Официальный сайт Python, где можно скачать интерпретатор (Python 3): [https://www.python.org/](https://www.python.org/#_blank)

Официальная документация по Python: [https://docs.python.org/3/](https://docs.python.org/3/#_blank)

Веб-сервис, позволяющий исполнять программы на Python прямо в вашем браузере: [https://trinket.io/python/41462f0f16](https://trinket.io/python/41462f0f16#_blank)

Среда для написания программ PyCharm Educational Edition или PyCharm Community Edition:  
[https://www.jetbrains.com/pycharm-educational/](https://www.jetbrains.com/pycharm-educational/#_blank)  
[https://www.jetbrains.com/pycharm/](https://www.jetbrains.com/pycharm/#_blank)

Текстовый редактор с подсветкой синтаксиса программ Sublime Text 3: [http://www.sublimetext.com/3](http://www.sublimetext.com/3#_blank)

Интерактивный учебник языка Python (на русском языке): [http://pythontutor.ru/](http://pythontutor.ru/#_blank)

Книга A Byte of Python: [http://www.swaroopch.com/notes/python/](http://www.swaroopch.com/notes/python/#_blank)  
и ее перевод на русский: [http://wombat.org.ua/AByteOfPython/](http://wombat.org.ua/AByteOfPython/#_blank)

Онлайн платформа для запуска программ на python: Trinket.io

Среда для локального запуска приложений на python: Jupyter Notebook

Установка Jupyter Notebook на Linux:

http://conda.pydata.org/miniconda.htm

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)

$ ./Miniconda3-latest-Linux-x86.sh

$ conda update conda --yes

$ conda install ipython --yes

$ conda install pyzmq --yes

$ conda install jinja2 --yes

$ conda install tornado --yes

$ conda install ipywidgets --yes

chown -R kblazheev: miniconda3

chmod -R 755 miniconda3

$ ipython3 notebook

**Операции с целыми числами**

+ сложение

- вычитание

\* умножение

/ деление

( ) приоритет операции

// частное от целочисленного деления

% остаток от целочисленного деления

x % 10 последняя цифра числа x

\*\* возведение в степень

-(x) сменить знак числа x

+(x) оставить знак числа x

**Работа с вещественными числами (с плавающей точкой)**

1 / 3 = 0.3333333333333333

0.3 + 0.3 + 0.3 = 0.8999999999999999

9 \*\* 0.5 = 3.0

5е-1 = 0.5

1234е-2 = 12.34

1234е2 = 123400.0

7.0 // 3 = 2.0 (округление в сторону -∞)

**Типы данных**

Неизменяемые (статические)

Числовые

int целые

float вещественные (действительные, с плавающей точкой)

bool логические

Строковые

str строковые

**Преобразование типов**

int(x) к целому типу (округление в сторону 0)

float(x) к вещественному типу

9 \*\* 19 - int(9.0 \*\* 19) = 89

type(7) = int

type(7.0) = float

**Переменные**

a = 5 5

a = a + 5 10

a += 5 15

a -= 5 10

a \*= 5 50

a /= 5 10.0

a // = 5 2.0

a %= 5 2.0

a \*\*= 5 32.0

a = 2

A = 3.0

print(a + A) = 5.0

**Динамическая типизация**

a = 2

type(a) = int

a = ꞌabcdꞌ

type(a) = str

a = foo()

…

**Ввод-вывод данных**

s = input('Введите текст: ') если хотим прочитать строку

a = int(input('Введите число: ')) если хотим прочитать число

print('Вы ввели:', s, a)

print('a','b','c',2) = a b c 2

X = int(input())

Y = int(input())

print(60 \* X + Y)

**Операции сравнения**

<

<=

>

>=

==

!=

a >= 10 and a <100

10 <= a < 100

**Логические операции**

False (0)

True (1)

Приоритет операций в порядке убывания (отрицание, логическое и, логическое или):

not x

x and y

x or y

x1, x2, x3 = False, True, False

not x1 or x2 and x3 = True

Сначала выполняются арифметические операции, затем операции сравнения, затем логические операции.

((y > (x \* x)) or ((y >= (2 \* x)) and (x < y)))

**Условия**

if условие\_0:

блок\_0

elif условие\_1:

блок\_1

...

elif условие\_n:

блок\_n

else:

блок

x1, x2, …, xn = 1, 2, …, n

m = x1

if x2 > m:

m = x2

...

elif xn > m:

m = xn

**Строки**

'string'

"string"

'''multiline

string'''

""multiline

string""

'abc' + 'def' = 'abcdef'

abc \* 3 = 'abcabcabc'

len('abcdef') = 6

'abc' < 'ac'

'abc' > 'ab'

\n

print('abc\ndef')

print('abc' + '\n' + 'def')

print('abc', '\n', 'def')

# комментарий

'''многострочный

комментарий'''

**Задание «вывод окончаний программиста»**

for x in range(0, 100):

y = 'программи'

if ((x % 10 == 1) and ((x // 10) % 10 != 1)):

y += 'ст'

elif (((x % 10 == 2) or (x % 10 == 3) or (x % 10 == 4)) and ((x // 10) % 10 != 1)):

y += 'ста'

else:

y += 'стов'

print(x, y)

**Задание «счастливый билет»**

s = input()

left = int(s[0]) + int(s[1]) + int(s[2])

right = int(s[3]) + int(s[4]) + int(s[5])

if left == right:

print('Счастливый')

else:

print('Обычный')

**Символ разделителя при печати**

for i in range(0, 10):

print(i, end = ' ')

a = range(1, 10, 2)

b = list(range(1, 10, 2)) # print out: [1, 3, 5, 7, 9]

for i in b:

print(i, end = ' ')

**Циклы**

a = 5

while a > 0:

print(a, end = ' ')

a -= 1

---------------------------------

i = 1

while i <= 6:

print('\*' \* i)

i += 1

---------------------------------

n = int(input())

s = '\*'

m = n

while len(s) <= n:

print(s, '\*' \* m)

s += '\*'

m -= 1

---------------------------------

a = int(input())

b = int(input())

d = 1

while not (d % a == 0 and d % b == 0):

d += 1

print(d)

------------------------------------------------------

a, b = input().split(';')

a = int(a)

b = int(b)

print(a,b)

a, b = input().split()

a = int(a)

b = int(b)

print(a,b)

---------------------------------------------------------------------------------

range(10) то же что и range(0,10) и range(0,10,1)

list(range(10)) # print out: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

list('список') # print out: ['с', 'п', 'и', 'с', 'о', 'к']

----------------------------------------------------------------------------------

for i in 2,3,5:

print(i \* i)

for i in range(10):

print(i \* i)

n = int(input())

for i in range(n):

print('\*' \* n)

n = int(input())

for i in range(n):

for j in range(n):

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

print()

----------------------------------------------------------------------------------

a = int(input())

b = int(input())

c = int(input())

d = int(input())

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

if i == a:

print('\t', end = '')

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

print(j, '\t', end = '')

print()

print(i, '\t', end = '')

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

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

print()

---------------------------------------------------------------------------------

a, b = input().split()

a = int(a)

b = int(b)

s = 0

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

if i % 2 != 0:

s += i

print(s)

----------------------------------------------------------------------------------

a, b = (int(i) for i in input().split())

s = 0

if a % 2 == 0:

a += 1

for i in range(a, b + 1, 2):

s += i

print(s)

**Строки**

genome = 'ATGC'

genome[0] = 'A'

genome[1] = 'T'

genome[2] = 'G'

genome[3] = 'C'

genome[-1] = 'C'

genome[-2] = 'G'

genome[-3] = 'T'

genome[-4] = 'A'

for i in genome:

print(i)

genome[1] = 'C' # ошибка

genome = input()

print(genome.count('C'))

genome.upper()

genome.lower()

genome.count('C')

genome.find('C')

genome.replace('C' ,'c')

s = 'CGtttCcccAgT'

s.upper().count('Gt'.upper())

-----------------------------------------------------------------------------

g = input()

gc = (g.upper().count('G') + g.upper().count('C')) \* 100 / len(g)

print(gc)

------------------------------------------------------------------------------

s = 'ATTCGGAGCT'

s[1] = 'T'

s[1:4] = 'TTC'

s[:4] = 'ATTC'

s[4:] = 'GGAGCT'

s[-4:] = 'AGCT'

s[1:-1] = 'TTCGGAGC'

s[1:-1:2] = TCGG

s[::-1] = 'TCGAGGCTTA'

------------------------------------------------------------------------------

dna =input()

dna2 = dna[0]

l = len(dna)

i = 1

if l == 1:

dna2 = dna[0] + str(1)

else:

for n in range(1, l):

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

i += 1

if n == l - 1:

dna2 += str(i)

elif i > 1:

dna2 += str(i) + dna[n]

i = 1

else:

dna2 += str(1) + dna[n]

if n == l - 1:

dna2 += str(1)

print(dna2)

**Списки**

stu = ['Misha', 'Masha', 'Sasha']

stu[0] == stu[-3] == 'Misha'

stu[1] == stu[-2] == 'Masha'

stu[2] == stu[-1] == 'Sasha'

len(stu) = 3

stu[::-1] == ['Sasha', 'Masha', 'Misha']

stu[:2] == ['Misha', 'Masha']

teach = ['Oleg', 'Alex']

stu + teach = ['Sasha', 'Masha', 'Misha', 'Oleg', 'Alex']

stu \* 2 = ['Misha', 'Masha', 'Sasha', 'Misha', 'Masha', 'Sasha']

stu.append('Dasha', 'Olga') # Добавить в конец

stu.insert(1, 'Dasha') # Вставить в позицию вместо

stu.remove('Olga')

stu.index('Sasha')

stu.sort() # изменит список

sorted(stu) # не изменит список

stu.min()

stu.max()

stu.reverse()

reversed(stu)

del stu[0]

if 'Olga' in stu

if 'Olga' not in stu

stu += ['Olga', 'Olga']

s = [] # пустой список

s += 'Olga'

s == ['O', 'l', 'g', 'a']

**Присвоение списков**

a = [1, 2, 3]

b = a

a[0] = 5

b == [5, 2, 3]

a = [1, 2]

b == [5, 2, 3]

a = [0] \* 5

a = [0 for i in range(5)]

a = [i \* i for i in range(5)]

a = [int(i) for i in input().split()]

----------------------------------------------------------

a = [int(i) for i in input().split()]

s = 0

for i in range(len(a)):

s += a[i]

print(s)

---------------------------------------------------------

a = [int(i) for i in input().split()]

if len(a) == 1:

print(a[0])

else:

l = len(a) - 1

for i in range(l):

print(a[i - 1] + a[i + 1], end = ' ')

print(a[-2] + a[0])

----------------------------------------------------------

a = [int(i) for i in input().split()]

a.sort()

f = 0

b = ''

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

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

if f == 0:

b += str(a[i]) + ' '

f = 1

else:

f = 0

print(b[0:-1])

----------------------------------------------------------

a = [[1,2,3], [4,5,6], [7,8,9]]

a[1] == [4,5,6]

a[2][2] == 9

----------------------------------------------------------

a = [[0]\*n for i in range(n)]

t = True

s1 = 0

s2 = 0

while t:

a = int(input())

s1 += a

s2 += a \* a

if s1 == 0:

print(s2)

t = False

---------------------------------------------------------

n = int(input())

j = 1

k = 1

t = True

while t:

for i in range(1,j + 1):

if k < n:

print(j, end=' ')

elif k == n:

print(j, end='')

else:

t = False

k += 1

j += 1

-----------------------------------------------------------

a = [int(i) for i in input().split()]

x = int(input())

s = ''

while x in a:

j = a.index(x)

s += str(j) + ' '

a[j] += 1

if s == '':

print('Отсутствует')

else:

print(s[:-1])

-------------------------------------------------------------

a = []

b = ['']

c = []

while b[0] != 'end':

b = [i for i in input().split()]

if b[0] != 'end':

for i in range(len(b)):

c.append(int(b[i]))

a.append(c)

c = []

row = len(a)

col = len(a[0])

c = [[0]\*col for i in range(row)]

for i in range(row):

for j in range(col):

x1 = a[i-1][j]

x3 = a[i][j-1]

if j == col-1:

x4 = a[i][0]

else:

x4 = a[i][j+1]

if i == row-1:

x2 = a[0][j]

else:

x2 = a[i+1][j]

c[i][j] = x1 + x2 + x3 + x4

if j != col-1:

print(c[i][j],end=' ')

else:

print(c[i][j])

-------------------------------------------------------------------------------

n = 5

a = []

for i in range(n):

a.append([int(j) for j in range(n\*i+1,n\*i+1+n)])

print(a)

b = [n]

for l in range(1,n):

b.append(n - l)

b.append(n - l)

print(b)

c = []

k = 0

for i in b:

c.append([int(j) for j in range(1+k,i+1+k)])

k += i

print(c)

------------------------------------------------------------------------------

n = int(input())

a = [[0]\*n for i in range(n)]

b = []

i = 1

while len(b) < n:

b.append(i)

b.append(i)

i += 1

m = 1

n -= 1

l = b[n]

for k in range(0,l):

for j in range(0+k,n-k+1):

a[0+k][j] = m

m += 1

if k != l:

for i in range(1+k,n-k+1):

a[i][n-k] = m

m += 1

for j in range(n-1-k,0+k-1,-1):

a[n-k][j] = m

m += 1

for i in range(n-1-k,1+k-1,-1):

a[i][0+k] = m

m += 1

elif n0 % 2 == 0:

for i in range(1+k,n-k+1):

a[i][n-k] = m

m += 1

for j in range(n-1-k,0+k-1,-1):

a[n-k][j] = m

m += 1

for i in range(n+1):

for j in range(n+1):

if j == n:

print(a[i][j])

else:

print(a[i][j],end=' ')

**Функции**

def min2(a,b):

if a < b:

return a

else:

return b

m = min2(2,min2(3,4))

def min(\*a):

for i in a:

if i < m:

m = i

return m

m(2,3,4)

m(5)

m([2,3,4,5])

def f(n):

return n \* 10 + 5

f(f(f(10))) # print out: 10555

-----------------------------------------------------

def my\_range(start, stop, step=1):

res = []

if step > 0:

x = start

while x < stop:

res += [x]

x += step

elif < 0:

x = start

while x > stop:

res += [x]

x += step

return res

print(my\_range(2,5))

[2,3,4]

print(my\_range(2,15,3))

[2,5,8,11,14]

print(my\_range(15,2,-3))

[15,12,9,6,3]

print(my\_range(stop=15, start=5))

[5,6,7,8,9,10,11,12,13,14]

----------------------------------------------------

def f(x):

if x <= -2:

return 1-(x+2)\*\*2

elif x > 2:

return (x-2)\*\*2+1

else:

return -x/2

----------------------------------------------------

lst = [1, 2, 3, 4, 5, 6]

print(modify\_list(lst)) # None

print(lst)   # [1, 2, 3]

modify\_list(lst) print(lst) # [1]

lst = [10, 5, 8, 3]

modify\_list(lst)

print(lst) # [5, 4]

---------------------------------------------------------

def modify\_list(l):

i = 0

while i < len(l):

if l[i] % 2 != 0:

del l[i]

else:

l[i] //= 2

i += 1

-----------------------------------------------------

**Множества**

s = set()

s = {'a','a','b','c','d','d'}

print(s) # {'d', 'a', 'b', 'c'}

print('a' in s) # True

print('z' in s) # False

s.add(el)

s.remove(el)

s.discard(el) # удаляет если есть

s.clear(el)

len(s)

**Словари**

Словарь (dictionary) - ассоциативный массив, отображение (map). Ключи различны и неизменяемы, элементы не имеют порядка

d = dict()

d = {'a':239,'b':333, 10:1}

print(d) # {'a': 239, 'b': 333, 10: 1}

print(d['a']) # 239

print(d[10]) # 1

dict = {}

key in dict

key not in dict

dict[key] = value

dict.get(key)

del dict[key]

d.keys

d.values

d.items

---------------------------------------------------

for key in d:

print(key, end='')

for key in d.keys():

print(key, end='')

for value in d.values():

print(value, end='')

for key, value in d.items():

print(key, value, end ='; ')

-------------------------------------------------------

d = {a: a \*\* 2 for a in range(7)}

print(d) # {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36}

d = dict.fromkeys(['a', 'b'], 100)

print(d) # {'a': 100, 'b': 100}

d = dict([(1, [1,1]), (2, [4,4])])

print(d) # {1: [1, 1], 2: [4, 4]}

------------------------------------------------------

def update\_dictionary(d, key, value):

if key in d:

d[key].append(value)

elif 2 \* key in d:

d[2 \* key].append(value)

else:

d.update({2 \* key:[value]})

d = {}

update\_dictionary(d, 1, -1)

print(d) # {2: [-1]}

update\_dictionary(d, 2, -2)

print(d) # {2: [-1, -2]}

update\_dictionary(d, 1, -3)

print(d) # {2: [-1, -2, -3]}

-----------------------------------------------------

s = [i for i in input().split()]

d = {}

for i in s:

i = i.lower()

if i in d:

d[i] += 1

else:

d.update({i:1})

for j in d:

print(j,d[j])

-----------------------------------------------------

n = int(input())

d = {}

for i in range(n):

j = int(input())

if j not in d:

d.update({j:f(j)})

for j in d:

print(d[j])

------------------------------------------------------

**Файлы**

import os

os.path.join('C:','Python','filename.txt')

**--------------------------------------------------------------------**

inf = open('C:\Python\data.txt','r')

s1 = inf.readline()

s2 = inf.readline()

print(s1[:-1],s2[:-1]+'!',end='')

inf.close()

with open('C:\Python\data.txt','r') as inf:

s1 = inf.readline()

s2 = inf.readline().strip() # '\t abc \n' -> 'abc'

print(s1[:-1],s2+'!',end='')

with open('input.txt') as inf:

for line in inf:

line = line.strip()

print(line)

---------------------------------------------------------------------

ouf = open('output.txt','w') as ouf:

ouf.write('text\n')

ouf.write(str(25))

ouf.close

with open('output.txt','w') as ouf:

ouf.write('text\n')

ouf.write(str(25)) # всегда str для вывода чисел

---------------------------------------------------------------------

with open('C:\Python\data.txt','r') as inf:

ouf = open('C:\Python\out.txt','w')

for line in inf:

line = line.strip()

res = ''

m = ''

i = len(line) - 1

while i >= 0:

s = line[i]

if s in '0123456789':

m = s + m

else:

res = s \* int(m) + res

m = ''

i -= 1

ouf.write(res)

ouf.write('\n')

ouf.close()

--------------------------------------------------------------------------

with open('C:\Python\data.txt','r') as inf:

ouf = open('C:\Python\out.txt','w')

res = {}

for line in inf:

line = line.strip()

s = [i for i in line.split()]

for i in s:

i = i.lower()

if i in res:

res[i] += 1

else:

res.update({i:1})

m = sorted(res.values(),reverse = True)[0]

s = []

for k in res:

if res[k] == m:

s += [k]

ouf.write(sorted(s)[0] + ' ' + str(m) + '\n')

ouf.close()

**Модули**

https://docs.python.org/3/library

import sys

print(len(sys.argv))

import subprocess

subprocess.call(args, \*, stdin=None, stdout=None, stderr=None, shell=False, timeout=None)

import subprocess

subprocess.call(['python', '-h'])

import math

r = float(input())

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

import sys

s = ''

for i in sys.argv[1:]:

s += str(i) + ' '

print(s[:-1])

Anaconda Powershell Pro -> conda install requests

pypi.org

import requests

r = requests.get('http://example.com')

print(r.text)

import requests

url = 'http://example.com'

par = {'key1': 'val1', 'key2': 'val2'}

r = requests.get(url, params=par)

print(r.url) # http://example.com/?key1=val1&key2=val2

import requests

with open('C:\Python\data.txt') as inf:

url = inf.readline().strip()

r = requests.get(url)

lines = r.text.splitlines()

print(len(lines))

**Задачи**

n = int(input())

c = {}

for i in range(n):

g = [j for j in input().split(';')]

if g[0] not in c:

c.update({g[0]: [0,0,0,0,0]})

if g[2] not in c:

c.update({g[2]: [0,0,0,0,0]})

c[g[0]][0] += 1

c[g[2]][0] += 1

if g[1] > g[3]:

c[g[0]][1] += 1

c[g[0]][4] += 3

c[g[2]][3] += 1

elif g[1] == g[3]:

c[g[0]][2] += 1

c[g[0]][4] += 1

c[g[2]][2] += 1

c[g[2]][4] += 1

else:

c[g[2]][1] += 1

c[g[2]][4] += 3

c[g[0]][3] += 1

for k, v in c.items():

print(k + ':', end='')

for i in v[:-1]:

print(i, end=' ')

print(v[len(v) - 1])

-----------------------------------------------------------------------

a = input()

b = input()

s1 = input()

s2 = input()

dct1 = {}

dct2 = {}

s11 = ''

s21 = ''

for i in range(len(a)):

dct1.update({a[i]:b[i]})

dct2.update({b[i]:a[i]})

for i in s1:

s11 += dct1[i]

for i in s2:

s21 += dct2[i]

print(s11)

print(s21)

-------------------------------------------------------------------------

n = int(input())

s1 = []

for i in range(n):

s1.append(input().lower())

m = int(input())

s2 = [[i.lower() for i in input().split()] for j in range(m)]

s3 = set()

for i in range(m):

for j in s2[i]:

if j not in s1:

s3.add(j)

for i in s3:

print(i)

----------------------------------------------------------------------------

m = int(input())

dct = {'север': 0, 'запад': 0, 'юг': 0, 'восток': 0}

for i in range(m):

d, e = input().split()

dct[d.lower()] += int(e)

print(dct['восток'] - dct['запад'], dct['север'] - dct['юг'])

-------------------------------------------------------------------------------

with open('C:\Python\marks.tsv') as inf:

dct = {}

for line in inf:

print(line.strip().split())

cl, fam, tal = line.strip().split()

cl = int(cl)

tal = int(tal)

if cl not in dct:

dct.update({cl:[tal, 1]})

else:

dct[cl][0] += tal

dct[cl][1] += 1

for cl in range(1,12):

if cl not in dct:

print(cl, '-')

else:

print(cl, dct[cl][0] / dct[cl][1])

**Библиотеки для анализа данных**

NumPy

matplotlib

SciPy

pandas

Scikit-learn

PythonSeaborn

Anaconda powershell2

conda install numpy

from numpy import \*

a = array([2, 3, 4])

b = array([(1.5, 2, 3),(4, 5, 6)])

b.ndim # 2

b.shape # (2, 3)

b.size # 6

z = zeros((3, 2)) array([[0, 0, 0], [0, 0, 0], [0, 0, 0]])

a = arange(10, 30, 5) # array([10, 15, 20, 25])

linspace(0, 2, 9) # array([0. , 0.25, 0.5 , 0.75, 1. , 1.25, 1.5 , 1.75, 2. ])

b = arange(12) # array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])

b.reshape(4, 3)

# array([[ 0, 1, 2],

[ 3, 4, 5],

[ 6, 7, 8],

[ 9, 10, 11]])

a = array([10, 20, 30])

b = arange(3)

print(a + b)

print(a - b)

# [10 21 32]

[10 19 28]

a < 20 # array([ True, False, False])

a \*\* 2 # array([100, 400, 900], dtype=int32)

2 \* sin(a) # array([-1.08804222, 1.8258905 , -1.97606325])

from pylab import \*

x = linspace(0, 5, 10)

y = x \*\* 2

%matplotlib inline

figure()

plot(x, y, 'r')

xlabel('x')

ylabel('y')

title('f(x) = x2')

show()

**Объекты**

Объект это абстракция для данных

Данные - объекты и отношения между ними

Объект определяется идентификатором, типом и значением

x = [1, 2, 3]

y = x

print(id(x), id(y), id([1, 2, 3])) # 88488448 88488448 88443648

print(y is x) # True

print(y is [1, 2 ,3]) # False

x = [1, 2, 3]

y = x

x.append(4)

print(x) # [1, 2, 3, 4]

print(y) # [1, 2, 3, 4]

print(x.\_\_sizeof\_\_()) # 96

Переменная - это ссылка на объект

Идентификатор объекта - это уникальное целое число

x = [1, 2, 3]

y = x

y.append(4)

s = "123"

t = s

t = t + "4"

print(str(x) + " " + s) # [1, 2, 3, 4] 123

x=[1,2,3]  
y=x  
**y=y\*2**  
print(x)  
**[1, 2, 3]**  
print(y)  
**[1, 2, 3, 1, 2, 3]**  
  
НО:  
x=[1,2,3]  
y=x  
**y\*=2**  
print(x)  
**[1, 2, 3, 1, 2, 3]**  
print(y)  
**[1, 2, 3, 1, 2, 3]**

**Git - система управления версиями проектов**

**GitHub - соцсеть для хранения исходного кода**

git help

git clone

git add

git status

git commit -m ""

git push

git init

git branch

git merge

git checkout

test

Запускаем Git Bash

git init

git config -- global user.name "Kirill Blazheev"

git config -- global user.email "blazheev.k@gmail.com"

echo "# main" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/kblazheev/main.git

git push -u origin main

Репозиторий состоит из трех «деревьев». Первое «дерево»  - это **рабочая директория,** в которой хранятся актуальные файлы. Второе  -  это **index** или область подготовленных файлов. А еще есть **head**  -  указатель на ваш последний коммит.

Допустим, в вашем проекте есть папка new\_project. Вы можете перейти в нее из окна терминала и добавить локальный репозиторий. Это делается через следующую команду:

cd new\_project  
git init

В вашем проекте появилась новая скрытая директория с названием .git. Именно здесь Git хранит все, что ему нужно для отслеживания проекта. Теперь вы можете последовательно добавлять файлы в область подготовки:

git add <имя\_файла>

или добавьте сразу все файлы через:

git add .

Создать коммит с этими изменениями можно через команду:

git commit -m “<сообщение\_коммита>”

Проверить, есть ли изменения для отправки, можно в любое время по команде:

git status

Если изменения вас устраивают, напишите:

git push

и отправьте эти изменения в репозиторий.

При внесении изменений следует обновить и сами файлы:

git add <имя\_файла>

или

git add -- all

Создайте коммит, добавьте нужное сообщение и отправьте этот коммит в репозиторий.