Python-

Variable- eg:-

rent=1220

item1=”rent”

item2=”gas”

gas=202.5

print(rent)

total=rent+gas

print(total) or total

print(“expenses items: “ item1,item2)

naming convention- (same as c,c++)

can’t use keyword as a variable name

can’t use special character in variable name except\_

can’t start with a digit but can contain digit in between

Numbers in python-

Input-2+3

Output-5 and same for other arithmetic operator

Power- 3^2🡪 3\*\*2 output-9

Function- round(time,2) time=6.65865 after using function round- time =6.65

No precise way of store floating no-

eg:

6-5.7

0.2999999999999998

round(6-5.7,2)

0.3

String—

Strings are immutable- can’t chnge a specific character after created

text="ice cream"

text

'ice cream'

text[2]

'e'

text[0:3]

'ice'

text[4:]

'cream'

text[:3]

'ice'

Can use single or double quotes to store a string(if string conatins single quoted then use dounle quoted string to store it and vice verse)

Multiline string use three single quotes

address='''i am coder

from birth

india'''

address

'i am coder\nfrom birth\nindia'

print(address)

i am coder

from birth

india

concatenate two strings

s1="hello"

s2="world"

s1+' '+s2

'hello world'

Concatenate a string and a no- convert no to string than concatenate-

s1="hello"

s2="world"

s1+' '+s2

'hello world'

s="my weight is: "

s2=50

s+' '+str(s2)

'my weight is: 50'

LIST

Can access by using index stat with 0… access from last of list use negative sign.. items[-1] represent last element of list

For range use :

Len(items)

In- to find that element is present in list or not

Replace- directly add at the index using assignment operator… over ride

Items.append(“butter”)- adding butter at the last location of list

Items.insert(2,”veggis”)- insert veggis at 2 location

items=["brread","rice","pasta","veggis"]

items

['brread', 'rice', 'pasta', 'veggis']

items[0]

'brread'

items[0]="chips"

items

['chips', 'rice', 'pasta', 'veggis']

items[0:2]

['chips', 'rice']

items[2:]

['pasta', 'veggis']

items[-1]

'veggis'

items[-2]

'pasta'

items[-0]

'chips'

items.append("butter")

items

['chips', 'rice', 'pasta', 'veggis', 'butter']

items.insert(1,"bread")

items

['chips', 'bread', 'rice', 'pasta', 'veggis', 'butter']

Concatenate two list-

items1=["bread","idli","dosa"]

items2=["fruits","veggis","flowers"]

items=items1+items2

items

['bread', 'idli', 'dosa', 'fruits', 'veggis', 'flowers']

"dosa" in items

True

"pav" in items

False

len(items)

6

……………………………………………………

If condition :

Statement

Else :

Statement

and,or,==,<=,>=,<,>,!=,not,not true

input from user- use input keyword

num=input("enter a number ")  
num=int(num)  
if num%2==0:  
 print("the no is even")  
else:  
 print("no is odd")

num store in string format so in second line it converted into integer type

indian=["samosa","parantha","daal"]  
chinease=["egg role","fried rice","momos"]  
italian=["pizza","pasta","risotto"]  
dish=input("enter a dish name ")  
if dish in indian:  
 print("Indian")  
elif dish in chinease:  
 print("chinease")  
elif dish in italian:  
 print("italian")  
else:  
 print("don't know")

for loop-

exp=[3456,6543,2342,2345,1234,1234,1234]  
total=0  
for item in exp:  
 total=total+item  
print(total)

Range-

for i in range(1,10):  
 print(i)

exp=[3456,6543,2342,2345,1234,1234,1234]  
total=0  
for i in range(len(exp)):  
 print("Month",(i+1),exp[i])  
 total=total+exp[i]  
print("total expense is: ",total)

break statement

continue statement

while loop-

i=1  
while i<=5:  
 print(i)  
 i=i+1

functions

def calculate(exp):  
 total=0  
 for item in exp:  
 total=total+item  
 return total  
  
tom\_exp\_list=[120,1234,134,1223,23]  
joe\_exp\_list=[234,654,333,23]  
  
toms\_total=calculate(tom\_exp\_list)  
joe\_total=calculate(joe\_exp\_list)  
print("toms total expensive: ",toms\_total)  
print("joe total expensive: ",joe\_total)

positional argument- when calling any function total=sum(4,5)

named argument- at time of calling- total=sum(b=4,a=5)

default argument

in above both the cases function is – def sum(a,b)

local variable and global variable

“ “ “……….”””multi line Comments or documents string

# single line comment

Dictionary

Key value pair…. Order not matter

code example with output

d={"shivani":1234567894,"niks":789456123,"poo":456789321}

d

{'shivani': 1234567894, 'niks': 789456123, 'poo': 456789321}

d["poo"]

456789321

d["sonii"]=1234567895

d

{'shivani': 1234567894, 'niks': 789456123, 'poo': 456789321, 'sonii': 1234567895}

del d["sonii"]

d

{'shivani': 1234567894, 'niks': 789456123, 'poo': 456789321}

for key in d:

print("key",key,"value:",d[key])

key shivani value: 1234567894

key niks value: 789456123

key poo value: 456789321

"shivi" in d

False

"shivani" in d

True

d.clear()

d

{}

Tuple-list of valued together,all values have different meaning and in list all values have same meaning

Immutable

Example- address(city name,pincode,state),point(x-cordinate,y-cordinate)

point=(5,9)

point[0]

5

point[1]

9

Eg: tuple is imutable

point[1]=3

Traceback (most recent call last):

File "<pyshell#18>", line 1, in <module>

point[1]=3

TypeError: 'tuple' object does not support item assignment

Module is a way to reuse a code written by someone else

import math

math.sqrt(16)

4.0

math.pow(2,5)

32.0

dir(math)

['\_\_doc\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'isqrt', 'lcm', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'nextafter', 'perm', 'pi', 'pow', 'prod', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc', 'ulp']

math.log10(100)

2.0

math.floor(2.3)

2

math.ceil(2.3)

3

import calendar

cal=calendar.month(2016,1)

print(cal)

January 2016

Mo Tu We Th Fr Sa Su

1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 31

dir(calendar)

['Calendar', 'EPOCH', 'FRIDAY', 'February', 'HTMLCalendar', 'IllegalMonthError', 'IllegalWeekdayError', 'January', 'LocaleHTMLCalendar', 'LocaleTextCalendar', 'MONDAY', 'SATURDAY', 'SUNDAY', 'THURSDAY', 'TUESDAY', 'TextCalendar', 'WEDNESDAY', '\_EPOCH\_ORD', '\_\_all\_\_', '\_\_builtins\_\_', '\_\_cached\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', '\_colwidth', '\_locale', '\_localized\_day', '\_localized\_month', '\_monthlen', '\_nextmonth', '\_prevmonth', '\_spacing', 'c', 'calendar', 'datetime', 'day\_abbr', 'day\_name', 'different\_locale', 'error', 'firstweekday', 'format', 'formatstring', 'isleap', 'leapdays', 'main', 'mdays', 'month', 'month\_abbr', 'month\_name', 'monthcalendar', 'monthrange', 'prcal', 'prmonth', 'prweek', 'repeat', 'setfirstweekday', 'sys', 'timegm', 'week', 'weekday', 'weekheader']

How to write our own python module:

same directory-

function.py- define any no of functions

another file me use as:- import function and then use of function name use them

when function as sub directory of any directory

import modules.function as f

where modules is directory function is sub directory

file name hello.py

def calculate\_triangle\_area(base,height):  
 return 1/2\*(base\*height)  
  
def calculate\_square\_area(length):  
 return length\*length

file name function.py

import hello  
area=hello.calculate\_square\_area(5)  
print(area)

when hello file is in system

import sys

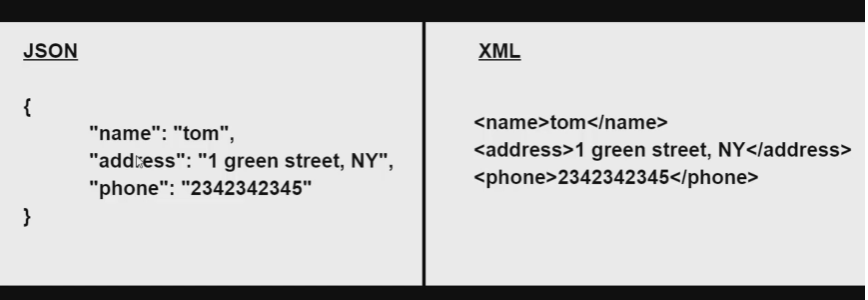
sys.path.append(“C:/code”)

import hello as h

……………………….

JSON-java script object notation

Lightweight as compared to XML



book={}  
book['tom'] = {  
 'name': 'tom',  
 'address': 'red street',  
 'phone': 4563217895  
}  
book['jerry'] = {  
 'name': 'jerry',  
 'address': 'black street',  
 'phone': 45454457895  
}  
book['joe'] = {  
 'name': 'joe',  
 'address': 'green street',  
 'phone': 454797895  
}  
  
  
import json  
s=json.dumps(book)  
print(s)

output-{"tom": {"name": "tom", "address": "red street", "phone": 4563217895}, "jerry": {"name": "jerry", "address": "black street", "phone": 45454457895}, "joe": {"name": "joe", "address": "green street", "phone": 454797895}}

import json  
s=json.dumps(book)  
with open("c://data//book.txt","w")as f:  
 f.write(s)

a book.txt file is created in data folder and above output is stored in that file

JSON exchange format-use any language to read this file

f=open("c://data//book.txt","r")

s=f.read()

s

'{"tom": {"name": "tom", "address": "red street", "phone": 4563217895}, "jerry": {"name": "jerry", "address": "black street", "phone": 45454457895}, "joe": {"name": "joe", "address": "green street", "phone": 454797895}}'

import json

book=json.loads(s)

book

{'tom': {'name': 'tom', 'address': 'red street', 'phone': 4563217895}, 'jerry': {'name': 'jerry', 'address': 'black street', 'phone': 45454457895}, 'joe': {'name': 'joe', 'address': 'green street', 'phone': 454797895}}

type(book)

<class 'dict'>

book['joe']

{'name': 'joe', 'address': 'green street', 'phone': 454797895}

book['joe']['phone']

454797895

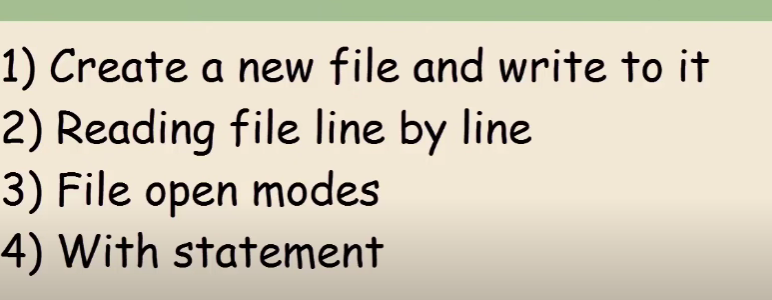
for person in book:

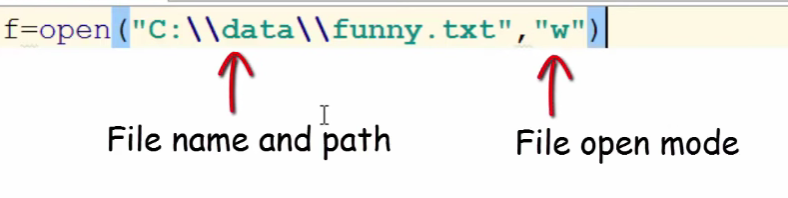
print(book[person])

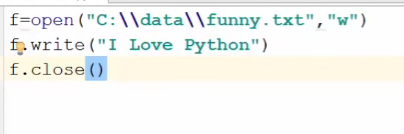
{'name': 'tom', 'address': 'red street', 'phone': 4563217895}

{'name': 'jerry', 'address': 'black street', 'phone': 45454457895}

{'name': 'joe', 'address': 'green street', 'phone': 454797895}

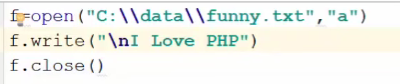






In w mode and do changes than it override the file

So for add something use append mode



f=open("c://data//funny.txt","r")  
print(f.read())  
f.close()

C:\Users\asus\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\asus\PycharmProjects\pythonProject\hello.py

teacher:why are u late?

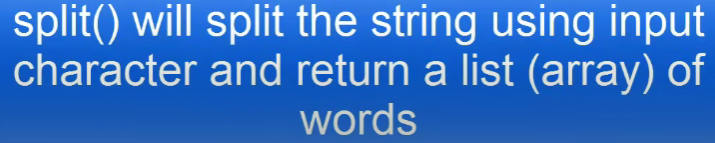
frank: because of the sign

teacher: what sign:

frank: the one that says "school ahead, fo slow"

….read line by line…

f=open("C://data//funny.txt","r")  
for line in f:  
 print(line)  
  
f.close()



f=open("C://data//funny.txt","r")  
for line in f:  
 tokens=line.split(' ')  
 print(str(tokens))  
  
f.close()

output-(without str output will be the same)

['teacher:why', 'are', 'u', 'late?\n']

['frank:', 'because', 'of', 'the', 'sign\n']

['teacher:', 'what', 'sign:\n']

['frank:', 'the', 'one', 'that', 'says', '"school', 'ahead,', 'fo', 'slow"']

f=open("C://data//funny.txt","r")  
for line in f:  
 tokens=line.split(' ')  
 print(len(tokens))  
  
f.close()

4

5

3

9

f=open("C://data//funny.txt","r")  
f\_out=open("C://data//funny\_wc.txt","w")  
for line in f:  
 tokens=line.split(' ')  
 f\_out.write("wordcount:"+str(len(tokens))+line)  
  
  
f.close()  
f\_out.close()

create a file funny\_wc-

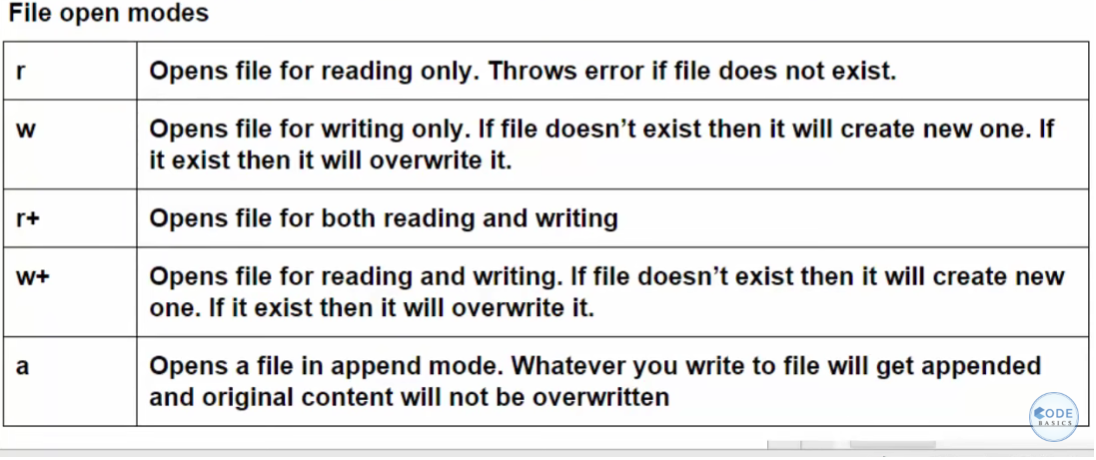
wordcount:4teacher:why are u late?

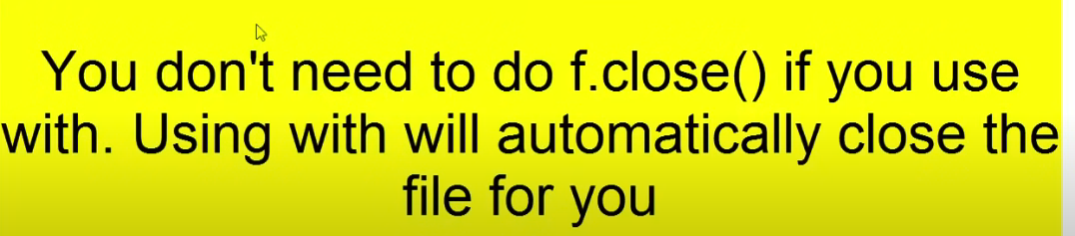
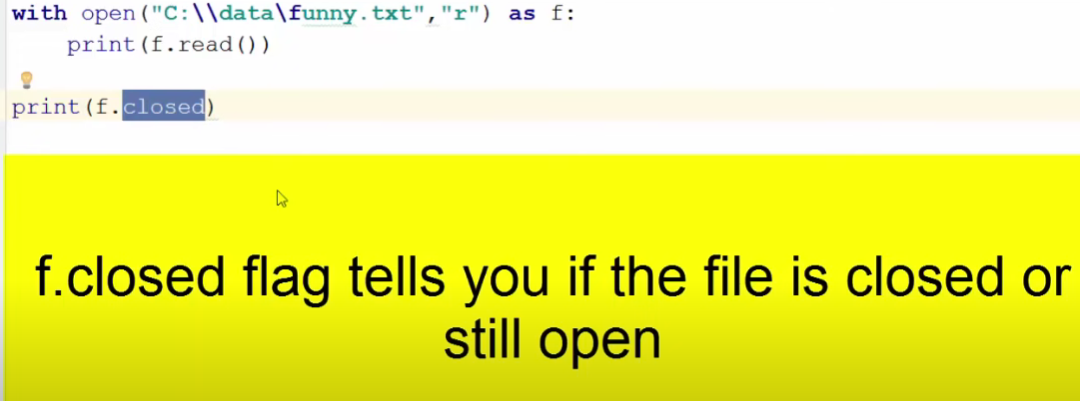
wordcount:5frank: because of the sign

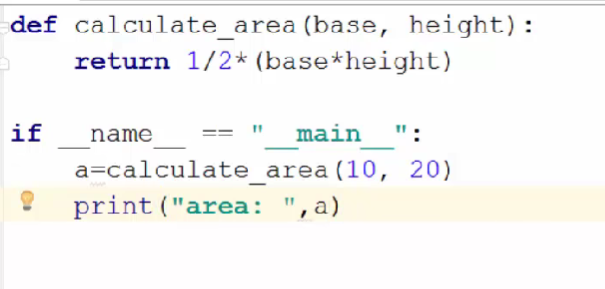
wordcount:3teacher: what sign:

wordcount:9frank: the one that says "school ahead, fo slow"

file opening modes……………….

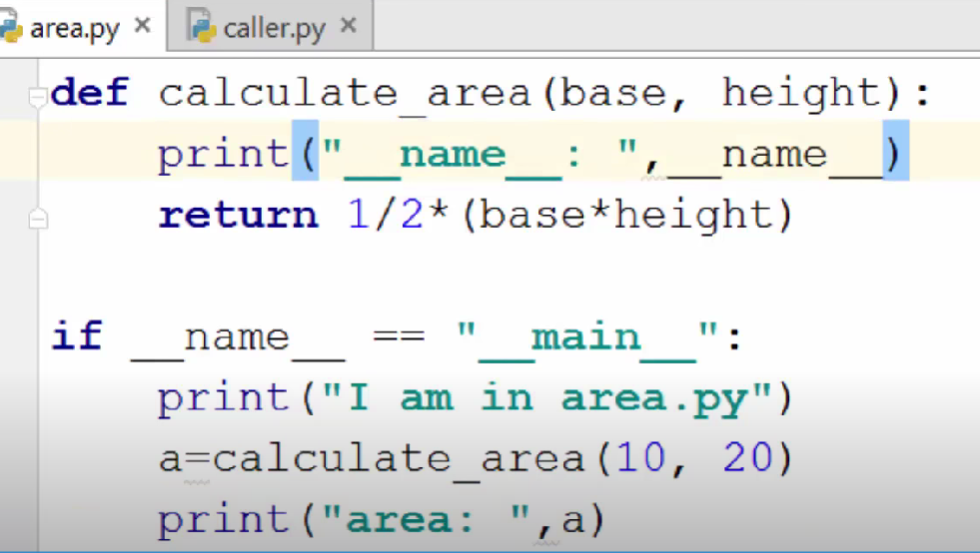
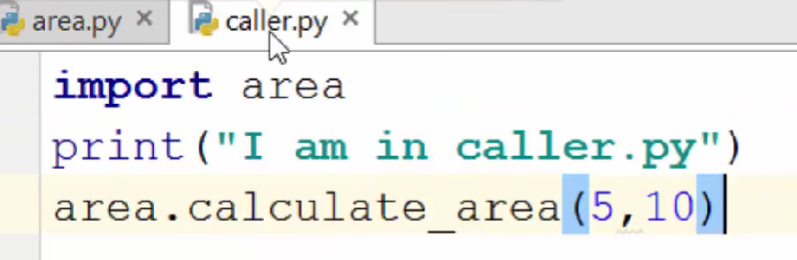
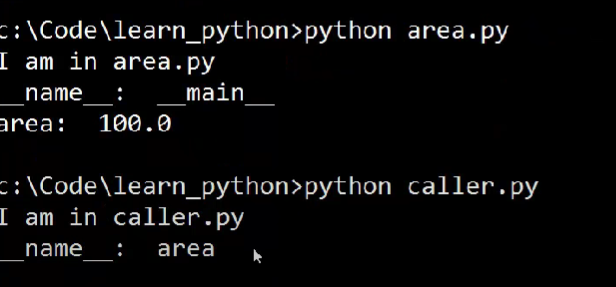




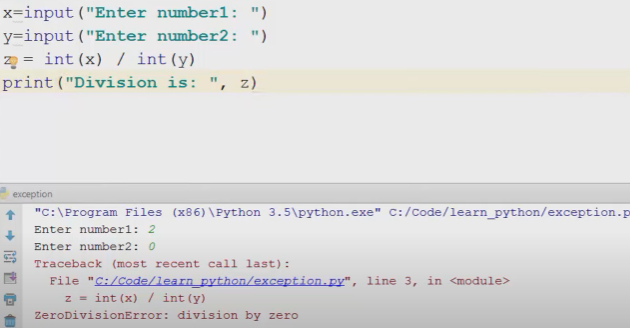
If \_\_name\_\_ == “\_\_main\_\_”: use for entry point in python like main function in c++;

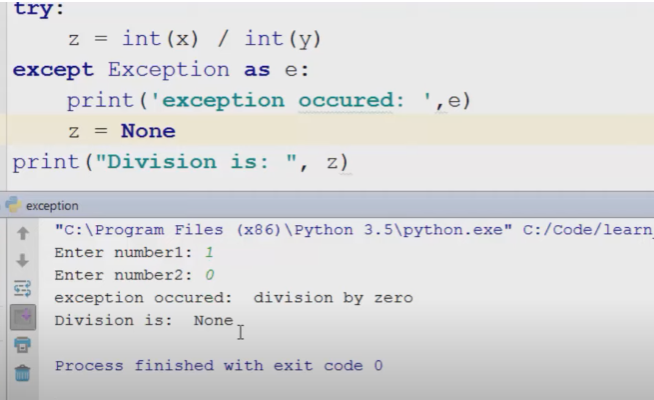
Value of \_\_name\_\_ is changes example-

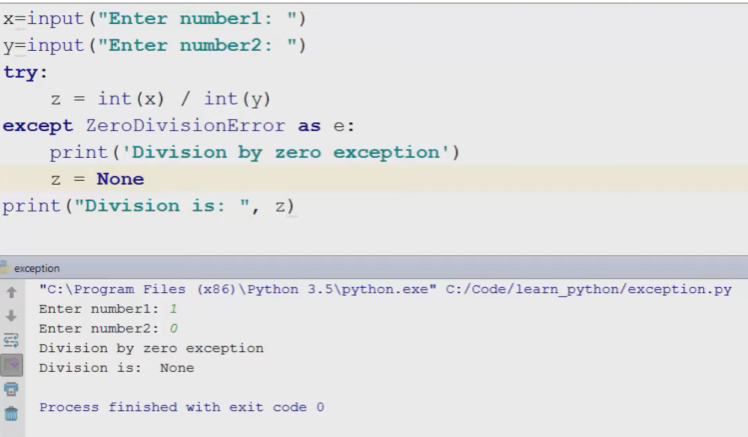
  

EXCEPTION HANDLING

Division by zero





Concatenate a string and a number