py - C:\Python34\Game_Orel-Reshka.py (3.4.4)

File Edit Format Run Options Window Help

Python Shell

Check Module Alt+X

Run Module F5

```
from rand
n=int(inp                    оличество раундов "))
k0=0
k1=0
print("Монета Чел Комп")
for i in range(n):
    for j in range(2):
        m=randint(0,1)
        if m==0:
            k0+=1
        else:
            k1+=1
        print("  ",m,"  ",k0,":",k1)
if k0>k1:
    print ("Выиграл человек со счетом",k0,":",k1)
elif k1>k0:
    print ("Выиграл компьютер со счетом",k1,":",k0)
else:
    print ("Ничья",k1,":",k0)
```

Ln: 1 Col: 0

Оператор вывода на Python

Untitled

File Edit Format Run Options Window Help

```
print (
```

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

print_1.py - C:/Python34/print_1.py (3.4.4)

File Edit Format Run Options Window Help

```
print (10, 20, 50.5, 3141592e-6)
print (10//2 + 20/5 - 5**2 + 3 * 6)
print (10,20,30,sep=":")
print (10,20,30,sep="   ")
print (100)
print (200)
print (400,end = ' ')
print (500,end = ' ')
print (600)
print ("Я сразу смазал карту будня,")
print ('Плеснувши краску из стакана.')
```

Ln: 10 Col: 0

Python 3.4.4 Shell

File Edit Shell Debug Options Window Help

```
C:/Python34/print_1.py ==========
=============
10 20 50.5 3.141592
2.0
10:20:30
10   20   30
100
200
400 500 600
Я сразу смазал карту будня,
Плеснувши краску из стакана.
>>>
```

Ln: 38 Col: 4

Конец строки

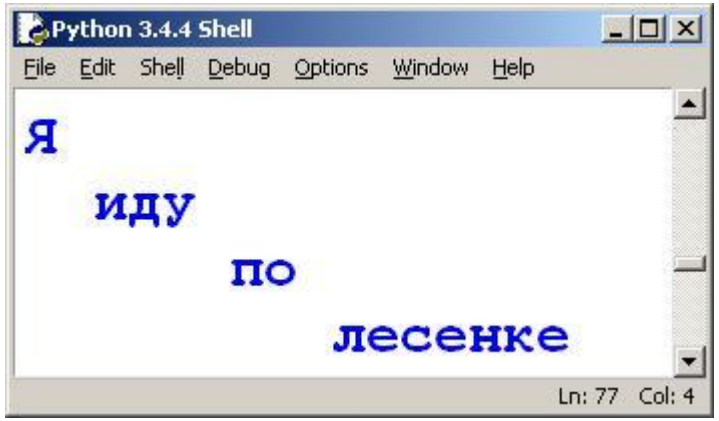
Разделитель

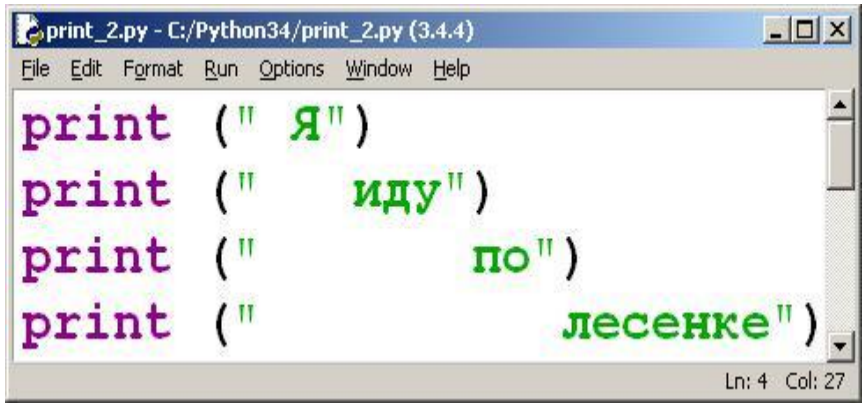
Результат работы программы

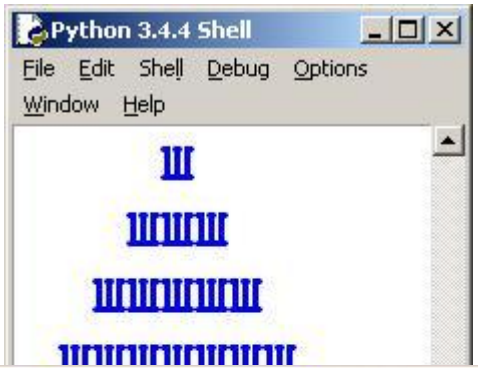
Закрепление пройденного

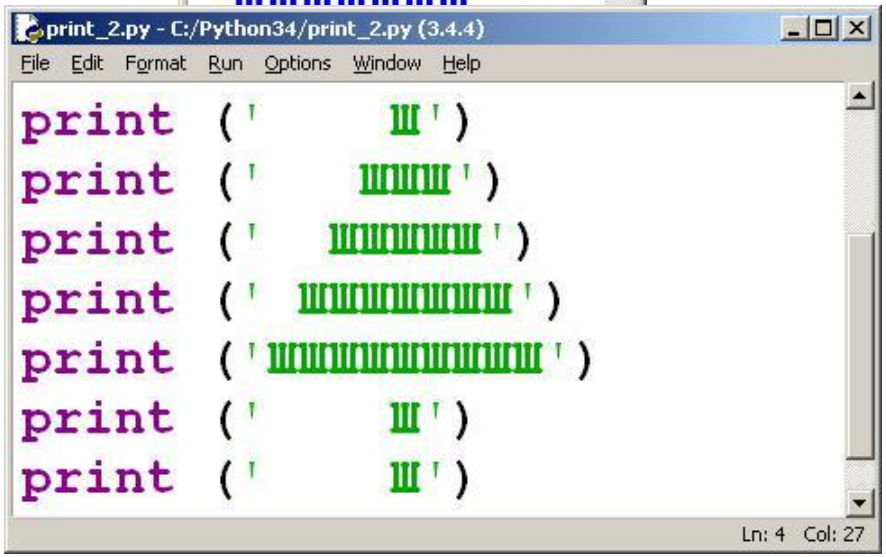
Упражнения (для самостоятельного выполнения):

- Напечатать текст «лесенкой»
- Нарисовать «Елочку»









Операторы ввода и присваивания в Pythone

inp_1.py - C:/Python34/inp_1.py (3.4.4)

File Edit Format Run Options Window Help

```
a = input(
```

```
input([prompt]) -> string
```

Подсказка

Вводятся понятия переменных и констант

- **Написать программу «Диалог с компьютером»**

inp_1.py - C:/Python34/inp_1.py (3.4.4)

File Edit Format Run Options Window Help

```
# Диалог с компьютером
a = input("Как тебя зовут? ")
print ("Привет, ", a, "! А я ПК!", sep = "")
n = input("Сколько тебе лет? ")
print ("Тебе ", n, ", а мне всего 5!", sep = "")
h = input("Назови свое хобби ")
print ("Твое ", h, ", а мое - решать задачи!", sep = "")
```

Ln: 7 Col: 13

Компьютер

Результат работы программы

Задачи на линейные алгоритмы

- **Вводится целое число. Вывести предыдущее и последующее числа.**

<u>Пример</u>

Входные данные:

8

Выходные данные:

Перед 8 число 7

За 8 число 9

- **Вводятся длины двух сторон прямоугольника (в метрах). Вывести его площадь и периметр.**

<u>Пример</u>

Входные данные:

1-я сторона 8

2-я сторона 5

Выходные данные:

Площадь 40 кв.м

Периметр 26 м

Python 3.4.4 Shell

File Edit Shell Debug Options Window Help

1_2_0.py - F:/1_Python/Python/1_2_0.py (3.4.4)

File Edit Format Run Options Window Help

```
a=int(input())
print('Перед',a,'число',
print('За',a,'число',a+1
```

Python 3.4.4 Shell

3_1.py - F:/1_Python/Python/3_1.py (3.4.4)

File Edit Format Run Options Window Help

```
a=int(input("1-я сторона "))
b=int(input("2-я сторона "))
print('Площадь',a*b,'кв.м')
print('Периметр',(a+b)*2,'м')
```

Ln: 4 Col: 29

Ln: 9 Col: 4

Задачи на линейные алгоритмы

(для самостоятельного выполнения)

- **Вводится трехзначное число. Вывести сумму его цифр.**

<u>Пример</u>

Входные данные:

876

Выходные данные:

21

- **Вводится целое число. Вывести следующее за ним четное число.**

<u>Пример</u>

Входные данные:

7 8

Выходные данные:

8 10

- **Дано число n. С начала суток прошло n минут. Определите, сколько часов и минут будут показывать электронные часы в этот момент. Программа должна вывести два числа: количество часов (от 0 до 23) и количество минут (от 0 до 59). Учтите, что число n может быть больше, чем количество минут в сутках.**

<u>Пример</u>

Входные данные:

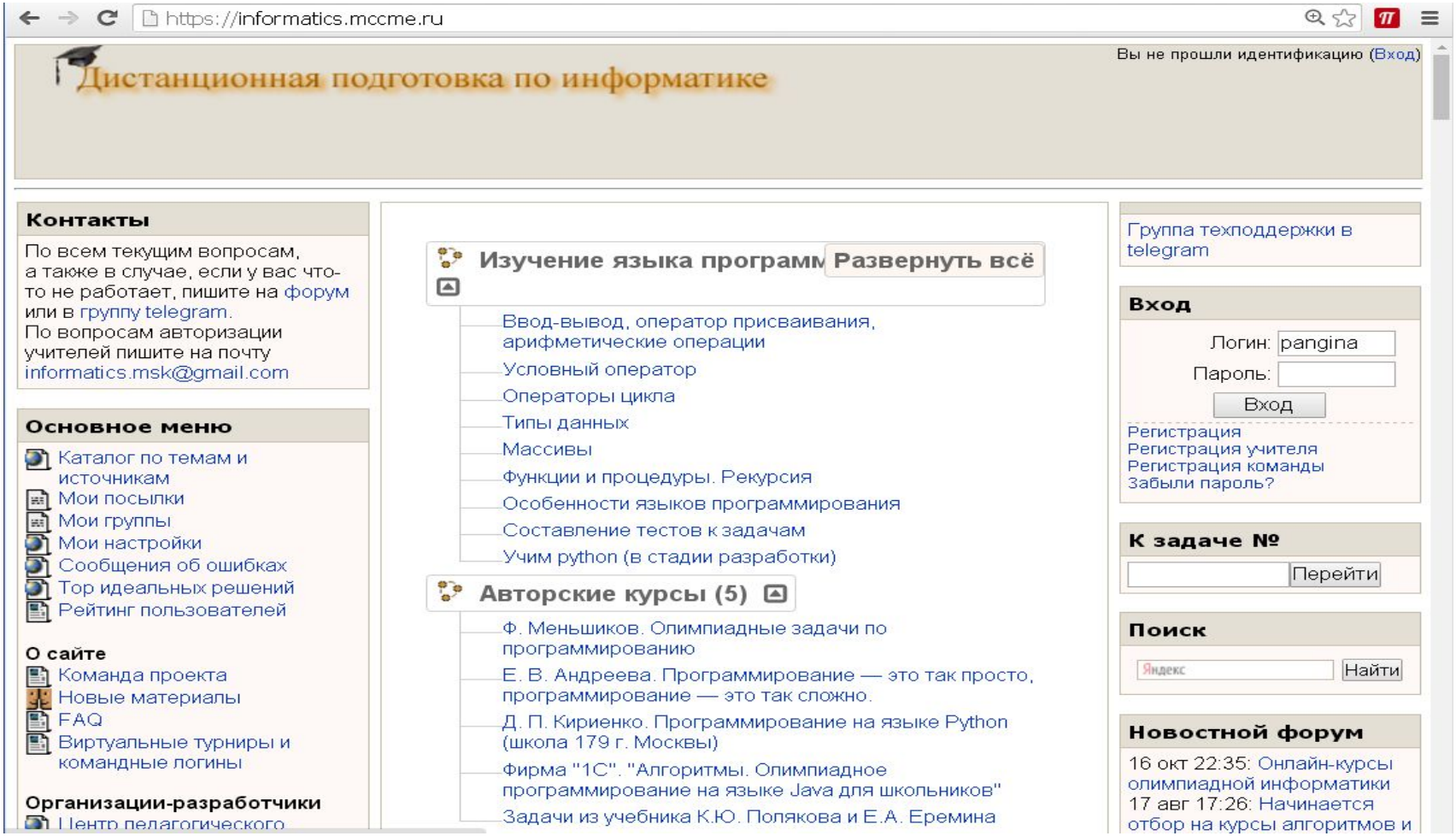
150 1441

Выходные данные:

2:30 0:1

Дистанционная подготовка

- https://informatics.mccme.ru/
- https://informatics.msk.ru/



СПАСИБО ЗА ВНИМАНИЕ!

Паньгина Нина Николаевна,
E-mail: nina_pangina@mail.ru