

**Знакомство с языком  
программирования Python. Ввод.  
Вывод. Оператор присваивания.  
Математические операции**

Программирование – это создание компьютерных программ.

Язык программирования – это язык, понятный компьютеру.

# Установка

Я введение в язык python типы п... X | Я tonny — Яндекс: нашлось 13 м... X | Th Thonny, Python IDE for beginner... X +

thonny.org

Сервисы Я Яндекс Вход RUDN Портал ЕИС RUDN Телекоммуникации... ScienceDirect.com |... BookReader - Библ... Вход Электронная библ... COMS | Conference... Mail.ru Google

## Thonny

Python IDE for beginners

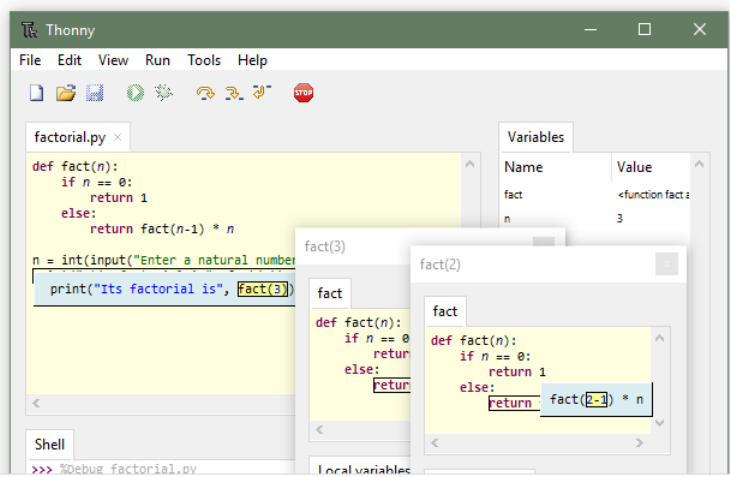
Download version **3.2.7** for  
[Windows](#) • [Mac](#) • [Linux](#)

For the curious: [3.3.ob2](#)

**NB! Windows installer is signed with new identity and you may receive a warning dialog from Defender until it gains more reputation.**

**Just click "More info" and "Run anyway".**

Fork me on GitHub



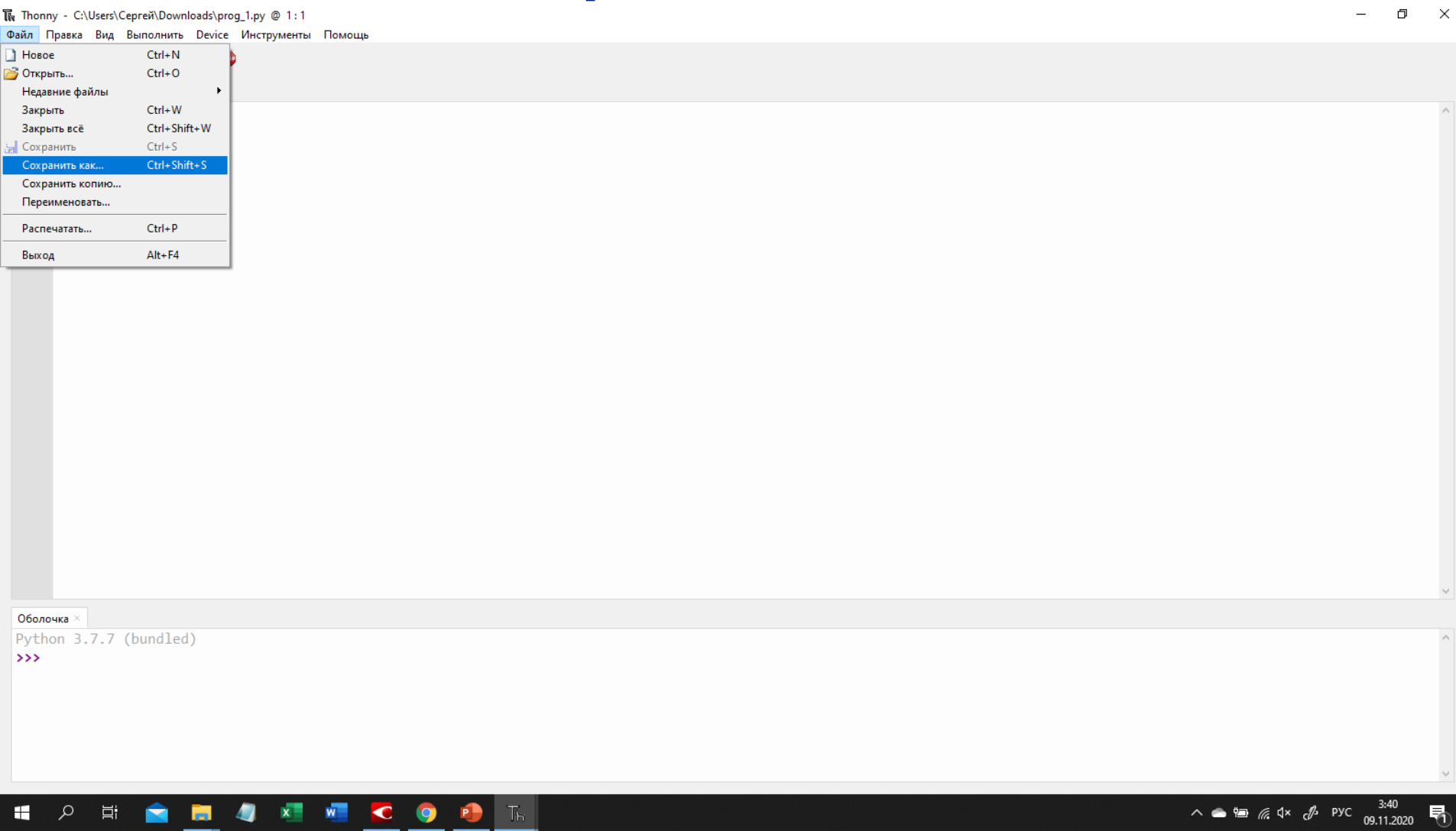
The screenshot shows the Thonny IDE window with a file named 'factorial.py'. The code defines a recursive function 'fact(n)' and uses it to calculate the factorial of 3. The interface includes a menu bar (File, Edit, View, Run, Tools, Help), a toolbar, a code editor, a 'Variables' panel, and a 'Shell' at the bottom. Overlaid on the IDE are three small windows illustrating the recursive process: 'fact(3)', 'fact(2)', and 'fact(1)'. Each window shows the function definition with the current value of 'n' and the return statement being executed. The 'fact(3)' window shows 'fact(3)' returning 'fact(2) \* 3', 'fact(2)' returning 'fact(1) \* 2', and 'fact(1)' returning '1'. The 'Shell' at the bottom shows the command '%debug factorial.py'.

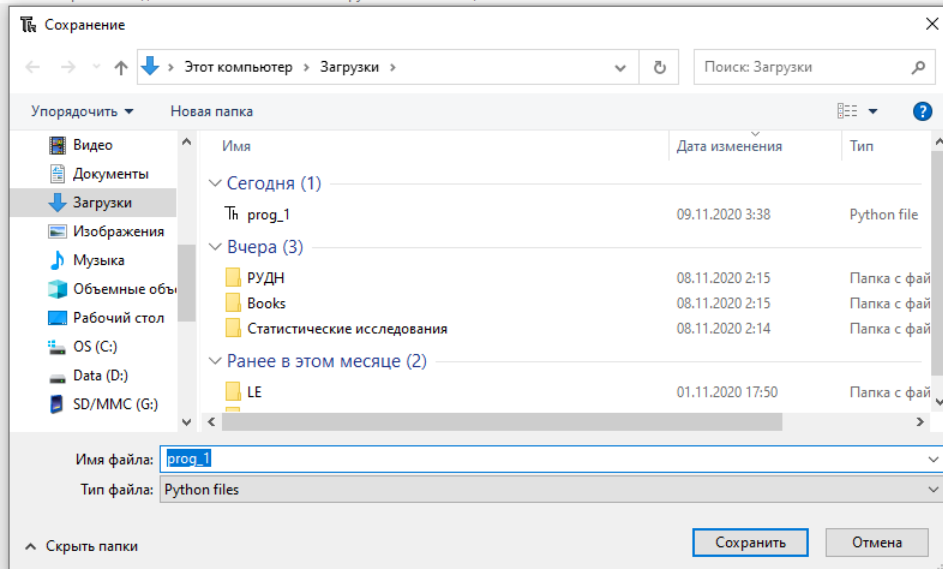
Windows taskbar: konspekt\_uroka\_....docx, urok\_1\_znakomst....pptx, YandexPackLoader.exe

System tray: 3:36, 09.11.2020, РУС

# Открыть среду программирования thonny:

## Файл => Сохранить как





Оболочка x

```
Python 3.7.7 (bundled)
>>>
```

# Программирование на языке Python

Простейшие программы

# Простейшая программа

```
# Это пустая программа
```



Что делает эта программа?

комментарии после #  
не обрабатываются

кодировка utf-8  
по умолчанию)

```
# coding: utf-8
```

```
# Это пустая программа
```

```
"""
```

```
Это тоже комментарий
```

```
"""
```

Первая программа:

```
print("Hello, World!!")
```

Функция вывода:

```
print("текст")
```

Запуск программы:

Клавиша F5

Или в меню:

Выполнить =>

Запустить текущий  
скрипт



# Переменная и оператор присваивания

```
message = 'Hello, World!'  
print(message)
```

**message** – переменная  
**=** – оператор присваивания

**Переменная** – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

## Вывод на экран

```
► print ( "2+2=?" )  
► print ( "Ответ: 4" )
```

автоматический  
переход на новую  
строку

### Протокол:

2+2=?

Ответ: 4

```
print ( ' 2+2=? ' )  
print ( ' Ответ: 4 ' )
```

# Имена переменных

Имена переменных могут состоять из:

- **Латинские буквы** (строчные и заглавные буквы различаются!)
- **Русские буквы** (не рекомендуется)
- **Цифры** (имя не может начинаться с цифры и состоять только из цифр)
- **Знак подчеркивания** \_

Нельзя использовать в именах переменных:

- **Пробелы**
- **Знаки +, -, >, <, =, (), ! и др.**
- **Ключевые слова языка Python**

Нельзя использовать как имена переменных  
ключевые слова языка Python:

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	print
break	except	in	raise	

## Имена переменных

**МОЖНО** использовать

- латинские буквы (A-Z, a-z)

заглавные и строчные буквы **различаются**

- русские буквы (**не рекомендуется!**)
- цифры

имя не может начинаться с цифры

- знак подчеркивания \_

**НЕЛЬЗЯ** использовать

- ~~скобки~~
- ~~знаки +, =, !, ? и др.~~

Какие имена правильные?

**AXby R&B 4Wheel Вася “PesBarbos”**  
**TU154 [QuQu] \_ABBA A+B**

## Типы переменных

```
a = 4  
print ( type(a) )  
<class 'int'>
```

целое число (*integer*)

```
a = 4.5  
print ( type(a) )  
<class 'float'>
```

вещественное число

```
a = "Вася"  
print ( type(a) )  
<class 'str'>
```

символьная строка

```
a = True  
print ( type(a) )  
<class 'bool'>
```

логическая

## Зачем нужен тип переменной?

Тип определяет:

- область допустимых значений
- допустимые операции
- объём памяти
- формат хранения данных

# Математические операции

```
a = 78001457  
b = 2546880  
c = a + b  
print(c)
```

```
a = 78  
b = 25  
c = (a-b) * (a+b) / 27  
print(c)
```



## Другие математические операции:

$x + y$	Сложение
$x - y$	Вычитание
$x * y$	Умножение
$x / y$	Деление
$x // y$	Получение целой части от деления
$x \% y$	Остаток от деления
$-x$	Смена знака числа
$\text{abs}(x)$	Модуль числа
$\text{divmod}(x, y)$	Пара ( $x // y, x \% y$ )
$x ** y$	Возведение в степень

# Функция ввода

```
name = input("Введите своё имя: ")  
print("Привет, ", name)
```



Измените программу так, чтобы она выводила в конце восклицательный знак.

Ввод строки:

```
s = input("Введите строку: ")
```

"Введите строку: " – обращение к пользователю  
(не обязательно, но очень желательно)

## Ввод значения с клавиатуры

```
a = input ()
```

ввести строку с клавиатуры  
и связать с переменной `a`

```
b = input ()
```

```
c = a + b
```

```
print ( c )
```

Протокол:

21

33

2133



Почему?



Результат функции `input` – строка символов!

преобразовать в  
целое число

```
a = int ( input () )
```

```
b = int ( input () )
```

## Ввод с подсказкой

```
a = input ( "Введите число: " )
```

Введите число: 26

подсказка



Что не так?

```
a = int( input("Введите число: ") )
```

По умолчанию все введённые данные интерпретатор Питона понимает, как строки. Поэтому, если мы хотим получить число, то строку придётся преобразовать в число.

Ввод целого числа:

```
n = int(input("Введите число: "))
```

Функция преобразования к целочисленному типу:

```
n = int(s)
```

Функция преобразования к типу с плавающей точкой:

```
n = float(s)
```

Функция преобразования к строковому типу:

```
s = str(n)
```

## Вывод данных

```
print ( a )
```

значение  
переменной

```
print ( "Ответ: ", a )
```

значение и  
текст

перечисление через запятую

```
print ( "Ответ: ", a+b )
```

вычисление  
выражения

```
print ( a, "+", b, "=", c )
```

2 + 3 = 5

через пробелы

```
print ( a, "+", b, "=", c, sep = " " )
```

2+3=5

убрать разделители

## Сложение чисел: полное решение

```
print ( "Введите два числа: " )  
a = int ( input() )  
b = int ( input() )  
c = a + b  
print ( a, "+", b, "=", c, sep=" " )
```

подсказка

Протокол:

компьютер

Введите два целых числа

25

30

пользователь

25+30=55

## Вывод данных через `format`

```
print ( a, "+", b, "=", c, sep = "" )
```

2+3=5

```
print ( "{}+{}={} ".format(a, b, c) )
```





Задание. Напишите программу, которая получает на вход два числа и выводит их сумму:

```
a = input("Введите число a: ")  
b = input("Введите число b: ")  
sum = a+b  
print("a+b=", sum)
```

Почему программа работает неправильно?  
Что исправить в программе, чтобы она работала правильно?

# Программирование на языке Python

Вычисления

# Арифметическое выражения

$$a = (c + b^{**5*3} - 1) / 2 * d$$

**Приоритет** (старшинство):

- 1) скобки
- 2) возведение в степень \*\*
- 3) умножение и деление
- 4) сложение и вычитание

$$a = \frac{c + b^5 \cdot 3 - 1}{2} \cdot d$$

$$a = (c + b^{*5*3} - 1) \backslash$$

$$/ 2 * d$$

перенос на  
следующую строку

$$a = (c + b^{*5*3} - 1) / 2 * d$$

перенос внутри  
скобок разрешён

# Деление

Классическое деление:

```
a = 9; b = 6
x = 3 / 4    # = 0.75
x = a / b    # = 1.5
x = -3 / 4   # = -0.75
x = -a / b   # = -1.5
```

Целочисленное деление (округление «вниз»!):

```
a = 9; b = 6
x = 3 // 4    # = 0
x = a // b    # = 1
x = -3 // 4   # = -1
x = -a // b   # = -2
```

## Остаток от деления

% – ОСТАТОК ОТ ДЕЛЕНИЯ

```
d = 85
```

```
b = d // 10
```

```
a = d % 10
```

```
d = a % b
```

```
d = b % a
```

```
a = 15
```

```
b = 19
```

```
d = a // b
```

```
a = a % b
```

## Операторы `//` и `%`

```
a = 1234
```

```
d = a % 10; print( d )
```

4

```
a = a // 10
```

```
d = a % 10; print( d )
```

3

```
a = a // 10
```

```
d = a % 10; print( d )
```

2

```
a = a // 10
```

```
d = a % 10; print( d )
```

1

```
a = a // 10
```

## Сокращенная запись операций

$a += b$      $\#$      $a = a + b$

$a -= b$      $\#$      $a = a - b$

$a *= b$      $\#$      $a = a * b$

$a /= b$      $\#$      $a = a / b$

$a // = b$      $\#$      $a = a // b$

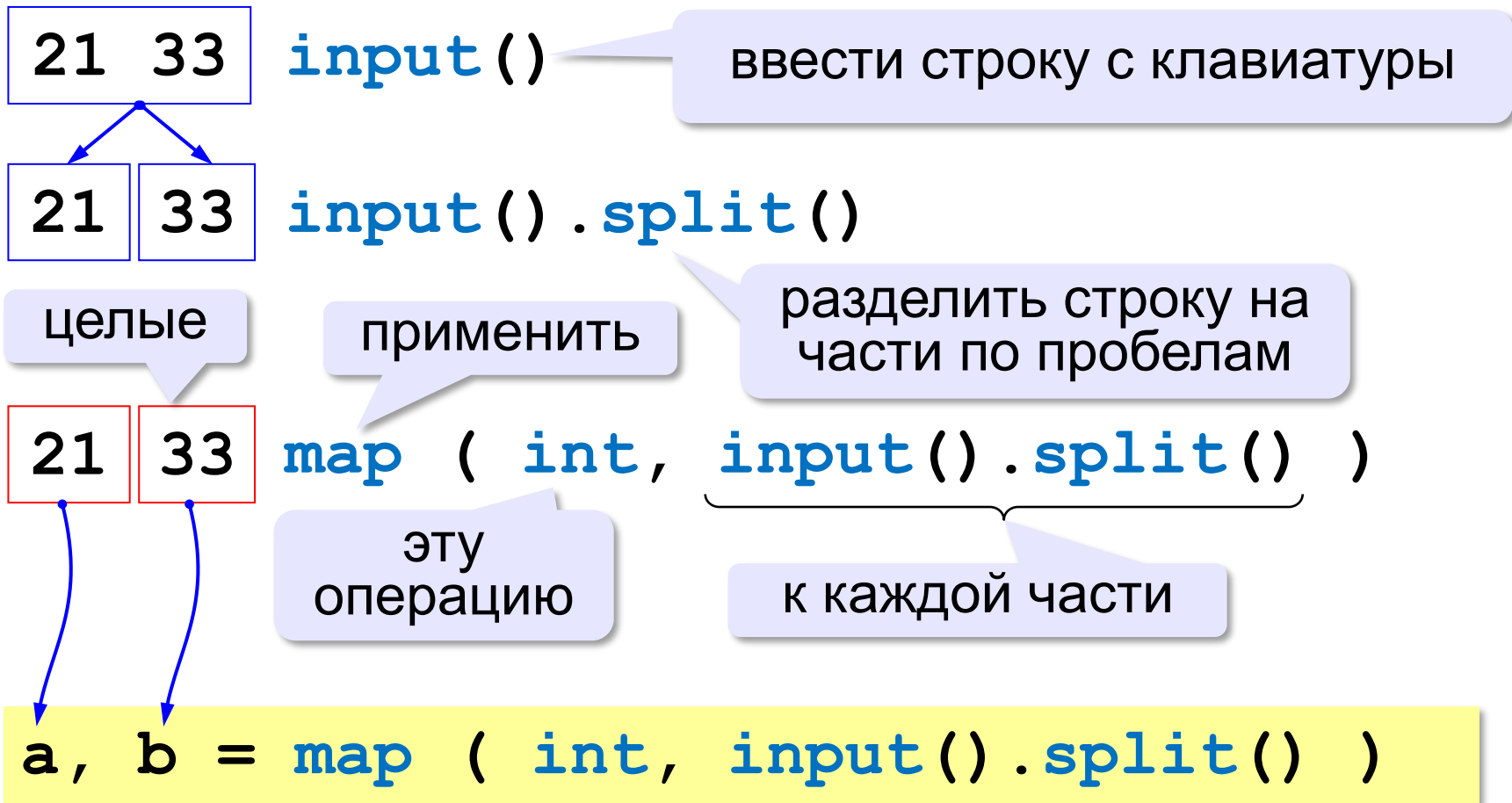
$a \% = b$      $\#$      $a = a \% b$

$a += 1$

увеличение на 1

## Ввод двух значений в одной строке

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







Задача. В каждой строке определить тип и значение переменной:

```
a = 5
n = input()      #человек вводит цифру 8
c = int(n)
d = a*c
d = d-a
s = "Рамамбахарумамбуру"
d = n+a
m = n+s
```

```
# Комментарии к программе, компьютер
# их не читает
```

# Генератор случайных чисел

Функция генерации случайного целого числа из отрезка  $[x, y]$ :

```
import random  
a = random.randint(x, y)
```

## Генератор случайных чисел

```
import random
```

англ. *random* – случайный

### Целые числа на отрезке [a,b]:

```
X = random.randint(1, 6) # псевдосл. число  
Y = random.randint(1, 6) # уже другое число!
```

### Генератор на [0,1):

```
X = random.random() # псевдосл. число  
Y = random.random() # уже другое число!
```

### Генератор на [a, b] (вещественные числа):

```
X = random.uniform(1.2, 3.5)  
Y = random.uniform(1.2, 3.5)
```

## Генератор случайных чисел

```
from random import *
```

подключить все!

англ. *random* – случайный

### Целые числа на отрезке [a,b]:

```
X = randint(10, 60) # псевдослучайное число  
Y = randint(10, 60) # это уже другое число!
```

### Генератор на [0,1):

```
X = random() # псевдослучайное число  
Y = random() # это уже другое число!
```



# Задания

- 1) Вывести на экран три введенных с клавиатуры числа в порядке, обратном их вводу.
- 2) Ввести с клавиатуры два числа и вывести целую часть от деления первого на второе.
- 3) Ввести с клавиатуры основание и высоту треугольника и вывести площадь треугольника.
- 4) Ввести с клавиатуры два катета и вывести гипотенузу.  
(Квадратный корень – это возведение в степень  $(1/2)$  )
- 5) Сгенерировать случайное двузначное число, вывести на экран это число, а также сумму и произведение его цифр.

Для получения цифр используйте целочисленное деление на 10 и взятие остатка от деления на 10.

Пример для числа 47:

$$47 // 10 = 4 \quad 47 \% 10 = 7$$