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:**

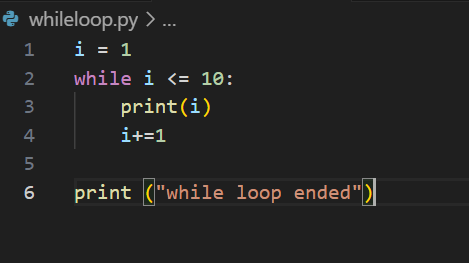


**Whileloop:**

* 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"

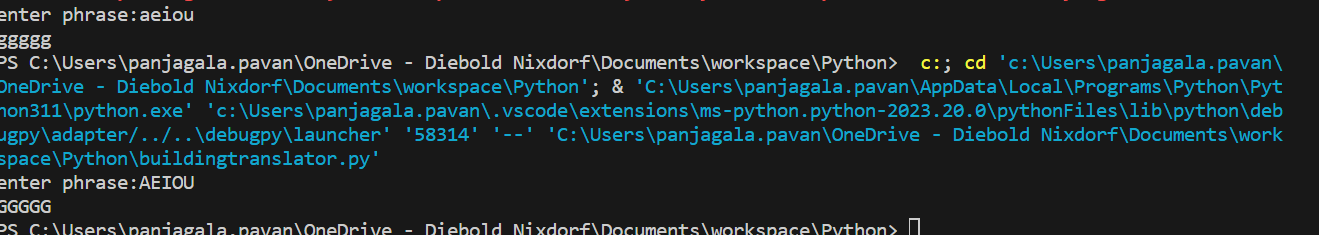
        else:

            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

