TARKVARA INSTALLEERIMINE JA PÕHISÜNTAKS

Python

Pythoni keele arendamist alustas 1980. aastate lõpus Hollandi instituudi CWI programmeerija Guido van Rossum

Python on oma nime saanud briti naljameeste telesarja “Monty Pythoni lendav tsirkus” järgi. (Monty Python’s Flying Circus)

Python on interpreteeritav programmeerimiskeel: lähtekood teisendatakse masinkoodiks interpretaatori täitmise ajal.

Pythonil on selge süntaks.

Selles programmeerimiskeeles on koodi lugemine üsna lihtne, sest sellel on vähe abielemente ja keelereeglid sunnivad programmeerijaid taandeid tegema.

Python – интерпретируемый язык программирования: исходный код частями преобразуется в машинный в процессе выполнения специальной программой — интерпретатором.

Python характеризуется ясным синтаксисом. Читать код на этом языке программирования достаточно легко, т.к. в нем мало вспомогательных элементов, а правила языка заставляют программистов делать отступы

https://www.python.org/

Welcome to Python.org - Google Chrome

python.org

Python | PSF | Docs | PyPI | Jobs | Community

python™

Donate | Search | GO | Socialize

About | Downloads | Documentation | Community | Success Stories | News | Events

All releases
Source code
Windows
Mac OS X
Other Platforms
License
Alternative Implementations

Download for Windows

Python 3.8.5

Note that Python 3.5+ *cannot* be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Download

Python source code and installers are available for download for all versions!

Latest: Python 3.8.5

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

docs.python.org

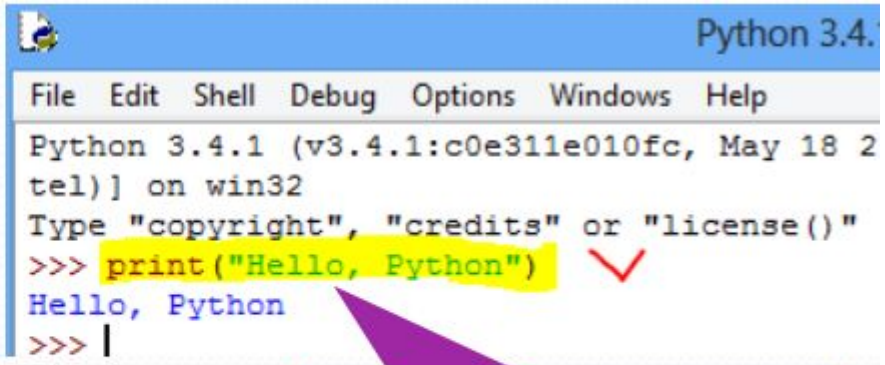
Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the

https://www.python.org/downloads/

EST 1:12 PM 9/1/2020

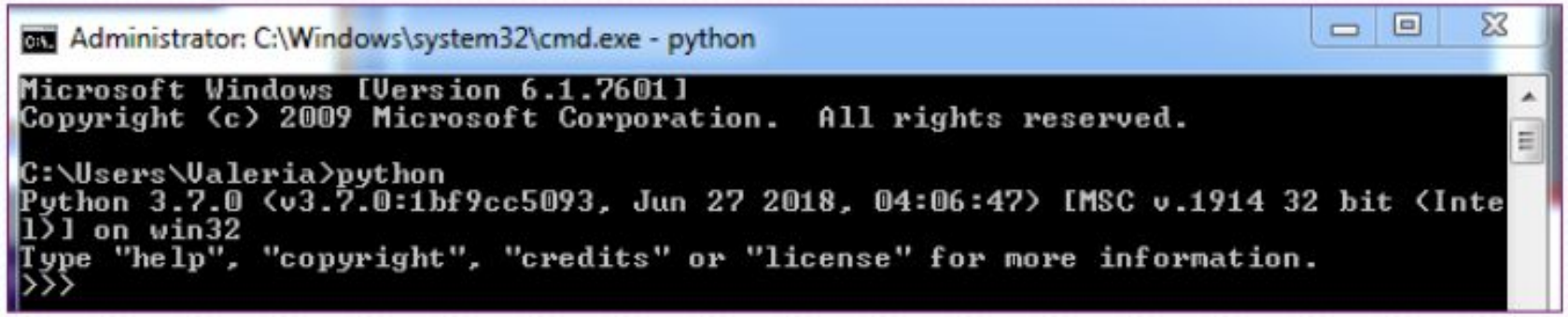
Käivitamine

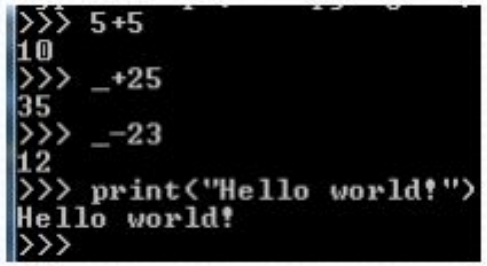


Süntaksi värvimine (käsud tõstetud esile kollasega)

Kuidas programme luua

- **Interaktiivne režiim**
- Interpretaator täidab käske ridade kaupa, st rida kirjutatakse -> interpretaator täidab selle -> tulemuse väljastus
- Käivitus ja kutse sisestamiseks (>>>)

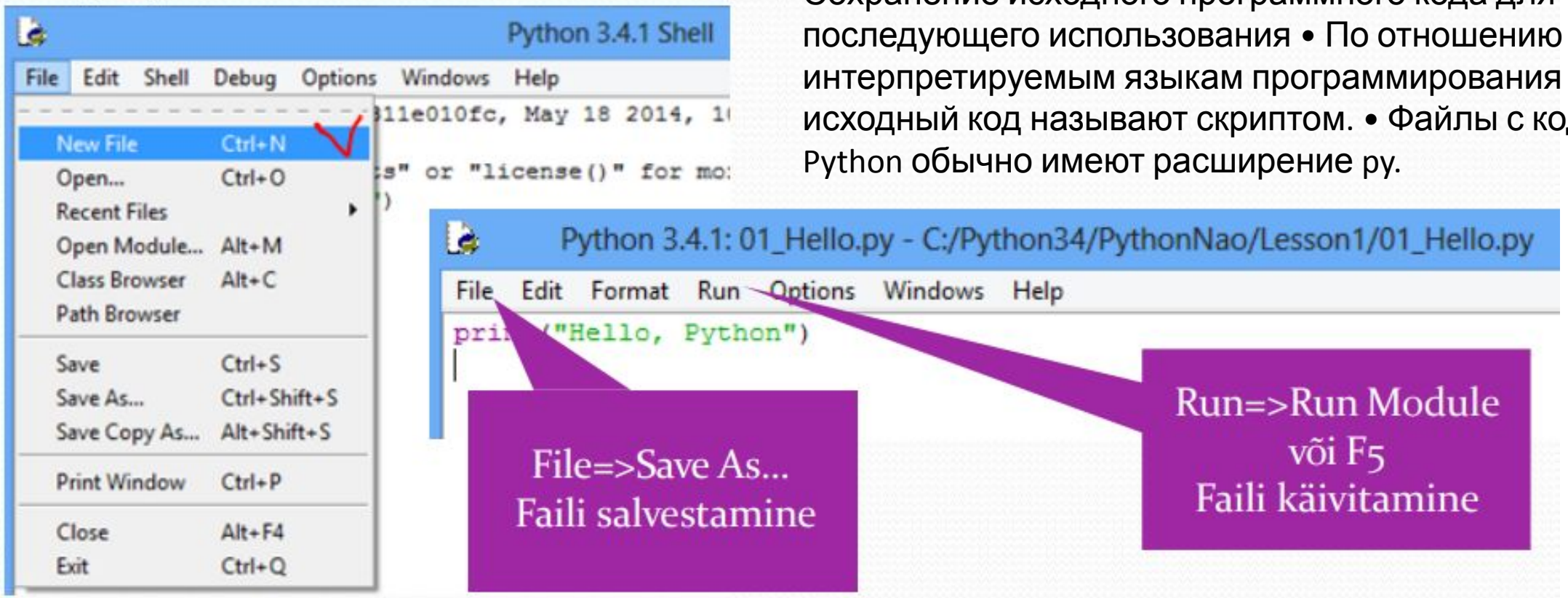




Интерактивный режим :интерпретатор выполняет команды построчно, т.е пишется строка -> интерпретатор выполняет ее -> вывод результата. Запуск и приглашение к вводу (>>>).

.py faili loomine

Сохранение исходного программного кода для последующего использования • По отношению к интерпретируемым языкам программирования часто исходный код называют скриптом. • Файлы с кодом на Python обычно имеют расширение py.



- Lähtekoodi salvestamine hilisemaks kasutamiseks
- Pythoni programmi sisaldavat tekstifaili nimetame edaspidi **skriptiks**.
- Pythoni failidel on tavaliselt **.py** laiend.

Põhisüntaks

Pythoni programm - käskudega tekstifail

Iga käsk eraldi reas

Kui käsk ei ole pesastatud, algab rea algusest (pesastatud käsu taane, näiteks 4 tühikut).

Valesti pandud taane on viga !!!

Käsu lõpus võite panna ; (kuid mitte soovitatav)

; pannakse, kui samas reas on mitu käsku (juhist)

Kui juhis on pikk, saab selle jagada, kasutades \

Программа на Python – текстовый файл с инструкциями

Каждая инструкция на отдельной строке

Если инструкция не вложенная начинается с начала строки (отступ вложенной инструкции, например, 4 пробела).

Неправильно поставленный отступ – ошибка!!!

В конце инструкции можно поставить ; (но не рекомендуется)

; ставится, если несколько инструкций в одной строке

Если инструкция длинная, то ее можно разбить с помощью \

Näide

```
import random

numberRandom=random.randint(1, 20)

numberUser=0;countAttempt=3
print("\n\nУгадай число от 1 до 20 за {0} попытки".format(countAttempt))
count=0
while numberRandom!=numberUser \
      and count<countAttempt:
    try:
        numberUser = int(input("Введите число: "))
        count+=1
        if numberUser<numberRandom:
            print("Число больше, попробуйте снова угадать.")
        if numberUser>numberRandom:
            print("Число меньше, попробуйте снова угадать.")
    except ValueError:
        print("Вы ввели не число...")
else:
    if (numberRandom==numberUser
        and count<=countAttempt):
        print("Молодец!!! Вы угадали, это число {0}.".format(numberRandom))
    else: print("Вы проиграли. Это было число {0}.".format(numberRandom))
```

Kommentaarid

Kommentaarid on programmiteksti osa selgitamise jaoks, mida kompileerimisel ignoreeritakse

Ainult üherealine kommentaar (paljudes programmeerimiskeeltes on olemas mitmerealine)

Однострочные комментарии

```
#see on kommentaar
print("Hello, world!") #väljund kuvarile
#print("Hello, world!")  see käsk programmi täitmisel ignoreeritakse
```

Dokumentatsiooni read - Interpretaator ei ignoreeri, vaid tajub seda tekstina

Многострочные комментарии

```
"""Selle osa interpretaator ei ignoreeri,
vaid tajub seda tekstina
print("Hello, world!") #väljund kuvarile
"""
```

```
'''Selle osa interpretaator ei ignoreeri,
vaid tajub seda tekstina
print("Hello, world!") #väljund kuvarile
'''
```

Väljund - näited

print() – - teabe kuvamine konsoolis

Sulgudes võivad olla mistahes tüüpi andmed.

Lisaks võib andmete hulk varieeruda.

Tekst pannakse jutumärkidesse.

print() – вывод информации на консоль В скобках могут быть любые типы данных. Кроме того, количество данных может быть различным Текст заключается в кавычки

```
1  print(1032)
2  print(2+2)
3  print(10 - 2.5/2)
4  print("Hello")
5  print("a:", 1)
6  print(1, 2, 3)
7  print("hello" + " world"*3)
8  print("hello\t world")
9  print("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun", sep="-")
10 print(1, 2, 3, sep="//", end="!!!!!")
```

Shell

```
1032
4
8.75
Hello
a: 1
1 2 3
hello world world world
hello    world
Mon-Tue-Wed-Thu-Fri-Sat-Sun
1//2//3!!!!!
```

Ülesanne

Задание

- Выведите на экран, используя соединение (+), повторение (*), специальные Escape символы:

\”	**Остается символ “**
\\	Остается символ \
\n	Перевод строки
\r	Возврат каретки
\t	Горизонтальная табуляция

- Текст:
  Корней Чуковский. "Федорино горе"
  И помчалися по улице ножи:
  "Эй, держи, держи, держи, держи, держи! “

- Добавьте комментарии

```
      Python Python Python Python
      Python Python Python Python
      Python Python Python Python
      Python Python Python Python
      Python Python Python Python
      Python Python Python Python
      Python Python Python Python
      Python Python Python Python

---------------
Корней Чуковский. "Федорино горе"

      ...
      И помчалися по улице ножи:
      "Эй, держи, держи, держи, держи, держи! "

      ...
```

Muutujad

ПЕРЕМЕННЫЕ

Muutuja

- Muutujal on nimi ja väärtus. Nime kasutatakse mäluala pääsemiseks, kus hoitakse väärtus. Muutuja hoiab kindlad andmed.
- Muutuja on kordumatu nimetus, mis koosneb ladina tähestiku tähtedest, numbritest ja allkriipsudest (Az, 0-9 ja _). Nimed EI TOHI sisaldada: vene tähed, tühikud, sulud, märgid +, =,!,? ja teised märgid.
  Muutuja nimi peab algama tähe või alakriipsuga, kuid mitte numbriga.
  Ei tohi kasutada märksõnu ja sisseehitatud käske (näiteks import, help...)
  Tõstutundlik - X ja x on erinevad muutujad

Переменная

- У переменной есть имя и значение. Имя служит для обращения к области памяти, в которой хранится значение. Переменная хранит определенные данные.
- Переменная – уникальное имя, состоящее из букв латинского алфавита, цифр и знака подчеркивания (Az, 0-9, и _). Имена НЕ могут включать: русские буквы, пробелы, скобки, знаки +, =, !, ? и др.
- Имя переменной должно начинаться с буквы или со знака подчеркивания, но <u>не с цифры</u>
- Нельзя использовать ключевые слова и встроенные команды (например, import, help…)
- Регистр букв учитывается - X и x разные переменные

Muutuja

Enne kasutamist tuleb kõik muutujad määratleda (defineerida) ja määratlemisel peab muutujale määrama algväärtuse, seda nimetatakse initsialiseerimiseks ehk lähtestamiseks.

Programmi täitmise ajal saab muutuja väärtust muuta.

Konstantideks nimetatakse muutumatuid väärtusi. Konstantide nimetamisel kasutatakse ainult suurtähti (näiteks PI, MAX_VALUE)

Перед использованием любая переменная должна быть определена и при определении нужно присвоить переменной начальное значение, это называется инициализацией.

Во время выполнения программы значение переменной можно изменять.

Константами называют неизменяемые величины. В именовании констант используются только заглавные буквы (например: PI, MAX_VALUE)

Näide

Python 3.7.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v
.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informa
tion.
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await',
'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except
', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try'
, 'while', 'with', 'yield']
>>> help(input)
Help on built-in function input in module builtins:

input(prompt=None, /)
    Read a string from standard input.  The trailing n
ipped.

    The prompt string, if given, is printed to stan
out a
    trailing newline before reading input.

    If the user hits EOF (*nix: C-D, Windows: Ctrl-Z+Return), ra
ise EOFError.
    On *nix systems, readline is used if available.

>>> help=12345
>>> help(input)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    help(input)
TypeError: 'int' object is not callable
>>>
```

Ключевые слова

Переопределение

1. Сначала функция help
2. Потом переменная
3. При вызове функции - **Ошибка**

Встроенные идентификаторы – фиолетовый цвет

Andmetüübid

Типы данных

- **bool** – булевский или логический тип True/False, 1/0
- **int** – целые числа от –147, 647
- **float** – вещественные числа. Примечание: разделителем целой и дробной части служит **точка**
  12.57 3.0
- **str** – строки Unicode (Unicode - стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков)
  Примечание: кавычки могут быть одинарными или двойными.
  “Hello world”
  ‘Hello “world” ’
- С помощью функции **type()** можно узнать текущий тип переменной

Tüübi teisendamine

Преобразование типов

- **int()** преобразует переданную ей строку (или число с плавающей точкой) в целое
- **str()** преобразует переданный ей аргумент в строку
- **float()** - в вещественное (дробное) число.

```
>>> int("100")
100
>>> int(12.99)
12
>>> int("1234 ")
1234
>>> int(" 12 34")

Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    int(" 12 34")
ValueError: invalid literal for int() with base 10: '12 34'
>>> float(1234)
1234.0
>>> str(54)
'54'
```

Operatsioonid

Операции

- **Операция** — это выполнение каких-нибудь действий над данными (**операндами**).

«+» по отношению к *числам* выполняет операцию сложения.
«+» по отношению к *строкам* — конкатенацию (соединение).

```
>>> 65+23
88
>>> "Hello"+" "+'world'
'Hello world'
>>> 25.12+35.17
60.290000000000006
>>> 12+2.5
14.5
```

```
>>> 15+" euro"

Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    15+" euro"
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Omistamine

Операция присваивания

- Связывание данных и имени (переменной происходит с помощью знака "=" (операция присваивания)
- Динамическая типизация - переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной (статическая).

```
number=10        #int
price=12.5       #float
isFound=True     #bool
text="Привет"    #str
print("number="+str(number))#str - преобразовать в строку
print(type(number)) #type - показать тип переменной
print(price)
print(isFound)
print(text);print(type(text))
print(int(price))

number="num 10" #тип изменился int->str
print(number);print(type(number))
```

результат

```
number=10
<type 'int'>
12.5
True
Привет
<type 'str'>
12
num 10
<type 'str'>
```

Matemaatika operaatorid

Математические операторы

Основные математические операторы:

+	сложение
-	вычитание
*	умножение
/	деление
%	остаток от деления
**	возведение в степень
//	целочисленное деление

```
>>> 100/3
33
>>> 100.0/3
33.333333333333336
>>> 25.5/5
5.1
>>> 17%2
1
>>> 23%10
3
>>> 2**8
256
>>> 12.3//4.1
3.0
>>> 12.3/5
2.46
>>> 12.3//5
2.0
>>> x=25;y=5
>>> result=x/y
>>> result
5
>>> x=x+1
>>> x
26
```

Tegevuste järjekord

Порядок выполнения операций

- вычисление выражений в скобках
- умножение, деление, //, % слева направо
- сложение и вычитание слева направо

2 3 5 4 1 7 8 6 9

```
z = (5*a*c+3*(c-d))/a*(b-c)/ b;
```

$$z = \frac{5ac + 3(c-d)}{ab}(b-c)$$

$$x = \frac{a^2 + 5c^2 - d(a+b)}{(c+d)(d-2a)}$$

2 6 3 4 7 5 1 12 8 11 10 9

```
x =(a*a+5*c*c-d*(a+b))/((c+d)*(d-2*a));
```

	Описание	Пример	Аналогичен
+=	добавляет значение правого оператора левому, и присваивает результат левому операнду	c += a	c = c + a
-=	вычитает правый оператор из левого	c -= a	c = c – a
*=	умножает правый оператор на левый	c *= a	c = c * a
/=	делит левый оператор на правый	c /= a	c = c / a
%=	получает остаток от деления	c %= a	c = c % a
**=	выполняет возведение в степень а левого операнда	c **= a	c = c ** a
//=	выполняет целочисленное деление левого операнда	c //= a	c = c // a

Ülesanne

1. Переменной numberInt присвойте значение 10, numberFloat - 8.4, text - «No».

2. Измените значение хранимое в переменной numberInt, увеличив его в 3.5 раза, результат сохраните в переменную numberNotInt.

3. Измените значение переменной numberFloat, уменьшив его на единицу, результат сохраните в той же переменной.

4. Разделите numberInt на numberFloat. Результат не нужно сохранять

5. Измените значение переменной text на «NoNo_YesYesYes». При формировании нового значения используйте операции конкатенации (+) и повторения строки (*).

6. Выведите значения всех существующих переменных

Andmete sisestamine

Ввод данных

- **input()** отвечает за ввод информации. В качестве параметра принимает приглашение к вводу и возвращает введенную строку.

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

→

```
Введите имя: Катя
Привет,  Катя
```

- **input()** ожидает ввода пользователем своего имени. Введенное имя попадает в переменную ***name***.
- **print()** – выводит приветствие на консоль

Ввод данных

```
print("Программа вычисления площади и периметра прямоугольника")

width=int(input("Введите ширину: "));
print(type(width))
height=int(input("Введите длину: "));
print(type(height))

area=width*height
perimetr=2*(width+height)

print("Площадь равна: "+str(area))
print("Периметр равен: "+str(perimetr))
```

```
>>>
Программа вычисления площади и периметра прямоугольника
Введите ширину: 12
<type 'int'>
Введите длину: 10
<type 'int'>
Площадь равна: 120
Периметр равен: 44
>>>
```

Ülesanne

Ввести три числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три числа:

4 5 7

Программа выводит информацию на экран пользователя

4+5+7=16

4*5*7=140

(4+5+7)/3=5. 33

Ülesanne

1. Даны две переменные. Запросить их значение. Выполнить основные арифметические действия с переменными, целочисленное деление, возведение в квадрат. Вывести результат.

2. Пользователь вводит сторону квадрата side. Найти его периметр p= 4▪side.

3. Пользователь вводит стороны прямоугольника a и b. Найти его площадь square = a▪b и периметр perimetr = 2▪(a + b)

4. Найти длину окружности L и площадь круга S заданного радиуса R: L = 2▪π▪R, S = π▪R^2. В качестве значения π использовать константу (3.14)

5. Пользователь вводит цены 3-х товаров. Найти сумму покупки и среднюю цену товаров. Осуществите вывод результатов с 2 цифрами после запятой в денежном формате.

```
number=5.33333333
print(round(number,2))
```

При решении задач комментируйте код, вывод информации должен быть понятным.

https://metanit.com/python/tutorial/5.3.php

```
euro=15
print(euro, u'\u20AC' ) #  15 €
print(euro,u'\u0024') #  15 $
```

Küsimused

???????