**Программирование на языке Python**

#комментарий

“”” комментарий “””

Print(**'Hello world'**)

Тип определяет:

* Область допустимых значений
* Допустимые операции
* Объем памяти
* Формат хранения данных

a=int(input(**'Введите первое число'**))   
b=int(input(**'Введите второе число'**))   
c=a+b   
print(c)  
  
a=int(input(**'Введите первое число'**))   
b=int(input(**'Введите второе число'**))   
c=a+b   
print(str(a)+**'+'**+str(b)+**'='**+str(c))  
  
a\*\*b – возведение в степень   
a,b=b,a смена чисел местами  
  
Целочисленное деление (округление вниз)  
X=3//4 *#=0*X=a//b *#=1*X=-3//4 *#-1*

% - остаток от деления

D=85  
B=d//10  
A=d%10  
D=a%b  
D=b%a

a=int(input(**'Введите первое число'**))   
b=int(input(**'Введите второе число'**))   
m=a **if** a>b **else** b   
print(a)

n=10.57   
print(**'%d'** % n) - %d целое  
print(**'%5d'** % n) - %5d целое, записанное в поле из пяти символов  
print(**'%-5d'** % n) - %-5d целое, записанное в поле из пяти символов, но выравненное влево  
print(**'%05d'** % n) - %05d вместо пробелов – нули  
print(**'%5.1f'** %n) - %5.1f вещественное с плавающей запятой, с одним знаком после запятой, шириной в поле шириной пять знаков  
print(**'%.3f'** % n) - %.3f вещественное с плавающей запятой, с тремя знаками после запятой, в поле минимально необходимой ширины  
print(**'%e'** % n) - %e вещественное с плавающей запятой, записанное в научной нотации  
print(**'%E'** % n) %E то же что %e, для обозначения экспоненты используется E  
print(**'%11.3e'** %n)   
print(**'%.3e'** % n)

Стандартные функции

abs(x)  
int(x)  
**import** math  
math.pi – число пи  
math.sqrt(x) – квадратный корень  
  
**import** random   
x=random.randint(1,5)   
print(x)   
y=random.randint(1,5)   
print(y)  
  
**import** random **for** c **in** range(21):   
x=random.randint(1,5)   
print(x)  
  
a=int(input())   
b=int(input())   
m=a **if** a>b **else** b   
print(**'Макстмальное из чисел '**+str(a)+**' и '**+str(b)+**' = '**+str(m))

cost = 1500   
**if** cost < 1000:   
print(**'Скидок нет'**)   
**elif** cost <2000:   
print(**'Скидка 2%'**)   
**elif** cost < 5000:   
print(**'Скидка 5%'**)   
**else**:   
print(**'Скидка 10%'**)  
  
**for** k **in** range(1,11): с 1 по 11, шаг по умолчанию = 1  
print(k\*\*2)  
  
**for** k **in** range(10,0,-1): с 10 по 0, шаг = -1  
print(k\*\*2)  
  
i=0  
**while** i<10:  
 print(**'Привет!'**)  
 i+=1

Символьные строки

s=**'Привет!'**print(s)  
print(s[5]) – выведется буква ‘т’  
print(s[-2]) –выведется буква ‘т’

n=len(s) – длина строки

Задача. Заменить в веденной строке все буквы ‘a’ на буквы ‘б’

s=input(**'Введите строку'**)  
s1=**''  
for** i **in** range(0,len(s)):  
 **if** s[i]!=**'а'**:  
 s1=s1+s[i]  
 **else**:  
 s1=s1+**'б'**print(s1)

Операции со строками

Срезы:

s=**'123456789'**s1=s[3:8] включая 4 до включая 7

s=**'123456789'**s1=s[:8] с начала до включая 7

s=**'123456789'**s1=s[3:] с включая 4 до конца

s=**'123456789'**s1=s[::-1] реверс строки

s=**'123456789'**s1=s[::-2] с конца через один символ

s=**'123456789'**s1=s[3::2] вывод с 3 символа через один символ

Удаление:

s=**'123456789'**s1=s[:3] + s[9:]

Привести верхний/нижний регистр

s=**'aAbBcC'**s1=s.upper()  
s2=s.lower()

Проверка на цифры

s=**'aAbBcC'**

print(s.isdigit())  
s1=**'123'**print(s1.isdigit())

Проверка вхождения в строку

s=**'Здесь был Вася'**n=s.find(**'с'**)

s.replace(str1, str2) замена строки на строку

Сперва вносим полные данные человека, потом выводим его инициалы

s=input(**'Введите фамилию имя отчество'**)  
fio=s.split()  
print(fio[0]+**' '**+fio[1][0]+**'.'**+fio[2][0])

**Списки**

students=[**'Ivan'**,**'Masha'**,**'Sasha'**]  
**for** student **in** students:  
 print(**'Hello,'**,student+**'!'**)

Можно изменить конкретный элемент списка

students=[**'Ivan'**,**'Masha'**,**'Sasha'**]  
students[1]=**'Oleg'**print(students)

Доступ к элементам списка

students=[**'Ivan'**,**'Masha'**,**'Sasha'**]  
print(len(students))

Сложение списков

students=[**'Ivan'**,**'Masha'**,**'Sasha'**]  
teachers=[**'Oleg'**,**'Alex'**]  
print(students+teachers)

Умножение списков

students=[**'Ivan'**,**'Masha'**,**'Sasha'**]  
teachers=[**'Oleg'**,**'Alex'**]  
print(teachers\*3)

Методы списков

|  |  |
| --- | --- |
| метод | что делает |
| list.append(x) проще list += [x] | добавляет элемент в конец списка |
| list.expend(l) проще list += l | расширяет список list, добавляя в конец все элементы списка |
| list.insert(i, x) | вставляет на i-ый элемент значение x |
| list.remove(x) проще del list[i] | удаляет первый элемент в списке, имеющий значение x. valueerror, если такого элемента не существует |
| list.pop(i) | удаляет i-ый элемент и возвращает его. если индекс не указан, удаляется последний элемент |
| list.index(x,[start [, end]]) | возвращает положение первого элемента со значением x (при этом поиск ведется от start до end) |
| list.count(x) проще len(x) | возвращает количество элементов со значением x |
| list.sort() | сортирует список |
| list.reverse() проще list = list[::-1] | разворачивает список |
| list.clear() проще list = [] | очищает список |

Нахождение самого длинного слова

q=[]  
**for** i **in** range(5):  
 a=input(**'Слово '**)  
 q.append(a)  
q1=sorted(q, key=len)  
print(q1[len(q1)-1])

Генератор списков

print(list(range(10)))

a=[i **for** i **in** range(10)]  
print(a)

a=[i\*i **for** i **in** range(10)]  
print(a)

**from** random **import** randint  
a=[randint(20,100) **for** x **in** range(10)]  
print(a)

Массивы

data=input()  
s=data.split()  
a=[int(x) **for** x **in** s]

Так попроще

a = list(map(int, input().split()))  
print(a)

**for** i **in** range(N):  
 print(A[i], end=**' '**)  
  
**for** x **in** A:  
 print(x, end=**' '**)

**Процедуры**

Процедура – вспомогательный алгоритм, который выполняет некоторые действия

**def** Error():  
 print(**'Ошибка программы'**)  
n=int(input())  
**if** n<0:  
 Error()

**def** prover(ss):  
 a=[0,1,2,3,4,5,6,7,8,9]  
 **for** c **in** ss:  
 **if** c **in** str(a):  
 print(**'Неккоректные данные'**)  
 **break**s=input(**'Введите фамилию '**)  
prover(s)  
s1=input(**'Введите имя '**)  
prover(s1)  
s2=input(**'Введите отчество '**)  
prover(s2)

**def** prover(ss):  
 t=1  
 a=[0,1,2,3,4,5,6,7,8,9]  
 **for** c **in** ss:  
 **if** c **in** str(a):  
 t=0  
 **break  
 return** t  
  
**while True**:  
 s=input(**'Введите фамилию '**)  
 **if** prover(s)==1:  
 **break  
 else**:  
 print(**'Некорректные данные'**)

**def** prover(ss):  
 t=1  
 a=[0,1,2,3,4,5,6,7,8,9]  
 **for** c **in** ss:  
 **if** c **in** str(a):  
 t=0  
 **break  
 return** t  
  
A=[**'Фамилию '**,**'Имя '**,**'Отчество '**]

**for** i **in** A:  
 **while True**:  
 s=input(**'Введите '**+i)  
 **if** prover(s)==1:  
 **break  
 else**:  
 print(**'Некорректные данные'**)

В примерах выше при преобразовании нуля в str, он начинает пробел считать нулем. Поэтому, если ставить пробелы при вводе – работать не будет. Надо что-то придумать.

**Локальные и глобальные переменные**

a = 5  
**def** qq():  
 a = 1  
 print (a)  
qq()  
print (a) // Выведет 5 и 1

a = 5  
**def** qq():  
 print(a)  
qq() // выведет 5

a = 5  
**def** qq():  
 **global** a  
 a = 1  
qq()  
print(a) //выведет 1

**Функции**

Найти квадрат суммы цифр числа

**def** sum(n):  
 sum=0  
 **while** n!=0:  
 sum+=n%10  
 n=n//10  
 **return** sum  
a=int(input(**'Введите число '**))  
s=sum(a)\*\*2  
print(s)

**Python и MSQL**

Рабочий процесс программы python состоит в следующем:

* Подключение к серверу MySQL
* Создание новой базы данных
* Соединение с базой данных
* Выполнение sql запроса, сбор результатов
* Обработка, если в таблицу занесены значения
* Закрытие соединения с сервером MySQL

**from** getpass **import** getpass  
**from** mysql.connector **import** connect, Error  
  
**try**:  
 **with** connect(  
 host=**'localhost'**,  
 user=input(**'Имя пользователя: '**),  
 password=getpass(**'Пароль: '**)  
 ) **as** connection:  
 print(connection)  
**except** Error **as** e:  
 print(e)