Variables & Datatypes:

Data Types

* Strings
* Numbers (common data types)
* Boolean values

Example:

A black background with white text

Description automatically generated

Working with Strings:

* Plain Text
* Escape character
* New line
* Backslash
* Concatenation
* Functions (these are used to modify strings and get information about strings)
* Index functions (passing value to the function is called as parameter)

Working with Numbers:

* Modular operators are used to the dividend of the number.
* A function is a collection of bunch of code which does something

User Inputs:

* Using of integers and float

Mad Libs Game:

* Python program has been written by passing stings and numbers as user inputs to create a story.

Lists

* List functions
* Extend
* Pop
* Remove

Tuples:

* Code examples and details are written in .py file

**Functions:**

* Code examples and details are written in .py file

**Return Statements:**

* Return statement is used to print the the output of the executed code in function
* It prints the output of the code base till the “return” line written in function.

Example:

In line “8” the print statement was ignored by return statement as it is returning output from line “6”

A screen shot of a computer code

Description automatically generated

Output

A computer screen shot of a black screen

Description automatically generated

**IFstatements:**

* Code examples and details are written in .py file

IFstatements & Comparisions:

* Code examples and details are written in .py file

Building a better calculator:

* Code examples and details are written in .py file

**Dictionaries:**

* Dictionaries are used to store the Key Value Pairs and it is defined in curly braces {}

**Example:**

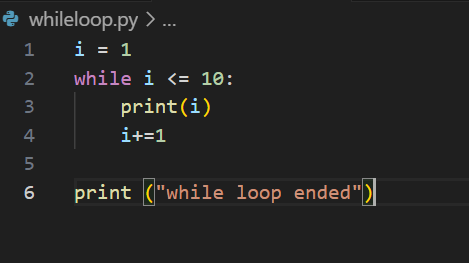


**Whileloop:**

* This is used to run piece of code in a loop until the defined condition is true, once the condition fails it get out of the loop.
* Here we used “shorthand operators **+= & -=”** for executing below example

**Note:** Refer this blog post for detailed explanation of “shorthand operators” [**https://ioflood.com/blog/equals-plus-or-minus-symbol/**](https://ioflood.com/blog/equals-plus-or-minus-symbol/)in simple words i+=1 equals to i= i+1

**Example**



**Output**



**Guessing Game:**

* Code examples and details are written in .py file

**Code example:**

A computer screen shot of a program code

Description automatically generated

**Forloop:**

This is used to iterate with in a loop using a variable dealing with all the functions like lists, arrays, Booleans...etc.

Code example:

#forloop

for alphabets in "panjagala chinna veera bhadrudu":

    print(alphabets)

#defining array now

friends = ["ragahava", "prashanth", "krishna", "mani", "viswa"]

for name in friends:

    print(name)

#now lets try print numbers by defining range in loop

for numbers in range(10):

    print(numbers)

#now lets try to print numbers by defining range between two numbers

for numbers in range(3, 10):

    print (numbers)

# now lets try to use length array

friends = ["ragahava", "prashanth", "krishna", "mani", "viswa"]

print(len(friends))

for index in range(len(friends)):

    print(friends[index])

# using if conditions within for loop

for numbers in range(10):

    if numbers == 0:

        print("first Iteration")

    elif numbers == 1:

        print ("second iteration")

    else:

        print (numbers)

Exponent Function:

Code:

#print(2\*\*3)-----> this means 2 to the power or 3 that is 2\*2\*2 = 8

#defining a function to mimic exponential functionality using user input

base\_num = input('enter base number:')

pow\_num = input('enter power number:')

def exponential(base\_num, pow\_num):

    result = 1

    for expo in range(int(pow\_num)):

        result = result\*int(base\_num)

    return result

print(exponential(base\_num, pow\_num))

output:

enter base number:2

enter power number:10

1024

**2d Lists and Nested For loop:**

# here we are gonna trying to learn handling 2 dimention values using list

number\_grid = [

    [0,1,2],

    [3,4,5],

    [6,7,8],

    [9]

]

#print(number\_grid[<row value><column value>]) to access particular value, lets try to access value 8.

print (number\_grid[2][2])

#now let try to access all the values using nested for loop

for row in number\_grid:

    for col in row:

        print(col)

output:

8

0

1

2

3

4

5

6

7

8

9

**Building Translator:**

Here we build a translator program that do if any letter in phrase has vowel converts that letter in to letter “g”

#defining  function to convert vowels in phrase to "g"

def translate(phrase):

    translation = ""

    for letter in phrase:

        if letter.lower() in "aeiou":

            if letter.isupper():

                translation = translation + "G"

            else:

                translation = translation + "g"

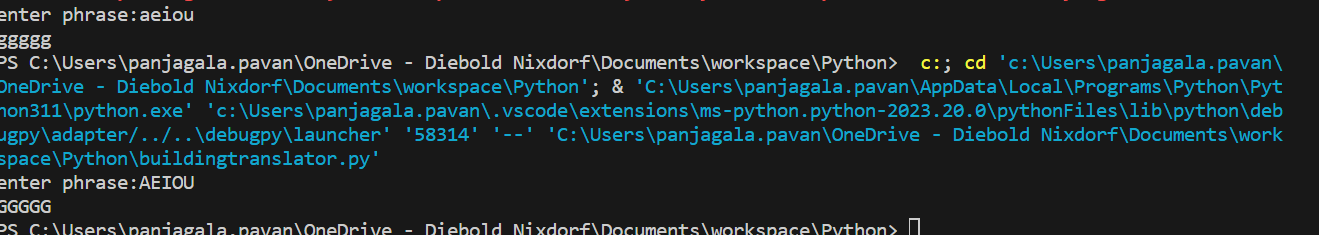
        else:

            translation = translation + letter

    return translation

print(translate(input("enter phrase:")))

output:



**Comments:**

Here we discuss about comments declaration, we can provide comments in our code base by using below two syntax:

* Line should start with “#”
* For multiple lines commenting use

‘’’<commentsline1>

<commentsline2>

<commentsline3>’’’

#single line commment

'''multiple line comments

line1

line2

line3

'''

print("printed comments above")

Output:



**Try/Except:**

Is something you try to execute code under **try block** and if it fails then catch that failure in **except block** instead of breaking the program execution.

#using try/Except block

try:

    division = 10/0

    number = float(input("enter a number: "))

    print(number)

    #here in except blocks we are trying to catch exact error by using expected errro function "ZeroDivisionError/ValueError" and trying to print the actual error

except ZeroDivisionError as err:

    print(err)

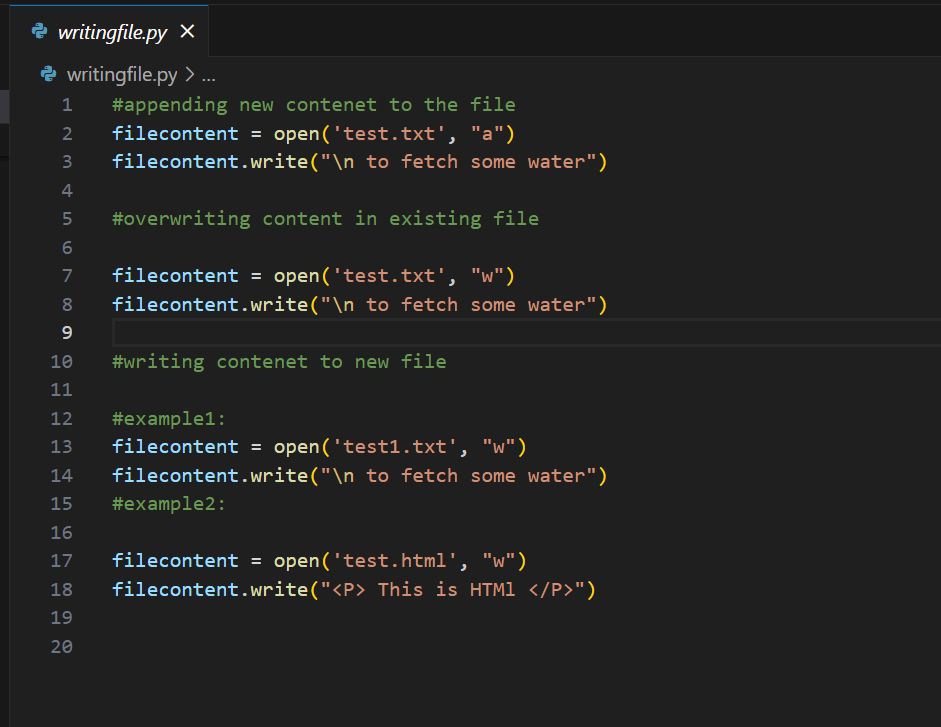
except ValueError as err:

    print(err)

**Writing/Reading/Appending to a file:**

A screenshot of a computer program

Description automatically generated



**Modules and PIP:**

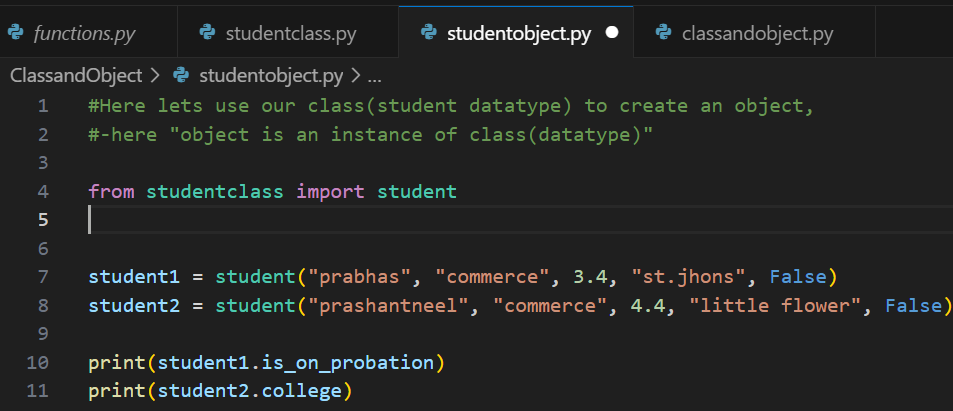
Modules in python are predefined set of python program like “PrettyTable” will be available for the specific use developed by some person, we have multiple python modules that we can make use of(<https://docs.python.org/3/tutorial/modules.html>).

Pip is package installer tool in python (<https://pypi.org/project/pip/>)

**Class and object:**

A computer screen shot of a program

Description automatically generated



A screen shot of a computer program

Description automatically generated

**Building a multiple-choice quiz(using class or object):**

**Example 1:**

#creating an questions array

questions\_prompt = [

    "1. who is the father of our nation India \n\n (a)Mahatma Gandhi (b)Subhash Chandrabose \n\n",

    "2. India's first citizen \n\n (a)Prime Minister of India (b)President of India \n\n"

]

#lets create question class

class question:

    def \_\_init\_\_(self, prompt, answer):

        self.prompt = prompt

        self.answer = answer

#now lets create an another array to access question and answer using "question" class defined above

questions = [

    question(questions\_prompt[0],"a"),

    question(questions\_prompt[1],"b")

]

# now let define a funtion to run our questionpaper

def test\_run(questions):

        score = 0

        for question in questions:

            answer = input(question.prompt)

            if answer == question.answer:

                score += 1

        print("you got " + str(score) + "/" + str(len(questions)) + " are correct")

test\_run(questions)

**example 2:**

#creating a question array

#from multiprocessing.connection import answer\_challenge

questions\_prompt = [

    "1. who is the hero in the Bahubali \n\n (a)Rana (b)Prabhas (c)Yash \n\n",

    "2. who is the president for India \n\n (a)Droupadi Murmu (b) Ramnath kovind (c) Narendra Modi \n\n",

    "3. who is the Chief Minister of Telangana State \n\n (a)K ChandraSekar Rao (b)A Revanth Reddy (c)Bhatti Vikramarka \n\n"

]

#from questionclass import question

# here defining question class(datatype)

class question:

    def \_\_init\_\_(self, prompt, answer):

        self.prompt = prompt

        self.answer = answer

#now creating one more array

questions = [

    question(questions\_prompt[0], "b"),

    question(questions\_prompt[1], "a"),

    question(questions\_prompt[2], "b")

]

#defining a funtion to run the test

def test\_run(questions):

    score = 0

    for question in questions:

        answer = input(question.prompt)

        if answer == question.answer:

            score += 1

    print("you got " + str(score) + "/" + str(len(questions)) + " are correct")

test\_run(questions)

Object/class functions:

Using function under class, below screenshot for reference(using honor\_roll function)

A computer screen shot of a program code

Description automatically generated

A computer screen with text and images

Description automatically generated

Output:

A black background with white text

Description automatically generated

Inheritance:

Here we learn about how we can using inheritance parent class with in child class

Parent class:

A screen shot of a computer program

Description automatically generated

Child Class:

In below screen shot you can observe that we are inheritance concept by calling parent class in child class.

A computer screen shot of a computer code

Description automatically generated

**Object file:**

Here we are passing parent class “chef()” to “chefdetails” and then trying to print the target function value and similar with child class “indianchef()”

**Note:** any function value we are trying to access via child class the value we get is parent class function value until we have defined similar function with new value in child class, which will override parent class function value.

A screen shot of a computer program

Description automatically generated

Output:

