## Wakari, Trinket и IPython Notebook

Для выполнения задач по курсу необязательно иметь установленный Python на своем компьютере, так как программы проверяются на сервере Stepik, однако это желательно для дальнейшей работы.  
  
При записи видео в курсе использовалась онлайн-платформа [Wakari](https://wakari.io/), сейчас она не работает.  
  
Вы можете при прохождении курса:  
1. пользоваться платформой [Trinket.io](https://trinket.io/python3).  
2. установить себе на компьютер IPython Notebook.

О том, как это сделать, мы расскажем в следующих уроках

**Установка интерпретатора python3**

Мы рекомендуем установить последнюю на текущий момент версию языка python3 - версию python 3.5

Чтобы загрузить установщики для операционных систем Windows и Mac OS, достаточно перейти по ссылке <https://www.python.org/downloads/release/python-351/> ﻿и в самом низу страницы ﻿выбрать ﻿нужный для вашей архитектуры установщик интерпретатора: для windows 64bit - "﻿Windows x86-64 executable installer", для windows 32bit - "Windows x86 executable installer﻿", для Mac OS ﻿64 bit ﻿- ﻿"Mac OS X 64-bit/32-bit installer", ﻿ для Mac OS 32bit/PPC - ﻿﻿"Mac OS X 32-bit i386/PPC installer".

Для того, чтобы установить python3 на операционных системах linux, необходимо ﻿использовать стандартные ﻿менеджеры пакетов﻿ ﻿(apt-get или yum) вашей операционной системы. Например, чтобы установить python3 на операционной системе ubuntu необходимо запустить в терминале следующую команду:

sudo apt-get install python3

Чтобы убедиться в том, что вы установили интерпретатор, необходимо запустить интерпретатор в терминале с помощью команды.

﻿python3

## Общие рекомендации по онлайн-курсу

Чтобы все функции видео-плеера (например, ускорение видео) работали корректно, используйте одну из последних версий браузера. Подробнее о поддерживаемых версиях: <http://caniuse.com/#search=mp4>

Несмотря на то, что в курсе есть видео и текстовые материалы, ведите конспект или хотя бы делайте заметки. Так материал будет лучше запоминаться.

При необходимости ставьте видео на паузу. Помните, что в онлайн-курсе материал даётся в более сжатом виде, чем на обычных лекциях. И наоборот: ускоряйте видео, если изложение кажется вам слишком медленным.

Под каждым шагом есть форум для обсуждения. В нём можно (и нужно!) задавать вопросы по материалам и задачам курса, а также помогать другим, если знаете ответ. **Пожалуйста, не выкладывайте свой код и решения задач**. Старайтесь разобраться, в чем проблема, по наводящим вопросам. Формат курса не подразумевает поиск преподавателями ошибок в индивидуальных программах студентов.

При возникновении технических вопросов по курсу пишите на почту info@bioinf.me.

## Удачного прохождения курса!

Ответы на основные вопросы по платформе вы можете найти [в справочном центре](https://support.stepik.org/).

Присоединяйтесь к нам в соцсетях, чтобы узнать последние новости о платформе и курсах: [ВКонтакте](https://vk.com/rustepik), [Facebook](https://www.facebook.com/rustepik), а также новостям Института биоинформатики: [https://vk.com/bioinfВКонтакте](https://vk.com/bioinf).

Рекомендуйте курс друзьям и знакомым, ведь вместе проходить его гораздо веселее!

Весь код на Python, рассматриваемый в рамках этой недели, опубликован по этой ссылке  
<https://stepic.org/media/attachments/course67/week_1.html> ﻿

Официальный сайт Python, где можно скачать интерпретатор (Python 3): <https://www.python.org/>

Официальная документация по Python: <https://docs.python.org/3/>

Веб-сервис, позволяющий исполнять программы на Python прямо в вашем браузере: <https://trinket.io/python/41462f0f16>

Среда для написания программ PyCharm Educational Edition или PyCharm Community Edition:  
<https://www.jetbrains.com/pycharm-educational/>  
<https://www.jetbrains.com/pycharm/>

Текстовый редактор с подсветкой синтаксиса программ Sublime Text 3: <http://www.sublimetext.com/3>

Интерактивный учебник языка Python (на русском языке): <http://pythontutor.ru/>

Книга A Byte of Python: <http://www.swaroopch.com/notes/python/>  
и ее перевод на русский: <http://wombat.org.ua/AByteOfPython/>

Мы советуем при прохождении курса:  
  
1. Пользоваться онлайн-платформой [Trinket.io](https://trinket.io/python3). которая позволяет запускать программы на Python в вашем браузере. Обратите внимание, что в Trinket используется не IPython, а просто Python. Поэтому для того, чтобы увидеть значение выражения, необходимо явно вызвать функцию print(), иначе оно не будет отображаться. Например, вместо 42 + 8 \* 3 нужно написать print(42 + 8 \* 3) и нажать кнопку "Run".  
2. Установить себе на компьютер IPython Notebook (сейчас приложение называется [Jupyter Notebook](https://jupyter.org/)).

Cсылка для скачивания пакета Miniconda: <http://conda.pydata.org/miniconda.html>

Установка IPython (для Windows):  
[https://stepic.org/media/attachments/course67/ipython\_install.cmd](https://stepik.org/media/attachments/course67/ipython_install.cmd)

Установка IPython (для OS X и Linux):  
[https://stepic.org/media/attachments/course67/ipython\_install.sh](https://stepik.org/media/attachments/course67/ipython_install.sh)

Команды для запуска:  
conda update conda --yes  
conda install ipython --yes  
conda install ipython-notebook --yes  
conda install pyzmq --yes  
conda install jinja2 --yes  
conda install tornado --yes

В этом уроке мы начнем пользоваться IPython Notebook.

Для вашего удобства получившийся при демонстрации notebook мы выложили в открытый доступ по следующей ссылке:  
<https://stepic.org/media/attachments/course67/week_1.html>﻿

Для лучшей читаемости текста в видео, рекомендуем поставить максимальное качество видео﻿ (число в правой нижней части видео-плеера) ﻿и развернуть плеер на полный экран (кнопка в правой верхней части плеера). ﻿

Код Python, с которым мы будем работать на дальнейших шагах урока, доступен по ссылке:

[https://stepic.org/media/attachments/course67/week\_1.html#Переменные](https://stepic.org/media/attachments/course67/week_1.html#%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5) ﻿

Код Python, используемый на дальнейших шагах, опубликован для ознакомления по ссылке:

[https://stepic.org/media/attachments/course67/week\_1.html#Логические-операции](https://stepic.org/media/attachments/course67/week_1.html#%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8) ﻿

1.10 Условия: if, else, elif. Блоки, отступы

Код Python, используемый далее, опубликован для ознакомления по ссылке:

[https://stepic.org/media/attachments/course67/week\_1.html#Условия](https://stepic.org/media/attachments/course67/week_1.html#%D0%A3%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F) ﻿

Требуется определить, является ли данный год високосным.

Напомним, что високосными годами считаются те годы, порядковый номер которых либо кратен 4, но при этом не кратен 100, либо кратен 400 (например, 2000-й год являлся високосным, а 2100-й будет невисокосным годом).

Программа должна корректно работать на числах 1900≤n≤3000.

Выведите "Високосный" в случае, если считанный год является високосным и "Обычный" в обратном случае (не забывайте проверять регистр выводимых программой символов).

year = int(input())

if year % 400 == 0:

print('Високосный')

elif year % 4 == 0 and year % 100 != 0:

print('Високосный')

else:

print('Обычный')

вариант 2

a = int(input())

if a % 4 == 0 and a % 100 != 0 or a % 400 == 0: print("Високосный")

else: print("Обычный")

вариант 3

y = int(input())

print('Високосный' if not y%400 or not y%4 and y%100 else 'Обычный')

вариант 4

import calendar

year = int(input())

if calendar.isleap(year):

print ("Високосный")

else: print ("Обычный")

вариант 5

y=int(input())

if 1900<=y<=3000:

if y%400==0 or y%4==0 and y%100!=0:

print("Високосный")

else: print("Обычный")

перевод строки \n

print('First', '\n', 'Second')

результат

First

Second

Код Python, используемый далее, опубликован для ознакомления по ссылке:

[https://stepic.org/media/attachments/course67/week\_1.html#Строки](https://stepic.org/media/attachments/course67/week_1.html#%D0%A1%D1%82%D1%80%D0%BE%D0%BA%D0%B8)  ﻿

В то далёкое время, когда Паша ходил в школу, ему очень не нравилась формула Герона для вычисления площади треугольника, так как казалась слишком сложной. В один прекрасный момент Павел решил избавить всех школьников от страданий и написать и распространить по школам программу, вычисляющую площадь треугольника по трём сторонам.

Одна проблема: так как эта формула не нравилась Павлу, он её не запомнил. Помогите ему завершить доброе дело и напишите программу, вычисляющую площадь треугольника по переданным длинам трёх его сторон по формуле Герона:

S=\sqrt{p(p−a)(p−b)(p−c)}*S*=*p*(*p*−*a*)(*p*−*b*)(*p*−*c*)​

где p=\dfrac{a+b+c}2*p*=2*a*+*b*+*c*​ – полупериметр треугольника. На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника.

**Sample Input:**

3

4

5

**Sample Output:**

6.0

a = int(input())

b = int(input())

c = int(input())

p = (a + b + c)/2

print((p\*(p-a)\*(p-b)\*(p-c))\*\*0.5)

Напишите программу, принимающую на вход целое число, которая выводит True, если переданное значение попадает в интервал (-15, 12] \cup (14, 17) \cup [19, +\infty)(−15,12]∪(14,17)∪[19,+∞) и False в противном случае (регистр символов имеет значение).

Обратите внимание на разные скобки, используемые для обозначения интервалов. В задании используются полуоткрытые и открытые интервалы. Подробнее про это вы можете прочитать, например, на википедии ([полуинтервал](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%83%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%B2%D0%B0%D0%BB), [промежуток](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BC%D0%B5%D0%B6%D1%83%D1%82%D0%BE%D0%BA_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)#.D0.A2.D0.B5.D1.80.D0.BC.D0.B8.D0.BD.D0.BE.D0.BB.D0.BE.D0.B3.D0.B8.D1.8F)).

a = int(input())

if (-15) < a <= 12 or 14 < a < 17 or a >= 19:

print(True)

else:

print(False)

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты.  
Для числа π в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

треугольник

a

b

c

где a, b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

figure = input()

if figure == 'треугольник':

a = int(input())

b = int(input())

c = int(input())

p = 0

p = (a + b + c)/2

print((p\*(p-a)\*(p-b)\*(p-c))\*\*0.5)

elif figure == 'прямоугольник':

a = int(input())

b = int(input())

print(float(a \* b))

elif figure == 'круг':

r = int(input())

print(3.14 \* (r \*\* 2))

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

**Sample Input 1:**

8

2

14

**Sample Output 1:**

14

2

8

a = int(input())

b = int(input())

c = int(input())

max\_n = a

min\_n = b

rest\_n = c

if b >= a and b >= c:

max\_n = b

if c >= a:

rest\_n = c

min\_n = a

else:

rest\_n = a

min\_n = c

elif c >= a and c >= b:

max\_n = c

if a <= b:

min\_n = a

rest\_n = b

else:

rest\_n = a

min\_n = b

elif a >= c and a >= b:

max\_n = a

if b <= c:

min\_n = b

rest\_n = c

else:

rest\_n = b

min\_n = c

print(max\_n)

print(min\_n)

print(rest\_n)

вариант 2

a,b,c=int(input()),int(input()),int(input())

maxim=a

minim=b

if b>a:

maxim=b

minim=a

if c>maxim:

maxim=c

if c<minim:

minim=c

ost=a+b+c-minim-maxim

print(maxim)

print(minim)

print(ost)

вариант 3

Верное решение #29284556

Python 3

a,b,c = int(input()), int(input()), int(input())

if a < b:

a, b = b, a

if a < c:

a, c = c, a

if b > c:

b, c = c, b

print (a,c,b,sep='\n')

вариант 4

Верное решение #28751556

Python 3

# put your python code here

a, b, c = int(input()), int(input()), int(input())

max\_int = max(a, b, c)

min\_int = min(a, b, c)

print(max\_int)

print(min\_int)

print((a + b + c) - max\_int - min\_int)

вариант 5

Верное решение #109609906

Python 3

# put your python code here

a, b, c = int(input()), int(input()), int(input())

max = a

min = a

if b > max: max = b

if c > max: max = c

if b < min: min = b

if c < min: min = c

mid = (a+b+c)-(max+min)

print(str(max)+'\n'+str(min)+'\n'+str(mid))

вариант 6

Верное решение #18391294

Python 3

a=int(input())

b=int(input())

c=int(input())

a, b = (a, b) if a >= b else (b, a)

b, c = (b, c) if b >= c else (c, b)

a, b = (a, b) if a >= b else (b, a)

print(a)

print(c)

print(b)

вариант 7

Верное решение #23755458

Python 3

x=sorted([int(input()),int(input()),int(input())])

print (x[2], x[0], x[1], sep="\n")

В институте биоинформатики по офису передвигается робот. Недавно студенты из группы программистов написали для него программу, по которой робот, когда заходит в комнату, считает количество программистов в ней и произносит его вслух: "n программистов".

Для того, чтобы это звучало правильно, для каждого n*n* нужно использовать верное окончание слова.

Напишите программу, считывающую с пользовательского ввода целое число n*n* (неотрицательное), выводящее это число в консоль вместе с правильным образом изменённым словом "программист", для того, чтобы робот мог нормально общаться с людьми, например: 1 программист, 2 программиста, 5 программистов.

В комнате может быть очень много программистов. Проверьте, что ваша программа правильно обработает все случаи, как минимум до 1000 человек.

**Дополнительный комментарий к условию:**  
Обратите внимание, что задача не так проста, как кажется на первый взгляд. **Если** ваше **решение не проходит** какой-то **тест**, это значит, что **вы не рассмотрели какой-то из случаев** входных данных (число программистов 0 \le n \le 10000≤*n*≤1000). Обязательно проверяйте свои решения на дополнительных значениях, а не только на тех, что приведены в условии задания.

Так как **задание повышенной сложности**, **вручную код решений проверяться не будет**. Если вы столкнулись с ошибкой в первых четырёх тестах, проверьте, что вы используете только русские символы для ответа. В остальных случаях ищите ошибку в логике работы программы.

**Sample Input 1:**

5

**Sample Output 1:**

5 программистов

n = int(input())

if n >= 0:

n = str(n)

if n[-1] == '1':

print(n, 'программист')

elif n[-1] == '2' or n[-1] == '3' or n[-1] == '4':

print(n, 'программиста')

elif n[-1] == '0' or n[-1] == '5' or n[-1] == '6' or n[-1] == '7' or n[-1] == '8' or n[-1] == '9':

print(int(n), 'программистов')

ВАРИАНТ 2

s = int(input())

n1, n2, n3 =" программистов", " программист", " программиста"

if s >= 0:

if s == 0:

print(str(s) + n1)

elif s % 100 >= 10 and s % 100 <= 20:

print(str(s) + n1)

elif s % 10 == 1:

print(str(s) + n2)

elif s % 10 >= 2 and s %10 <= 4:

print(str(s) + n3)

else:

print(str(s) + n1)

ВАРИАНТ 3

Верное решение #18393966

Python 3

i=int(input())

d=i%10

h=i%100

if d==1 and h!=11:

s=""

elif 1<d<5 and not 11<h<15:

s="а"

else:

s="ов"

print(i," программист"+s)

Верное решение #87176988

Python 3

a=int(input())

if 2<=a%10<=4 and not 12<=a%100<=14:

b='программиста'

elif a%10 ==1 and a%100!=11:

b='программист'

else:

b='программистов'

print(str(a), b)

Верное решение #29294037

Python 3

n=int(input())

print(n,'программист'+('ов' if n%10==0 or 4<n%10<10 or 10<n%100<15 else 'а' if 1<n%10<5 else ''))

Верное решение #9390994

Python 3

a = int(input())

n = a%10

x = a%100

if 11 <= x <= 14:

print(a, "программистов")

else:

if 2 <= n <= 4:

print(a, "программиста")

elif (5 <= n <= 9) or (n == 0):

print(a, "программистов")

elif n == 1:

print(a, "программист")

Верное решение #135170190

Python 3

t = 'программист'

i = int(input()); n = i % 100; m = n % 10

if m == 0 or 10 < n < 20 or 4 < m < 10: t += 'ов'

elif 1 < m < 5: t += 'а'

print(i, t)

Верное решение #36193896

Python 3

dic, n = {'1': '', '2': 'а', '3': 'а', '4': 'а'}, input()

print(n, 'программист'+(dic.get(n[-1], 'ов') if not 10 < int(n[-2::]) < 20 else 'ов'))

Верное решение #101485545

Python 3

n = int(input())

k = n % 10

print(n, 'программист' + (('а', '')[k == 1], 'ов')[n % 100 // 10 == 1 or not(0 < k < 5)])

Верное решение #122440417

Python 3

# put your python code here

n = int(input())

end = ''

if 11 <= n % 100 <= 14 or 5 <= n % 10 <= 9 or n % 10 == 0:

end = 'ов'

elif 1 < n % 10 < 5:

end = 'а'

print(n, 'программист'+end)

[https://open-vsx.org/extension/humao/rest-client](https://st1.zoom.us/web_client/9zdhk1t/html/externalLinkPage.html?ref=https://open-vsx.org/extension/humao/rest-client)

расширение для тестирования

postman.com

<?xml version="1.0"?>

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.  
  
Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.  
  
На вход программе подаётся строка из шести цифр.

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

**Sample Input 1:**

090234

**Sample Output 1:**

Счастливый

**Sample Input 2:**

123456

**Sample Output 2:**

Обычный

# put your python code here

n = input()

s1 = int(n[0])+int(n[1])+int(n[2])

s2 = int(n[3])+int(n[4])+int(n[5])

if s1==s2:

print('Счастливый')

else:

print('Обычный')

Верное решение #28832839

Python 3

code=int(input())

a=code//100000

b=code//10000%10

c=code//1000%10

d=code//100%10

e=code//10%10

f=code%10

if a+b+c==d+e+f:

print('Счастливый')

else:

print('Обычный')

Верное решение #22385178

Python 3

n = list(map(int, list(input())))

print('Счастливый' if sum(n[:3]) == sum(n[3:]) else 'Обычный')

Верное решение #34918907

Python 3

a,b,c,d,e,f=(int(n) for n in input())

print(('Обычный','Счастливый')[a+b+c == d+e+f])

Приветствуем вас!

За вторую неделю курса мы познакомимся с различными видами циклов в Python и с двумя типами данных — строками и списками.

Весь код на Python, который мы будем использовать в этой неделе, опубликован по этой ссылке  
<https://stepic.org/media/attachments/course67/week_2.html> ﻿

[Текстовый конспект лекций этого модуля](https://wiki.stepik.org/index.php/Category:%D0%A6%D0%B8%D0%BA%D0%BB%D1%8B._%D0%A1%D1%82%D1%80%D0%BE%D0%BA%D0%B8._%D0%A1%D0%BF%D0%B8%D1%81%D0%BA%D0%B8_(M-148))

**Цикл while**

Пример

a = 5

while a > 0:

print(a, end= ' ')

a -= 1

вывод 5 4 3 2 1

a = 5

while a <= 55:

print(a, end=' ')

a += 2

вывод 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55

a = 5

while a <= 55:

if a % 2 ==1:

print(a)

a += 1

вывод

5

7

9

…

55

Код для задач, которые разбираются в следующей части урока, доступен для просмотра по ссылке:

[https://stepic.org/media/attachments/course67/week\_2.html#Вывести-треугольник-из-звезд](https://stepic.org/media/attachments/course67/week_2.html#%D0%92%D1%8B%D0%B2%D0%B5%D1%81%D1%82%D0%B8-%D1%82%D1%80%D0%B5%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA-%D0%B8%D0%B7-%D0%B7%D0%B2%D0%B5%D0%B7%D0%B4)  ﻿

Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

**Sample Input 1:**

5

-3

8

4

0

**Sample Output 1:**

14

**# put your python code here**

**sum\_a = 0**

**a = int(input())**

**while a != 0:**

**sum\_a += a**

**a = int(input())**

**print(sum\_a)**

Верное решение #22401922

Python 3

s, n =0, int(input())

while n:

s, n = s + n, int(input())

print(s)

Верное решение #55523151

Python 3

summa = 0

i = True

while i:

i = int(input())

summa += i

print(summa)

В Институте биоинформатики между информатиками и биологами устраивается соревнование. Победителям соревнования достанется большой и вкусный пирог. В команде биологов a*a* человек, а в команде информатиков — b*b* человек.

Нужно заранее разрезать пирог таким образом, чтобы можно было раздать кусочки пирога любой команде, выигравшей соревнование, при этом каждому участнику этой команды должно достаться одинаковое число кусочков пирога. И так как не хочется резать пирог на слишком мелкие кусочки, нужно найти минимальное подходящее число.

Напишите программу, которая помогает найти это число.  
Программа должна считывать размеры команд (два положительных целых числа a*a* и b*b*, каждое число вводится на отдельной строке) и выводить наименьшее число d*d*, которое делится на оба этих числа без остатка.

**Sample Input 1:**

1

2

**Sample Output 1:**

2

**Sample Input 2:**

7

5

**Sample Output 2:**

35

**a, b = int(input()), int(input())**

**i = min(a, b)**

**while True:**

**if i % a == 0 and i % b == 0:**

**break**

**i += 1**

**print(i)**

Верное решение #21511745

Python 3

a = int(input())

b = int(input())

d = a

while d%b:

d += a

print(d)

Верное решение #145331102

Python 3

a = int(input())

b = int(input())

m = a

if b > a:

m = b

while m % a != m % b:

m = m + 1

print(m)

Верное решение #314019799

Python 3

a, b = int(input()), int(input())

s = a

while s % a or s % b:

s += a

print(s)

Верное решение #118142278

Python 3

# put your python code here

import math

A=int(input())

B=int(input())

print (A\*B//math.gcd(A,B))

Верное решение #50361200

Python 3

a=int(input())

b=int(input())

d=1

while (d%a+d%b)!=0:

d+=1

print (d)

Верное решение #149628909

Python 3

from math import gcd

a, b = int(input()), int(input())

print(int(abs(a \* b) / gcd(a, b)))

**i = 0**

**while i < 5:**

**a, b = input().split()**

**a = int(a)**

**b = int(b)**

**if (a == 0) and (b == 0):**

**break**

**if (a == 0) or (b == 0):**

**continue**

**print(a \* b)**

**i += 1**

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

i = 0

s = 0

while i < 10:

i = i + 1

s = s + i

if s > 15:

break

i = i + 1

**пример использования опретора BREAK**

**i = 0**

**while i < 5:**

**a, b = input().split()**

**a = int(a)**

**b = int(b)**

**if (a == 0) and (b == 0):**

**break**

**print(a \* b)**

**i += 1**

**вывод**

**4 5**

**20**

**Опретор CONTINUE**

**i = 0**

**while i < 5:**

**a, b = input().split()**

**a = int(a)**

**b = int(b)**

**if (a == 0) and (b == 0):**

**break**

**if (a == 0) or (b == 0):**

**continue**

**print(a \* b)**

**i += 1**

**цикл FOR**

**пример**

**for I in 2, 3, 5:**

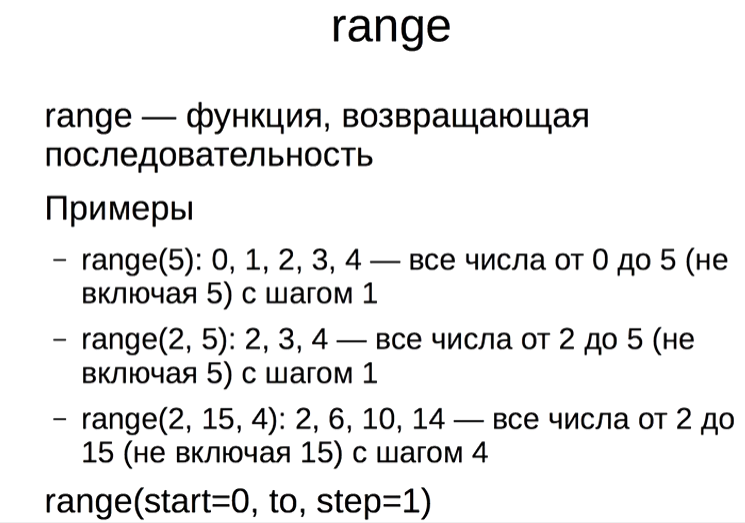
**print(I + i)**

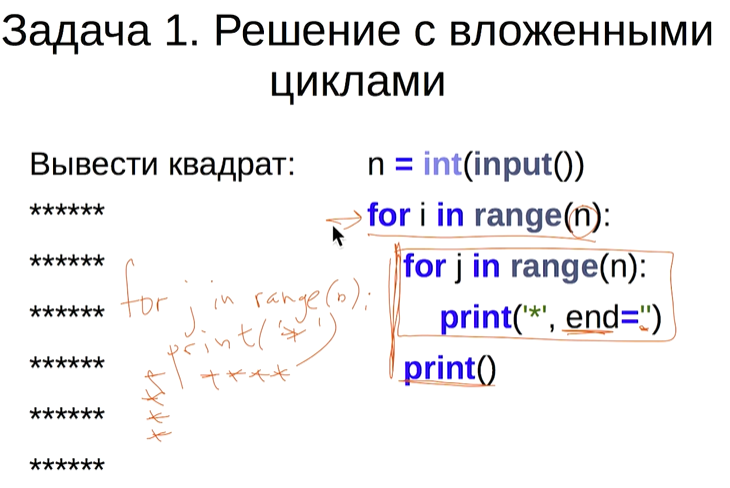
**вывод**

**4**

**9**

**25**

****

****

Когда Павел учился в школе, он запоминал таблицу умножения прямоугольными блоками. Для тренировок ему бы очень пригодилась программа, которая показывала бы блок таблицы умножения.

Напишите программу, на вход которой даются четыре числа a*a*, b*b*, c*c* и d*d*, каждое в своей строке. Программа должна вывести фрагмент таблицы умножения для всех чисел отрезка [a; b][*a*;*b*] на все числа отрезка [c;d][*c*;*d*].

Числа a*a*, b*b*, c*c* и d*d* являются натуральными и не превосходят 10, a \le b*a*≤*b*, c \le d*c*≤*d*.

Следуйте формату вывода из примера, для разделения элементов внутри строки используйте '\t' — символ табуляции. Заметьте, что левым столбцом и верхней строкой выводятся **сами числа из заданных отрезков** — заголовочные столбец и строка таблицы.﻿

**Sample Input 1:**

7

10

5

6

**Sample Output 1:**

5 6

7 35 42

8 40 48

9 45 54

10 50 60

**a, b, c, d = int(input()), int(input()), int(input()), int(input())**

**for g in range (c, d + 1):**

**print('\t'+str(g),end='')**

**print(end='\n')**

**for i in range (a, b + 1):**

**print(str(i) + '\t', end='')**

**for j in range (c, d + 1):**

**print(str(i \* j), end='\t')**

**print(end='\n')**

Верное решение #32197976

Python 3

a=int(input())

b=int(input())

c=int(input())

d=int(input())

print('\t', \*range(c, d+1), sep='\t')

for i in range(a,b+1):

print(i, \*range(i\*c,(i\*d)+1, i), sep='\t')

Верное решение #100627482

Python 3

a, b, c, d = int(input()), int(input()), int(input()), int(input())

for i in range(c, d + 1):

print("\t" + str(i), end="")

print()

for i in range(a, b + 1):

print(i, end="\t")

for n in range(c, d + 1):

print(i \* n, end="\t")

print()

Верное решение #20487758

Python 3

a, b, c, d = (int(input()) for x in range(4))

print('', \*range(c,d+1), sep='\t')

for x in range(a, b+1):

print(x, \*[y\*x for y in range(c, d+1)], sep='\t')

Верное решение #32198061

Python 3

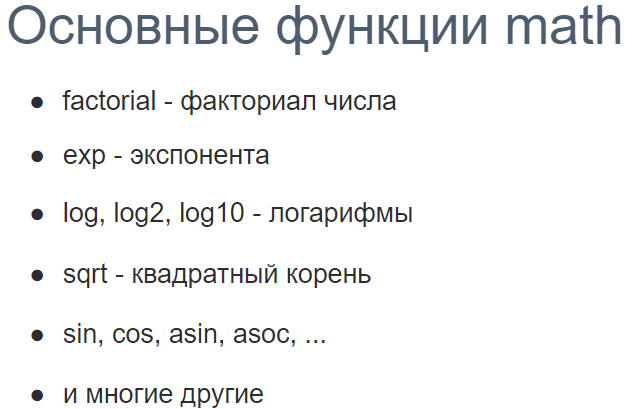
a,b,c,d = ( int(input()) for i in range(4) )

print('\t', \*range(c, d+1), sep='\t')

[ print(\*[i, \*range(i\*c, i\*d+1, i)], sep='\t') for i in range(a, b+1) ]

**МОДУЛИ**

|  |  |  |
| --- | --- | --- |
| **import math**  **import random as rd**  **print(math.pi)**  **print(math.sin(38))**  **print(rd.randint(1, 10))** | **from math import \***  **print(pi)**  **print(sin(30))** | **from random import randint, randrange**  **print(randint(1, 10))** |
| **вывод**  **3.141592653589793**  **0.2963685787093853**  **3** | **вывод**  **3.141592653589793**  **-0.9880316240928618** | **вывод**  **7** |

****

**# Найти длину окружности с определенным радиусом**

**from math import pi, pow, sqrt, factorial**

**r = int(input())**

**print(2 \* pi \* r)**

**# Найти площадь окружности с определенным радиусом**

**print(pi \* r \*\* 2)**

**print(pi \* pow(r, 2))**

**# По координатам 2-х точек определить расстояние между ними**

**x1 = 10**

**y1 = 10**

**x2 = 50**

**y2 = 100**

**print(sqrt((x1 - x2) \*\* 2 + (y1 - y2) \*\* 2))**

**#Найти факториал числа 9**

**print(factorial(9))**

****

**import random**

**# Загадать случайное число от 0 до 100**

**print(random.randint(0, 100))**

**#Выбрать победителя лотереи из списка Players**

**players = ['Max', 'Leo', 'Kate', 'Ron', 'Bill']**

**print(random.choice(players))**

**# выбрать 3-х победителей лотереий из списка Players**

**print(random.sample(players, 3))**

**# перемешать карты в стопке (списке) cards**

**cards = ['6', '7', '8', '9', 'T', 'J', 'Q', 'K', 'A']**

**random.shuffle(cards)**

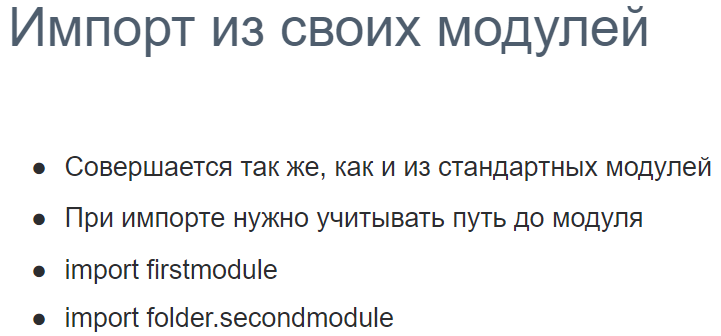
**print(cards)**

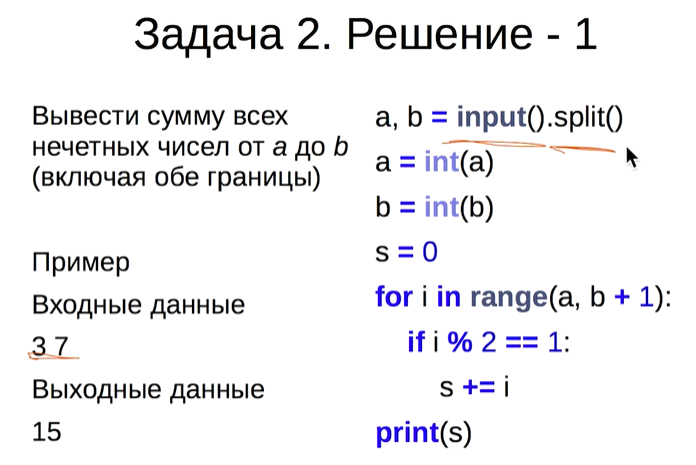
Создание своего модуля

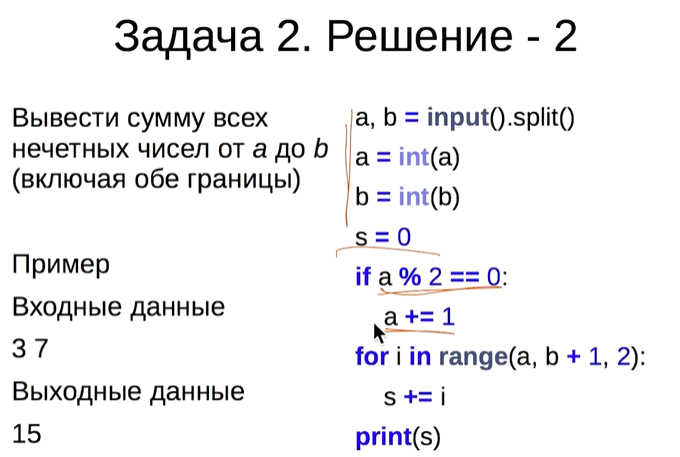
Импорт данных из своего модуля

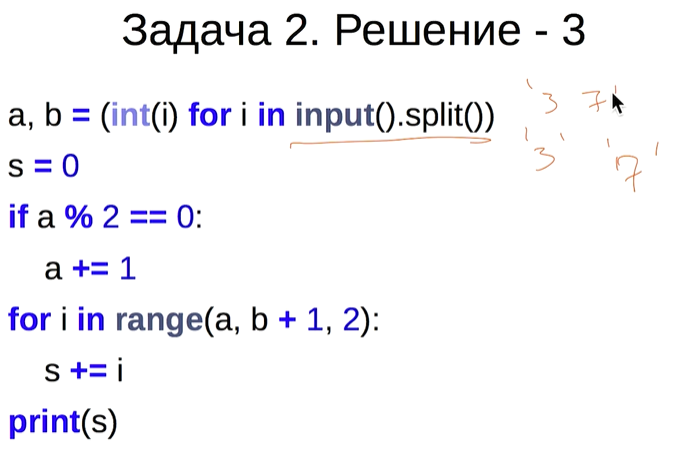
Импорт скриптов

if \_\_name\_\_ == ‘\_\_main\_\_’

****

****

****

****

Напишите программу, которая считывает с клавиатуры два числа a*a* и b*b*, считает и выводит на консоль среднее арифметическое всех чисел из отрезка [a; b][*a*;*b*], которые кратны числу 33.

В приведенном ниже примере среднее арифметическое считается для чисел на отрезке [-5; 12][−5;12]. Всего чисел, делящихся на 33, на этом отрезке 66: -3, 0, 3, 6, 9, 12−3,0,3,6,9,12. Их среднее арифметическое равно 4.54.5

На вход программе подаются интервалы, внутри которых всегда есть хотя бы одно число, которое делится на 33.﻿

**Sample Input:**

-5

12

**Sample Output:**

4.5

**# put your python code here**

**a, b = int(input()), int(input())**

**sum = 0**

**count = 0**

**for i in range (a,b+1):**

**if i % 3 == 0:**

**sum += i**

**count += 1**

**i += 1**

**print(sum / count)**

Верное решение #20768494

Python 3

a,b = int(input()), int(input())

a += -a%3

b -= b%3

print((a+b)/2)

Верное решение #24348232

Python 3

x = [x for x in range(int(input()),int(input()) + 1) if x % 3 == 0]

print(sum(x)/len(x))

Верное решение #61516211

Python 3

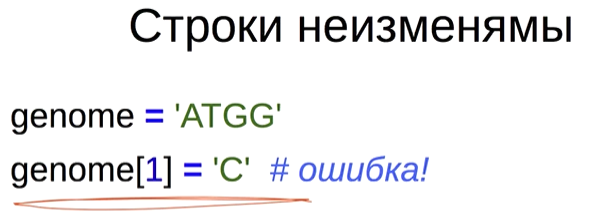
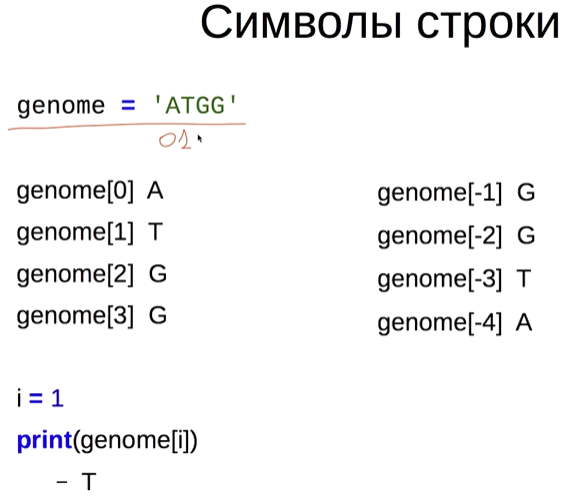
print((((int(input())-1)//3)+1+int(input())//3)\*1.5)

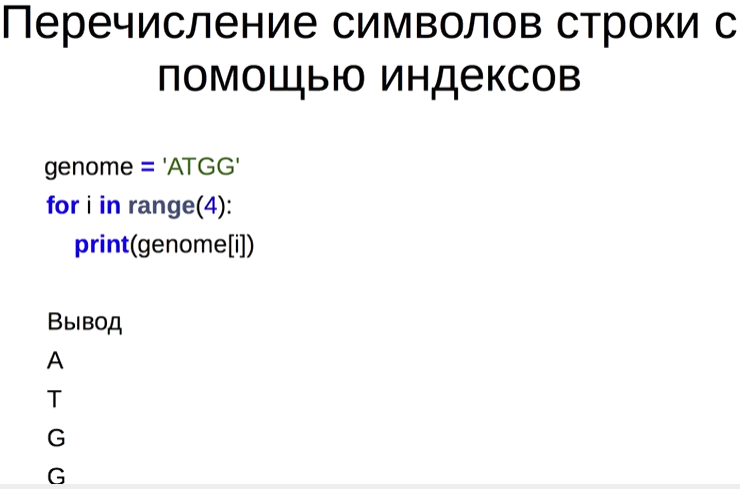
Верное решение #44578372

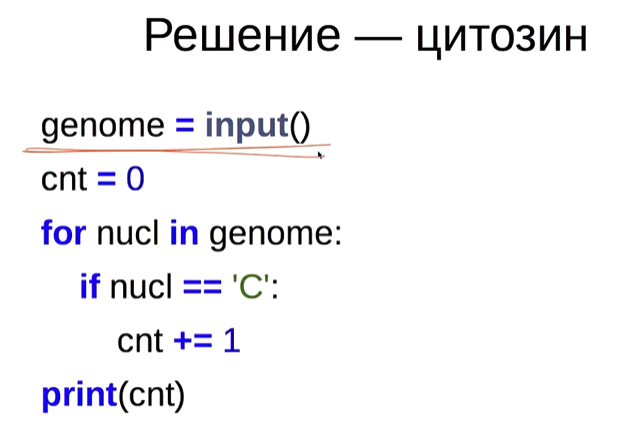
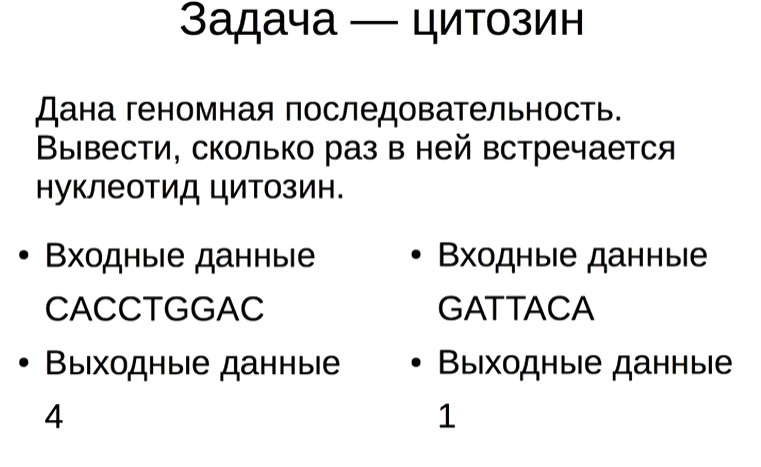
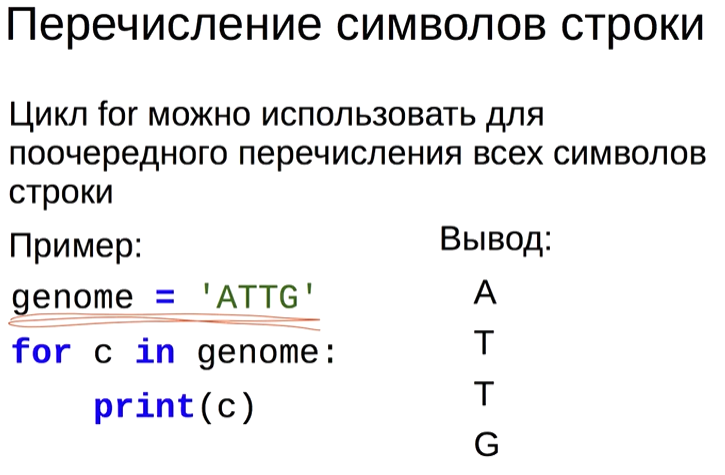
Python 3

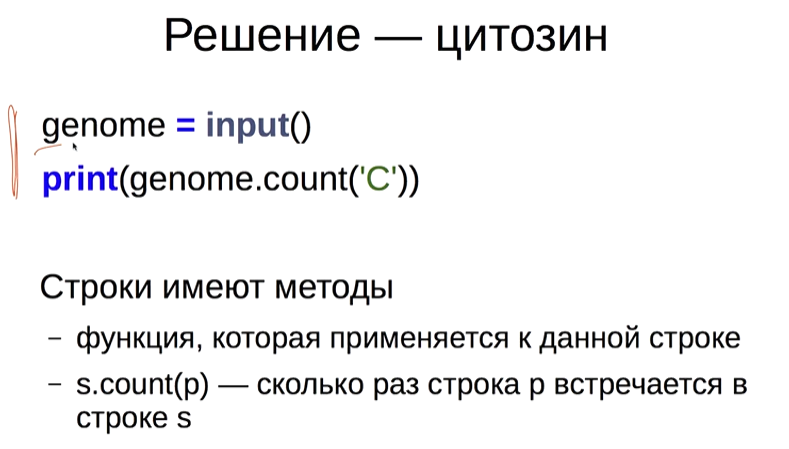
print((((int(input()) + 2 ) // 3)+(int(input()) // 3))/2\*3)

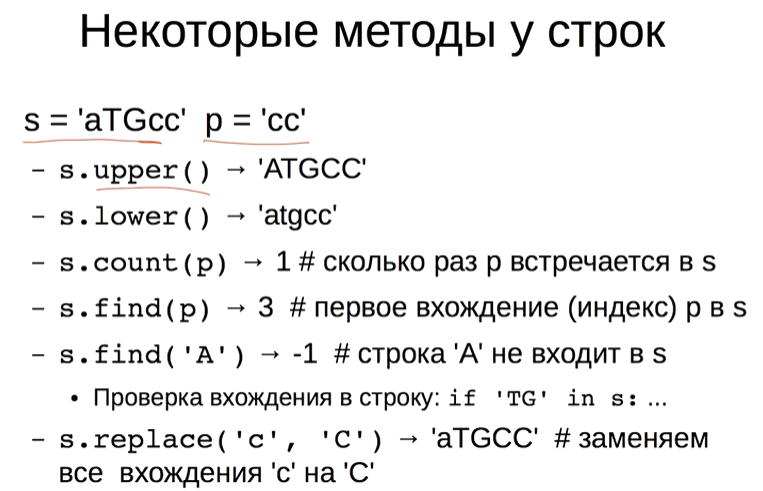
**2.4 Строки и символы**

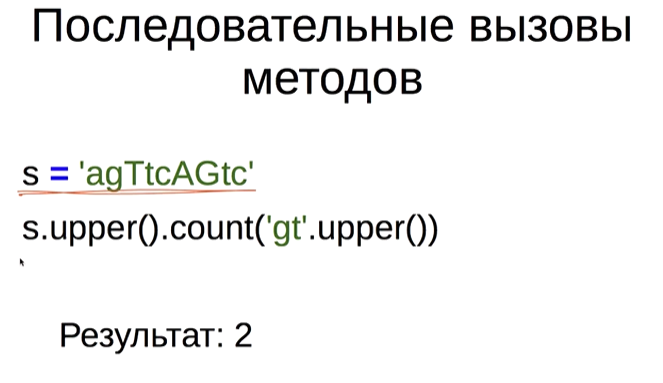
****

****

****

****

****

****

[GC-состав](https://ru.wikipedia.org/wiki/GC-%D1%81%D0%BE%D1%81%D1%82%D0%B0%D0%B2) является важной характеристикой геномных последовательностей и определяется как процентное соотношение суммы всех гуанинов и цитозинов к общему числу нуклеиновых оснований в геномной последовательности.

Напишите программу, которая вычисляет процентное содержание символов G (гуанин) и C (цитозин) в введенной строке (программа не должна зависеть от регистра вводимых символов).

Например, в строке "acggtgttat" процентное содержание символов G и C равно \dfrac4{10} \cdot 100 = 40.0104​⋅100=40.0, где 4 -- это количество символов G и C,  а 10 -- это длина строки.

**Sample Input:**

acggtgttat

**Sample Output:**

40.0

**n = input()**

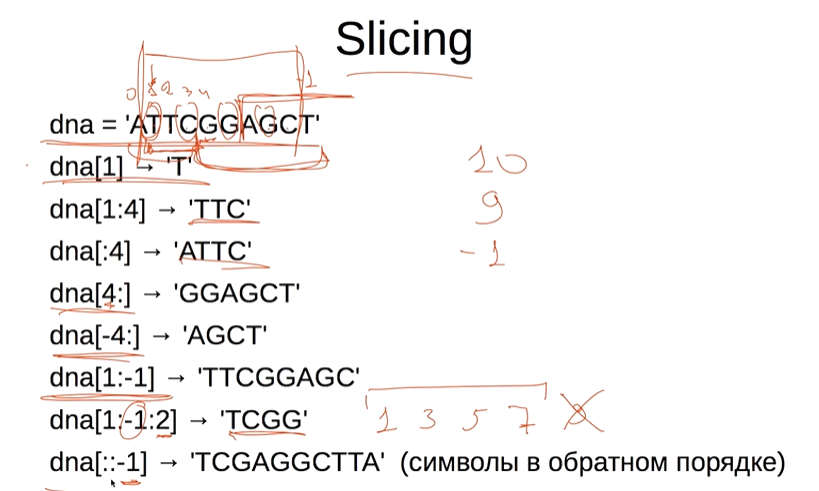
**print((n.lower().count('c') + n.lower().count('g')) / len(n) \* 100)**

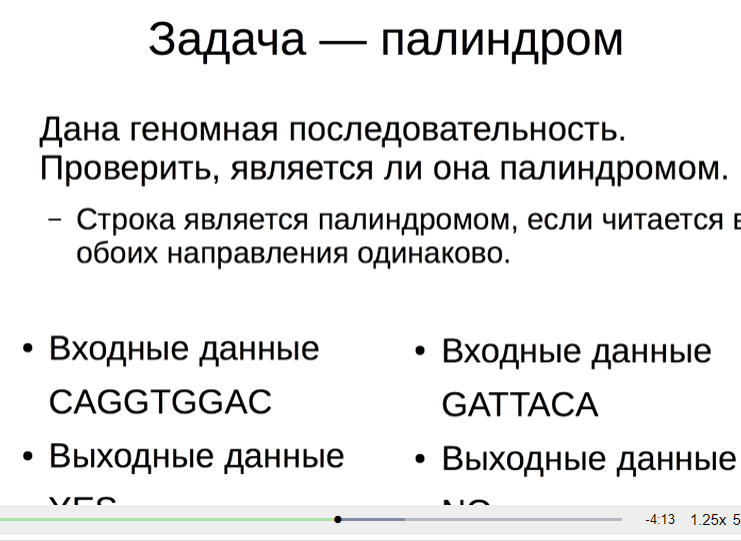
Верное решение #26263333

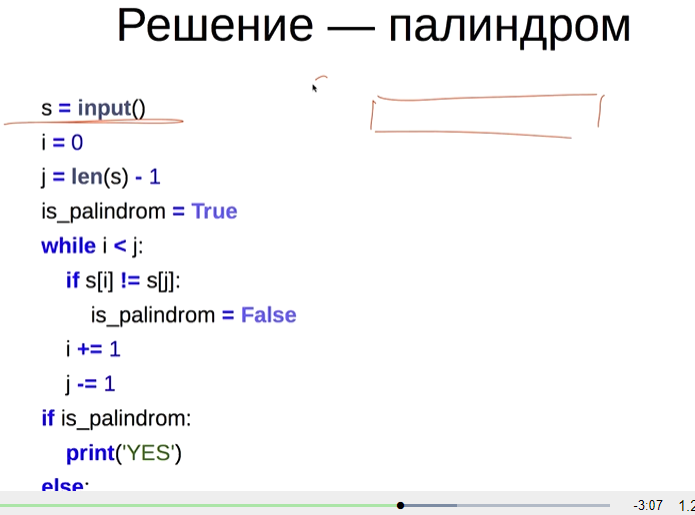
Python 3

s = input().upper()

print((s.count('G') + s.count('C'))/len(s) \* 100)

****

****

****

|  |  |
| --- | --- |
| **s = input()**  **i = 0**  **j = len(s)-1**  **is\_palidrom = True**  **while i<j:**  **if s[i] != s[j]:**  **is\_palidrom = False**  **i += 1**  **j -= 1**  **if is\_palidrom:**  **print('YES')**  **else:**  **print('NO')** | **s = input()**  **i = 0**  **j = len(s)-1**  **is\_palidrom = True**  **while i<j:**  **if s[i] != s[j]:**  **is\_palidrom = False**  **break**  **i += 1**  **j -= 1**  **if is\_palidrom:**  **print('YES')**  **else:**  **print('NO')** |
|  | **s = input()**  **r = s[::-1]**  **j = len(s)-1**  **if s == r:**  **print('YES')**  **else:**  **print('NO')** |

|  |  |
| --- | --- |
| **s = 'abcdefghijk'**  **print(s[3:6])**  **print(s[:6])**  **print(s[3:])**  **print(s[::-1])**  **print(s[-3:])**  **print(s[:-6])**  **print(s[-1:-10:-2])** | **вывод**  **def**  **abcdef**  **defghijk**  **kjihgfedcba**  **ijk**  **abcde**  **kigec** |

Узнав, что ДНК не является случайной строкой, только что поступившие в Институт биоинформатики студенты группы информатиков предложили использовать алгоритм сжатия, который сжимает повторяющиеся символы в строке.

Кодирование осуществляется следующим образом:

s = 'aaaabbсaa' преобразуется в 'a4b2с1a2', то есть группы одинаковых символов исходной строки заменяются на этот символ и количество его повторений в этой позиции строки.

Напишите программу, которая считывает строку, кодирует её предложенным алгоритмом и выводит закодированную последовательность на стандартный вывод. Кодирование должно учитывать регистр символов.

**Sample Input 1:**

aaaabbcaa

**Sample Output 1:**

a4b2c1a2

**Sample Input 2:**

abc

**Sample Output 2:**

a1b1c1

**s = input()**

**count = 1**

**for i in range(len(s)):**

**if i == (len(s)-1):**

**print(s[i] + str(count), end = '')**

**else:**

**if s[i] == s[i+1]:**

**count += 1**

**else:**

**print(s[i] + str(count),end = '')**

**count = 1**

Верное решение #28646944

Python 3

genome = input()+' '

s = 0

n=genome[0]

for i in genome:

if n!=i:

print(n + str(s), end = '')

s=0

n=i

s+=1

Верное решение #24808228

Python 3

dna = input() # считываем строку

print(dna[0],end='') # выводим первый символ

cnt = 1 # счетчик символов на единице

for i in range(0,len(dna)-1): # итератор проходит по всем индексам символов кроме предпоследнего

if dna[i] == dna[i+1]: # сравниваем символ по текущему индексу со следующим

cnt+=1 # если символы одинаковые, то увеличиваем счетчик

else :

print(cnt,end='') # если разные, то выводим значение счетчика

print(dna[i+1],end='') # выводим следующий символ

cnt = 1 # счетчик текущего символа на единице

print(cnt) # в конце распечатываем значение счетчика последнего символа

Верное решение #36087001

Python 3

s,n = 0,input()+' '

for i in n:

if n[0] != i:

print(n[0],s,end='',sep='')

s,n = 0,i

s += 1

Верное решение #29208057

Python 3

a = input()+' '

count = 1

for i in range(len(a)-1):

if a[i] != a[i + 1]:

print(a[i] + str(count), end='')

count = 1

else:

count += 1

Верное решение #46315579

Python 3

s=input()+' '

x=1

for i in range (0,len(s)-1):

if s[i]==s[i+1]:

x+=1

else:

print((s[i]+str(x)),end='')

x=1

Верное решение #209950961

Python 3

s = input() + ' '

w = 0

for i in range(len(s)-1):

if s[i] != s[i+1]:

print(s[i],len(s[w:i+1]), sep='', end='')

w=i+1

Верное решение #590397148

Python 3

# put your python code here

genome = input()

while genome:

print(genome[0], len(genome) - len(genome.lstrip(genome[0])), sep='', end='')

genome = genome.lstrip(genome[0])

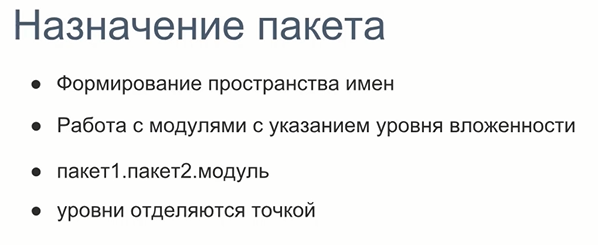
if \_\_name\_\_ == ‘\_\_main\_\_’

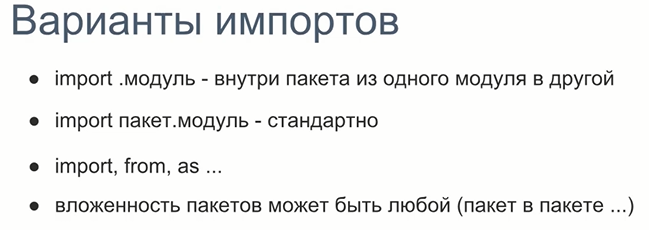
Ограничивает выполнение скриптов

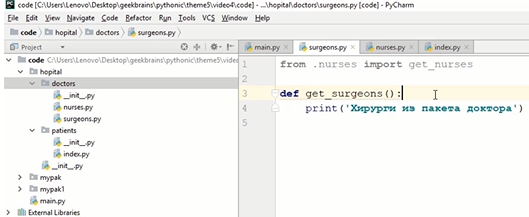
При импорте код не будет выполняться

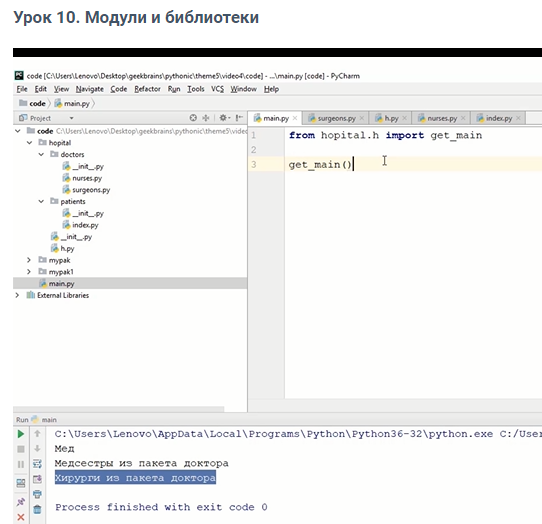
Он будет выполняться при запуске модуля

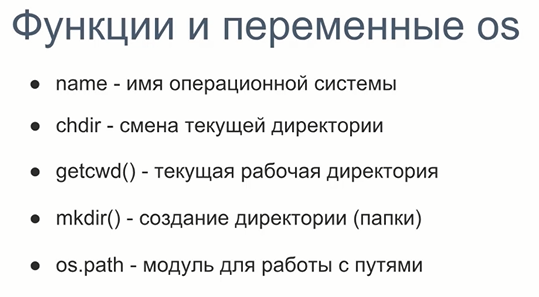
**ПАКЕТЫ**

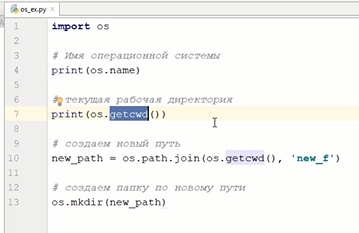
****

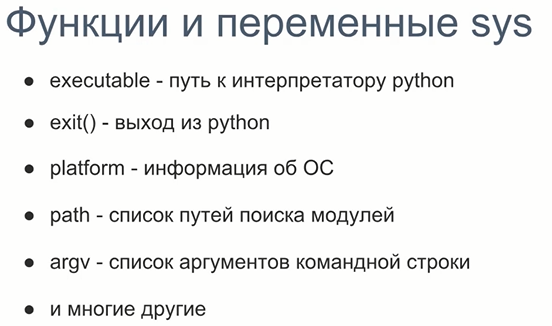
****

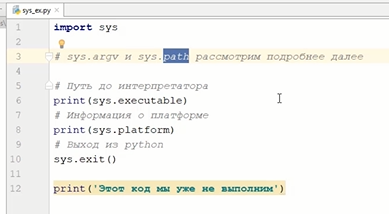
****

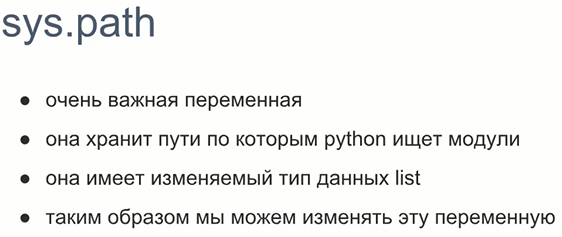
****

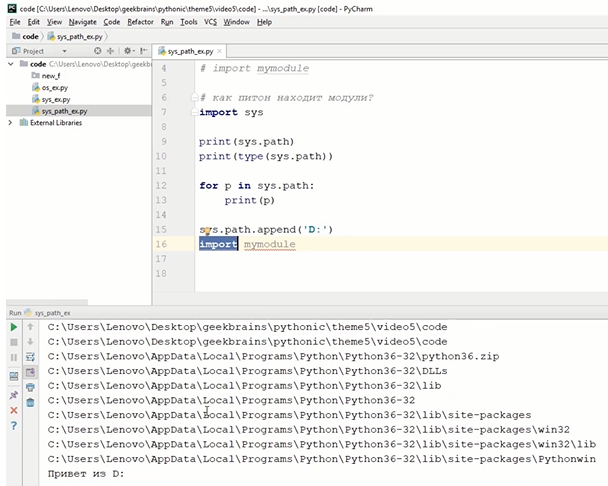
****

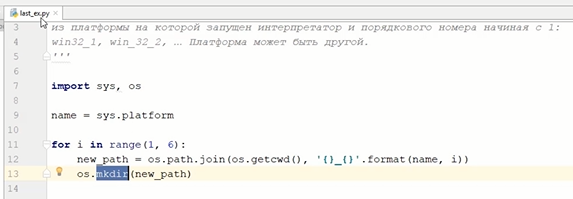
****

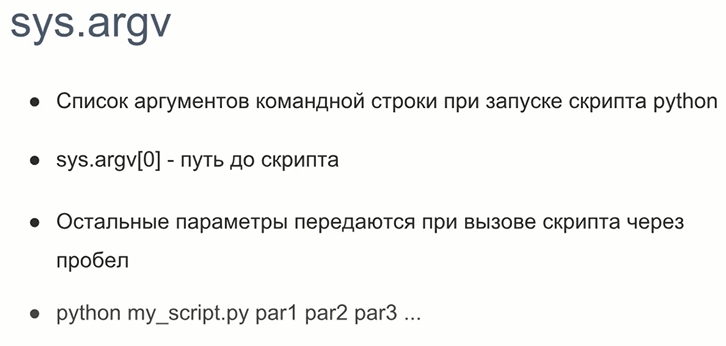
****

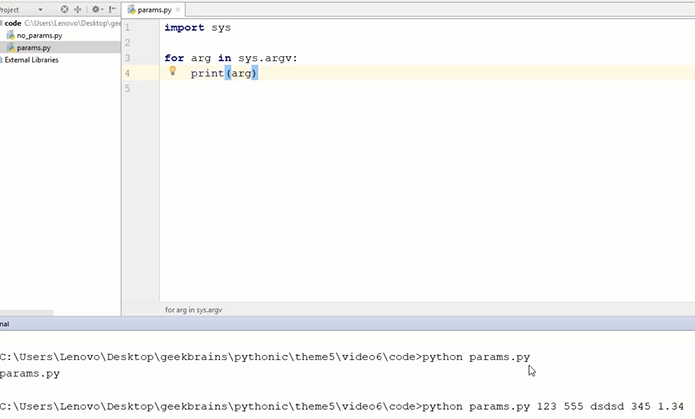
****

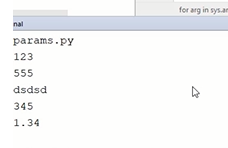
****

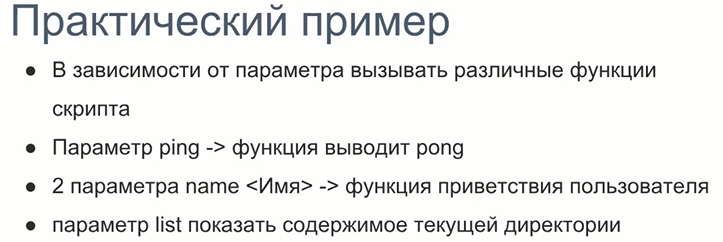
****

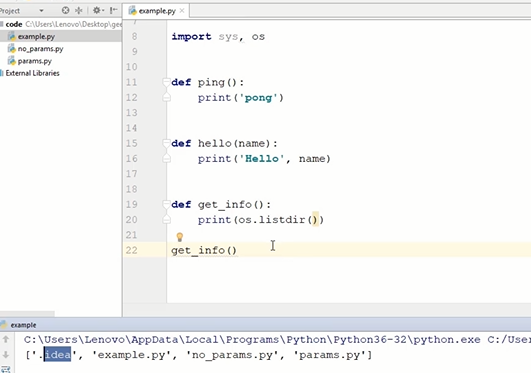
****

****

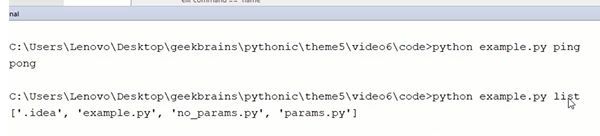
****

****

****

****

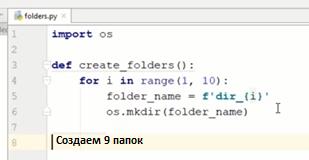
****

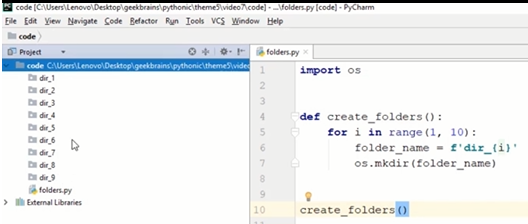
****

****

## Практическое задание

### 1: Создайте модуль (модуль - программа на Python, т.е. файл с расширением .py). В нем создайте функцию создающую директории от dir\_1 до dir\_9 в папке из которой запущен данный код. Затем создайте вторую функцию удаляющую эти папки. Проверьте работу функций в этом же модуле.

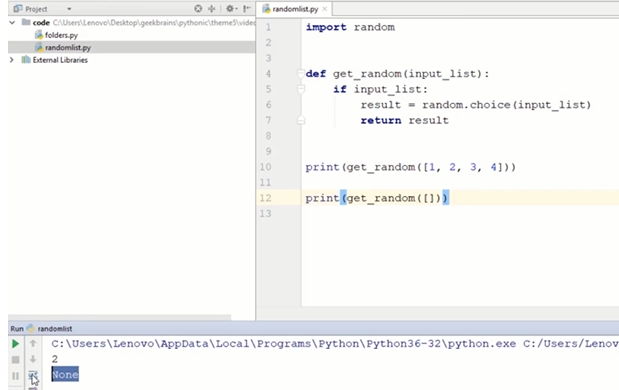






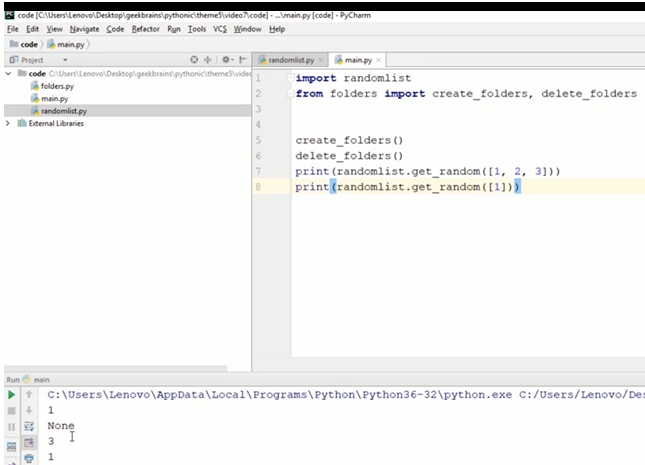
### 2: Создайте модуль. В нем создайте функцию, которая принимает список и возвращает из него случайный элемент. Если список пустой функция должна вернуть None. Проверьте работу функций в этом же модуле.

Примечание: Список для проверки введите вручную. Или возьмите этот: [1, 2, 3, 4]

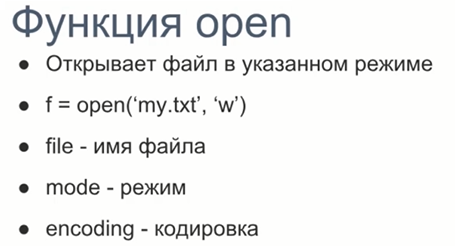


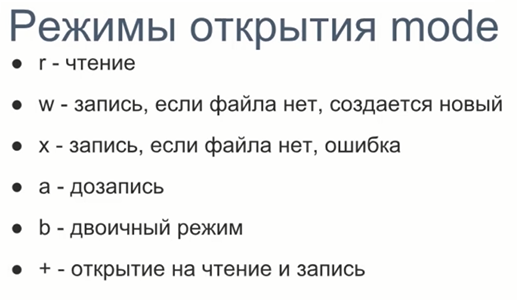
### 3: Создайте модуль main.py. Из модулей реализованных в заданиях 1 и 2 сделайте импорт в main.py всех функций. Вызовите каждую функцию в main.py и проверьте что все работает как надо.

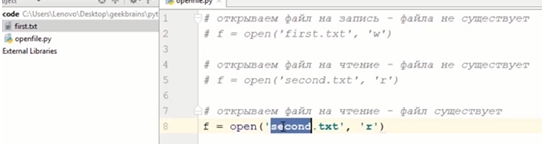
Примечание: Попробуйте импортировать как весь модуль целиком (например из задачи 1), так и отдельные функции из модуля.

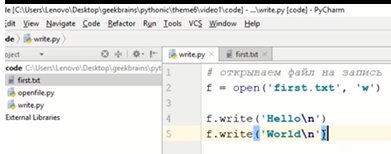


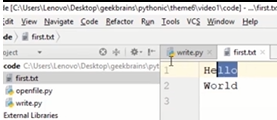
### Урок 12. Работа с файлами. Кодировки

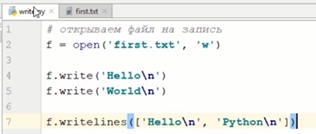
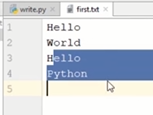




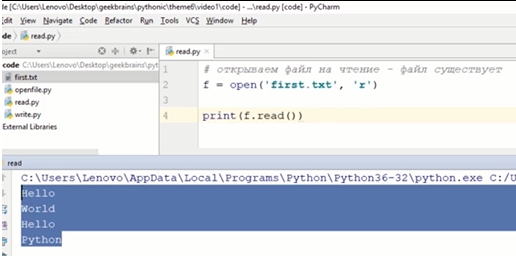


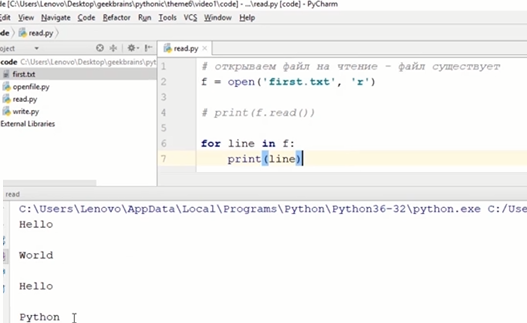


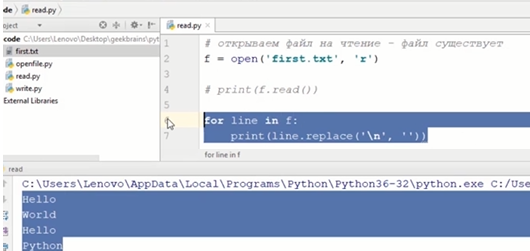


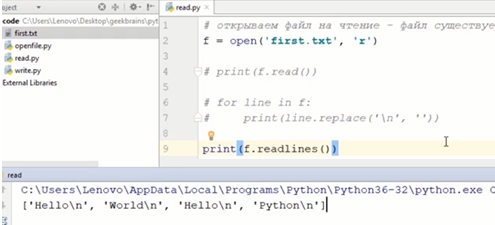
 

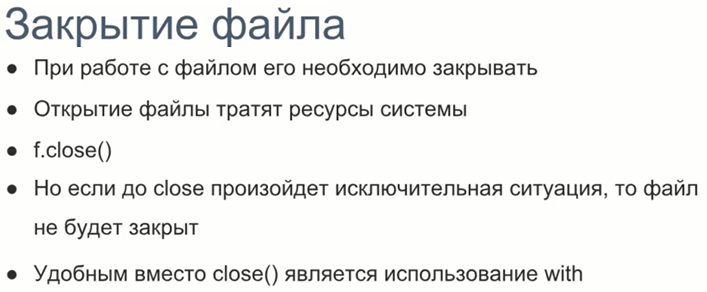
Чтение файла



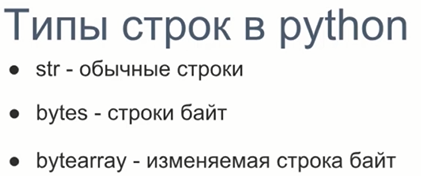


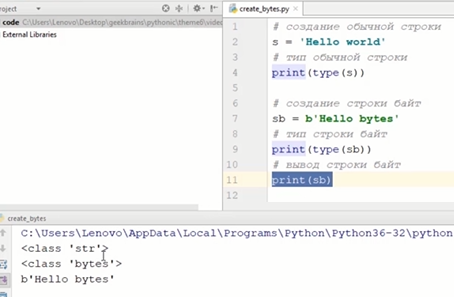


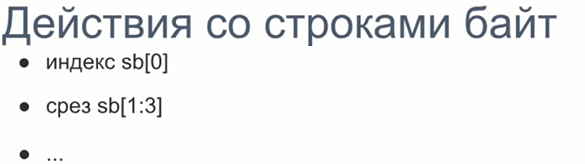


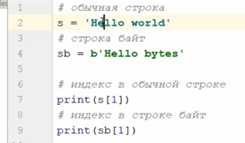


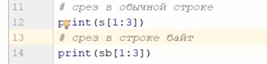


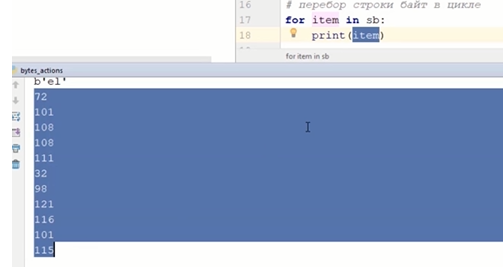


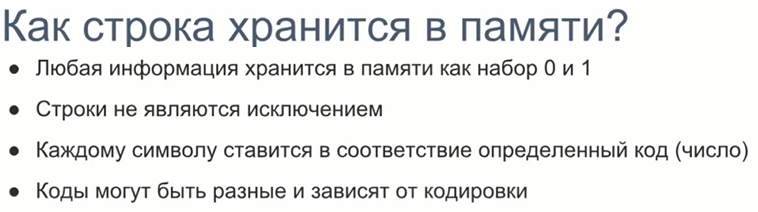


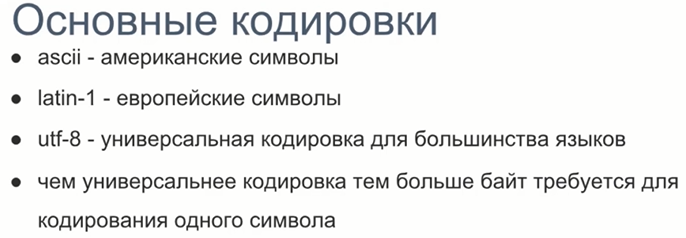
****

****

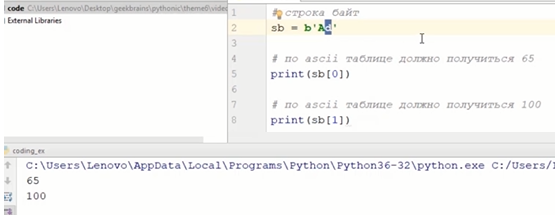
****

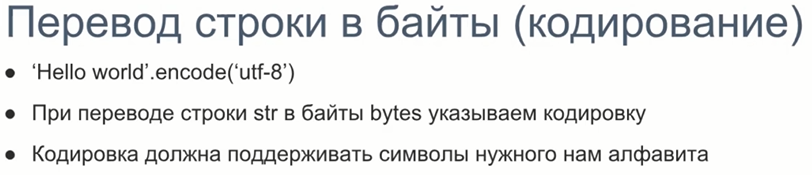
****

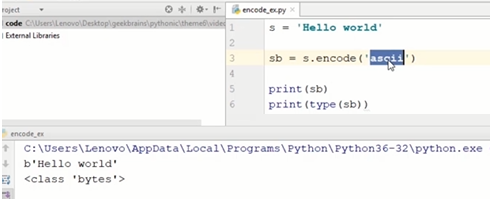
****

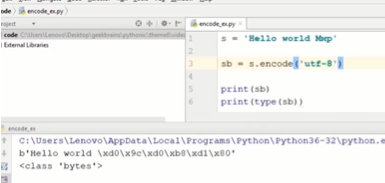
****

****

****

****

****

****

## 

## 

## 

## 

## 

## 

## Практическое задание

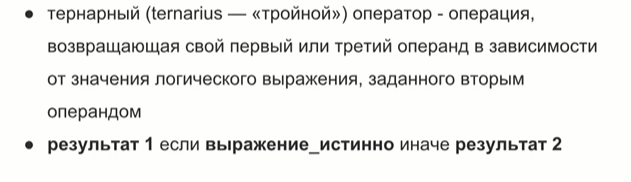
### 1: Создать модуль music\_serialize.py. В этом модуле определить словарь для вашей любимой музыкальной группы, например:

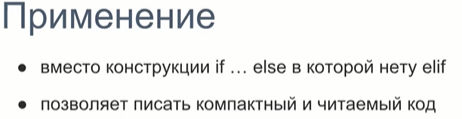
my\_favourite\_group = {  
‘name’: ‘Г.М.О.’,  
‘tracks’: [‘Последний месяц осени’, ‘Шапито’],  
‘Albums’: [{‘name’: ‘Делать панк-рок’,‘year’: 2016},  
{‘name’: ‘Шапито’,‘year’: 2014}]}

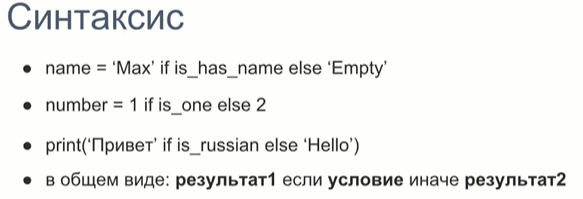
### С помощью модулей json и pickle сериализовать данный словарь в json и в байты, вывести результаты в терминал. Записать результаты в файлы group.json, group.pickle соответственно. В файле group.json указать кодировку utf-8.

### 2: Создать модуль music\_deserialize.py. В этом модуле открыть файлы group.json и group.pickle, прочитать из них информацию. И получить объект: словарь из предыдущего задания.

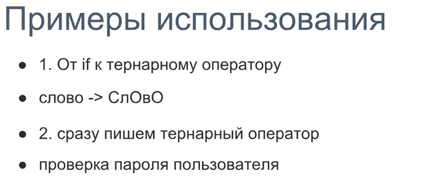
### **Урок 14. Полезные инструменты. Обработка исключений**

****

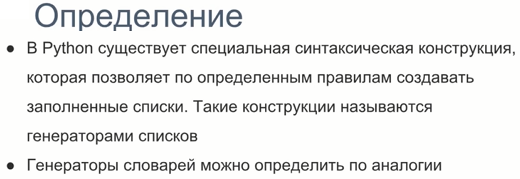
****

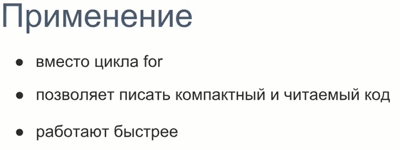
****

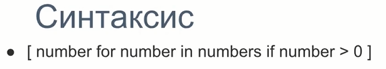
|  |  |
| --- | --- |
| **is\_has\_name = True**  **name = 'Max' if is\_has\_name else 'Empty'**  **print(name)** | Max |
| **is\_one = False**  **number = 1 if is\_one else 2**  **print(number)** | 2 |
|  |  |



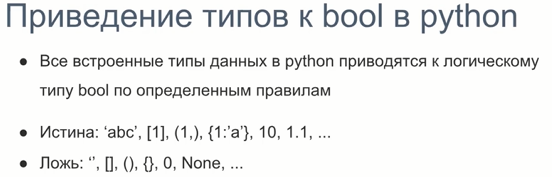
|  |  |
| --- | --- |
| word = 'word'  result = []  for i in range(len(word)):  if i%2 !=0:  letter = word[i].lower()  else:  letter = word[i].upper()  result.append(letter)  result = ''.join(result)  print(result) | WoRd |
| word = 'Фабрика'  result = []  for i in range(len(word)):  letter = word[i].lower() if i%2 != 0 else word[i].upper()  result.append(letter)  result = ''.join(result)  print(result) | ФаБрИкА  > |
| password = input('Enter password')  print('Enter' if password == 'secret' else 'no entry') | Enter password secret  Enter |







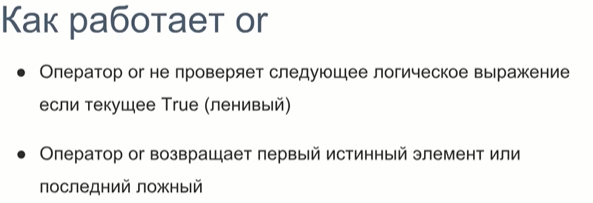
|  |  |
| --- | --- |
| numbers = [1, 2, 3, 4, 5, 6, -1, -2, -3]  # Классический способ  result = []  **for number in numbers:**  **if number > 0:**  **result.append(number)**  **print(result)**  # через функцию filter  **result = filter(lambda number: number > 0, numbers)**  **print(list(result))**  # через генератор  **result = [number for number in numbers if number > 0]**  **print(result)** | [1, 2, 3, 4, 5, 6]  [1, 2, 3, 4, 5, 6]  [1, 2, 3, 4, 5, 6]  > |
| pairs = [(1, 'a'), (2, 'b'), (3, 'c')]  # Классический способ  result = {}  for pair in pairs:  key = pair[0]  val = pair[1]  result[key] = val  print(result)  # через генератор  result = {pair[0]: pair[1] for pair in pairs}  print(result) | {1: 'a', 2: 'b', 3: 'c'}  {1: 'a', 2: 'b', 3: 'c'}  > |
| #Создать список из случайных чисел от 1 до 100  import random  numbers = [random.randint(1, 100) for i in range(10)]  print(numbers)  # Создать список квадратов чисел  import numpy as np  numbers = [1, 2, 3, 4]  numbers = [np.power(number, 2) for number in numbers]  print(numbers)  # Создать список имен на букву А  names = ['Ivan', 'Ann', 'Alex', 'Peter']  names = [name for name in names if 'A' in name]  print(names)  names = [name for name in names if name.startswith('A')]  print(names) | [43, 76, 95, 73, 22, 92, 82, 74, 65, 74]  [1, 4, 9, 16]  ['Ann', 'Alex']  ['Ann', 'Alex']  > |



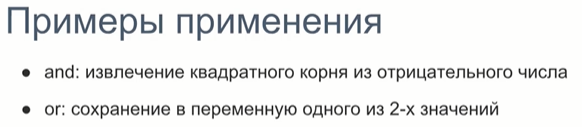
|  |  |
| --- | --- |
| s = 'abc'  #classic method  if len(s) != 0:  print('String not empty')  else:  print('String empty')  # comfortable method  if s:  print('String not empty')  else:  print('String empty') | String not empty  String not empty |
| l = [1, 2, 3]  d = {1: 'a'}    #classic method  if len(l) != 0 and len(d) != 0:  print('List and String not empty')  else:  print('List and String empty')  # comfortable method  if l and d:  print('List and String not empty')  else:  print('List and String empty') | List and String not empty  List and String not empty |



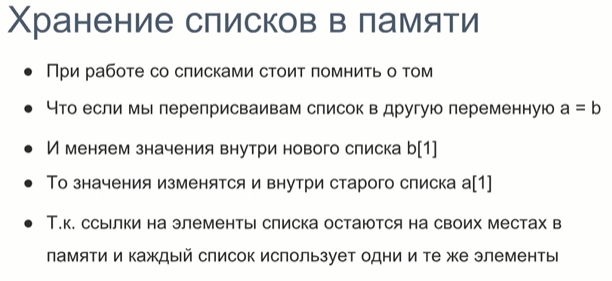
|  |  |
| --- | --- |
| **import math**  **if 1 > 2 and math.sqrt(-1):**  **print('ошибки не будет т.к. первое условие ложь')**  **print('Двигаемся дальше')**  **if math.sqrt(-1) and 1 > 2:**  **print('Если променять местами то будет ошибка')**  **print([1] and [] and '' and 1)**  **print([1] and 1 and 20 and 1.1)** | Двигаемся дальше  []  1.1 |
|  |  |

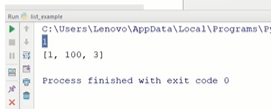
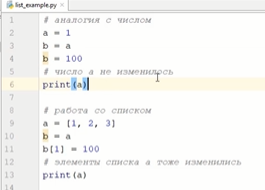
****

|  |  |
| --- | --- |
| **import math**  **if 2 > 1 or math.sqrt(-1):**  **print('ошибки не будет т.к. первое условие ложь')**  **print(0 or [] or 8 or 5)**  **print(0 or [] or () or ())** | ошибки не будет т.к. первое условие ложь  8  () |

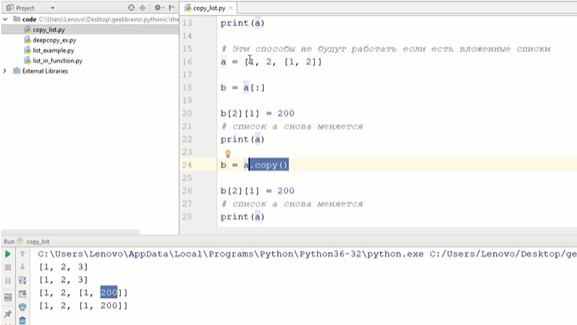
****

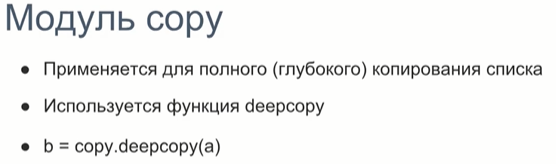
|  |  |
| --- | --- |
| **import math**  **nums = [4, 1, 2, 3, -4, -2, 7, 16]**  **result = []**  **for num in nums:**  **if num > 0:**  **sqrt = math.sqrt(num)**  **if sqrt < 2:**  **result.append(num)**  **print(result)**  **# через ленивый and**  **result = []**  **for num in nums:**  **if num > 0 and math.sqrt(num) < 2:**  **result.append(num)**  **print(result)**  **# через генератор**  **result = [num for num in nums if num > 0 and math.sqrt(num) < 2]**  **print(result)**  **#добавление элемента в список**  **def add\_to\_list(input\_list=None):**  **if input\_list is None:**  **input\_list = []**  **input\_list.append(2)**  **return input\_list**  **result = add\_to\_list([0, 1])**  **print(result)**  **result = add\_to\_list()**  **print(result)**  **#через or**  **def add\_to\_list(input\_list=None):**  **input\_list = input\_list or []**  **input\_list.append(2)**  **return input\_list**  **result = add\_to\_list([0, 1])**  **print(result)**  **result = add\_to\_list()**  **print(result)** | [1, 2, 3]  [1, 2, 3]  [1, 2, 3]  [0, 1, 2]  [2]  [0, 1, 2]  [2] |

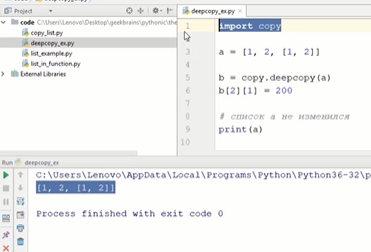
****

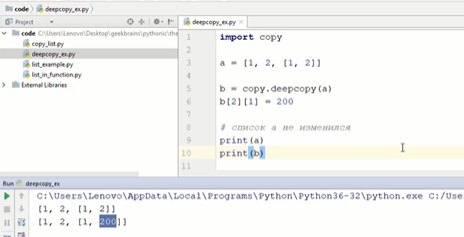
****

|  |  |
| --- | --- |
| **a = [1, 2, 3]**  **b = a[:]**  **b[1] = 200**  **print(a)**  **b = a.copy()**  **b[1] = 200**  **print(a)** | **[1, 2, 3]**  **[1, 2, 3]** |
|  |  |

****

****

****

****