Name:-Yash solanki
Reg email id:-yash87015@gmail.com
Course name:-Data analytic
Assignment name:-python basic assignment
Submission date:-not declare
Git link:-https://github.com/Yash87015/YASH
(in git hub use assignment folder or these file name)

python basic assignment

August 11, 2024

1 python basic assignment

1.1 Question 1 Explain the key feature of python that make it a popular choice for programming?

1.1.1 Ans

(1) python is very easy to learn and huge active community also python have a lot of libraries around 137000.

(2)python easy to use machine language , data scientist and data industry etc.. There are lot of alsouse in other industry like automation, image process, etc…

(3)python is interpreted language , dynamically typed and free and open source programing language. above all of the reasons and key feature of python that make it a popular choice for programming

1.2 Question 2 Describe the role of predefined keywords in Python and provideexamples of how they are used in a program?

1.2.1 Ans

```
[13]: help('keywords')
```

```
Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

1.2.2

above all of the world is predefiend key world and their are different type of uses that i explain below

first of all lets run simple program

```
[4]: a="i love python" #print statement one of the predefind keyworld that use print␣
      ↪the output that i asiign the varible. variable means [a]
     print(a)
```

```
i love python
```

```
[12]: #hence its prove True and False is bool value and also we use type as a␣
       ↪predefiend keyworld to see which value we assign in variable
      #example 2.3
      a = True
      type(a)
```

```
[12]: bool
```

```
[ ]:
```

1.2.3

now let see other predefined keyword and their uses in program

first of all if , else , True , False , and , or , break , not , continue, in , while these are the keyworld and their uses of program
(1) if, elif, else :-those are conditional statement that we can use when ever we have multiple choice in decision making there we can use if,elif and else statement lets try with example those above statement use to creat google form to collect information from users and also use gread assigning system in education that we can use in example. {example 6 in question no . 6}

(2)True and False also in, not, is or statement :- True and False is bool value. when ever we use identity operator(example 4.1) and membership operator(example 4.2) there we can use as a output true and false. these statement is True value indicate 1 and False value indicate 0 those are also seein example. Also see how to use those statement in pythone programming language.

(3)while and for :-those are loop statement its using for creating a loop problem in multiple choices . while we can also used of continue and break statement in there problem there we can see exampleof each in { example 7 in question no 7}

```
[4]: a is b #her we use is statement
```

```
[4]: False
```

```
[5]: a is not b #here we use is and not stament in combine and ther is output is␣
      ↪true or false
```

[5]:
```
True
```

[104]:
```
#hence its prove these statement is True value indicate 1 and False value indicate 0 those are also see in example
#Also use and statement and or statement as a predefiend keyworld
#example 2.3
True and False
```

[104]:
```
False
```

[105]:
```
True or True
```

[105]:
```
True
```

[5]:
```
5 == 3
```

[5]:
```
False
```

1.2.4

those are the the predefiend keyworld in python and here we see 2 example how we can use keyworld in program also other. we can see in next quations

[ ]:
```

```

1.3 Question 3 Compare and contrast mutable and immutable objects in Python with examples ?

1.3.1 Ans

1.3.2

Mutable:-Object whose state or value can be changed after they are created are called mutable object on container mutable object we can say list is a mutable object

Immutable: Object whose state or value can not be changed after they are created are called im-mutable object on container we can say tuple is immutable object string also immutable objet

lets see with example

[107]:
```
#mutablity #hence its prove list is a mutable object
list1=[1,2,3,"ajay",4,"bijay"]
list1[0]=100
print(list1)
```

```
[100, 2, 3, 'ajay', 4, 'bijay']
```

[16]:
```
#immutablity #hence its prove tuple is immutable object
a=(2,4)
a[0]=5
```

```
print(a) #it will give you erorr
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 3
      1 #immutablity #hence its prove tuple is immutable object
      2 a=(2,4)
----> 3 a[0]=5
      4 print(a) #it will give you erorr

TypeError: 'tuple' object does not support item assignment
```

1.4 Question 4 Discuss the different types of operators in Python and provide examples of how they are used?

1.4.1 Ans

1.4.2

there are 7 different type of operator in python

(1)arithmetic operator (2)comparison operator (3)logical operator (4)bitwise operator (5)assignment operator (6)identity operator (7)membership operator

1.4.3

(1) arithmetic operator:-this operator are used basic mathematical operation on number example :- additional(+),substraction(-),multipication(*),division(/),modular(%),exponention(**),and floor division(//)

[114]:

```
#these are simple addition opration example we can also do other operation in␣
  ↪arithmatic operator
a = 56
b = 797
a+b
```

[114]:

```
853
```

1.4.4

(2)comparison operator:-These operator compare two values . Equal(==),Not equal(!=),Greater then(>),Less then(<),Greater than equal to (>=), Less then eual to(<=)

[6]:

```
#these are simple 2 example greater than eual to and less than eual to . the␣
  ↪output return value is boolean
10<=5
```

[6]:

```
False
```

[7]:

```
10>=5
```

[7]:

```
True
```

1.4.5

(3)logical operator:-These are used to combined multiple coditions.as output retrn in Boolean value And(if both condition is true then value output is true otherwise always false.) Or(if one of the condition is true then value output is true otherwise false.) Not(if not true=false or not false = true.)

[9]:

```
#her are some example
True and False
```

[9] :

```
False
```

[10]:

```
True or False
```

[10] :

```
True
```

[11]:

```
not False
```

[11] :

```
True
```

1.4.6

(4)Bitwise operator:These operator work at the binary level and used for manuplating bits. (1.1)Bitwise AND(&):-Compare each bit of the first operand to the corresponding bit of the secound operand and return 1 if both bits are 1.

(1.2)Bitwise OR(|):-Compare each bit of the first operand to the corresponding bit of the secound operand and return 1 if at least one of the bits is 1.

(1.3)Bitwise XOR(^):-Compare each bit of the first operand to the corresponding bit of the secound operand and return 1 if the bits are different.

(1.4)Bitwise NOT(~):-invert the bits of the operand.

(1.5)Bitwise shift operator:- there are two type of bitwise operator {1} left shift operator(«) ,{2}right shift operator(»)

{1.5.1}Left shift operator(«) :- shift the bits of the first operand to the left by the number of positions specified by the second operand.(left shift operator adding last 3 bit 0 from left in operation) {1.5.2}Right shift operator(»):-shift the bits of the first operand to the right by the number of positions specified by the second operand.(Right shift operator remove last 3 ditit from left)

Above all the opeartor we see each one example

[1]:

```
#Bitwise AND(&)
5 & 4 #0101 & 0100 = 0100 hence its prove 4
```

[1]: 4

[2]:
```
#Bitwise OR(|)
5 | 4 #0101 | 0100 = 0101 hence its prove 5
```

[2]: 5

[3]:
```
#Bitwise XOR(^)
5 ^ 4 #0101 ^ 0100 = 0001
```

[3]: 1

[8]:
```
#Bitwise NOT(~)
~5 # 0101 it gives -0110
```

[8]: -6

[9]:
```
#Left shift operator(<<)
5 << 4 #0101 adding last 4 times 0 its output is 01010000 hence its prove 80
```

[9]: 80

[11]:
```
#Right shift operator(>>)
5 >> 1 #0101 remove last 1 digit the output is 010 hence its prove 2
```

[11]: 2

1.4.7

(5)assignment operator:-These operator assign value to variable.

Assign(=),Add and assign(+=),Subtract and assign(-=),Multiply and assign(*=),Divide and assign(/=),Modular and assign(%=),Exponetiation and assign(**=),Floor division and assign(//=),Bitwise And and assign(&=),Bitwise or and assign(|=),Bitwise xor and assign(^=),Left shift and assign(«=),Right shift and assign(»=).

[25]:
```
#Assign(=),Left shift and assign(<<=)  these  example show using two assignment operator other
#also we can do with same type of example only change the sign
x = 5
x <<= 1
print(x)
```

```
10
```

1.4.8

(6)identity operator:-These operator check if two variable refer to the same object in memory. is , is not these two operator use as identity operator

[28]:

```
#example 4.1
#identity operator >> compare the location of the two object/variable
x=6
y=5
x is y
```

[28]:

```
False
```

[29]:

```
x is not y
```

[29]:

```
True
```

1.4.9

(7)Membership operator:-These operator check if a valur is present in a sequance in a sentence.(such as a list,string or tuple)

[30]:

```
#example 4.2
# membership operator  #here we use in statment as predefiend keyworld
a = "jk upadyay sir"
"j" in a
```

[30]:

```
True
```

[31]:

```
"k" not in a
```

[31]:

```
False
```

1.5 Quastion 5 explain the concept of type casting in Python with examples.

1.5.1 Ans

1.5.2

Type casting:-whenever we take input from user it take all input as a string we can convert thant input in interger,flote etc that conversion called type casting or type conversion type casting are two type (1) implicit (2) explicit what we above we done its called explicit implicit means python are understand what type of data type its called implicit at other like java does not understand

[48]:

```
number = input()        #here you can see we take a input from user then aotomatically convert string to integer that called type casting
print(type(number))
print(type(int(number)))
```

```
 45
<class 'str'>
<class 'int'>
```

[53]:
```
#float to  integer
b=6.90000
type(b)
flout_value = int(b)
type(flout_value)
```

[53]:
```
int
```

[52]:
```
#integer to string
a = 50
type(a)
b  = str(a)
type(b)
```

[52]:
```
str
```

1.5.3

above we see 3 example of type casting and concept of type casting

[ ]:
```

```

1.6 Question 6 How do conditional statements work in Python? Illustrate with examples

1.6.1 Ans

1.6.2

conditional statement :-if, elif , else those are conditional statement that we can use when ever we have multiple choice in decision making there we can use if,elif and else (and nested if) statementlets try with example

#those above statement use to create google form to collect information from users and also use grad assigning system in education that we can use in example.

[54]:
```
#here we can use  if , else and elif keyworld to clarify that student is clever␣
 ↪,average ,good or topper by obtaied input from user [mark]
#example 6
student_obtained_mark = int(input())
if (student_obtained_mark   <= 40):
    print("just passing mark not good student need improvment")
elif (40 <= student_obtained_mark   <=50):
    print("good mark but below average student need improvment")
elif (50 <= student_obtained_mark   <= 80):
    print("average student but not clever student need improvment")
elif (80 <= student_obtained_mark   <=95):
    print("cleaver student but not topper need improvment")
elif (student_obtained_mark >= 95):
    print("topper student very good")
else:
    print("cleaver student")
```

```
 55
```

```
average student but not clever student need improvment
```

[55]:
```
#example no 6.2
#nested if else >> you can have multiple if else inside else
x = 10
y = 50
if x > 5:
    if y > 5:
        print("both x and y is greater than 5")
    else:
        print("x is greater than 5 but y is less than 5")
else:
    print("x is not greater than 5")
```

```
both x and y is greater than 5
```

[56]:
```
#use cases of conditional statment
#form fiilling
name = input("Please enter your name")
email = input("Please enter your email")
passworld=input("Please enter your passworld")

if name == "":
    print("please enter a valid name,can not be empty")
else:
    if "@" not in email:
        print("Please enter a valid email")
    else:
        if len(passworld) <= 0:
            print("No password entered , pls provide a valid password")
        else:
            print("Login/resigstration is sucessfull")
```

```
Please enter your name jk gupta
Please enter your email @hcwuh.com
Please enter your passworld sdjuuvnw123
Login/resigstration is sucessfull
```

1.7 Question 7 Describe the different types of loops in Python and their use cases with examples.

1.7.1 Ans

1.7.2

Loop statement :- loop statement it allows you to executed a block of code

repeatedly:loops are two type (1) while loop (2) for loop

(1)while loop:- repeatedly executed a block of code until a condition is met (2)for loops :- iterateover a sequence of element .string, list and for loops iterate over a sequence of element

(1.1)break:- This break statement use to exit the while loop (1.2)continue:-This continue statement skip the conditional variable and continue the while loop(skip the interaction)

[4]:

```
y = 10
k = 0
while k <= y:
    if k == 7:
        break
    print(k)
    k=k+1
else:
    print("number is higher then 10")
```

```
0
1
2
3
4
5
6
```

[3]:

```
for i in range(0,11):
    if i==4:
        continue
    print(i)
else:
    print("number is higher then 10")
```

```
0
1
2
3
5
6
7
8
9
```

```
10
number is higher then 10
```

[ ]:

```
#these are some use cases with example of while loop and for loop
```

[1]:

```
row = 1
while row < 5:
    col = 1
    while col <= row:
        print("*", end = " ")
        col = col+1
    print()
    row =row+1
```

```
*
* *
* * *
* * * *
```

[2]:

```
for i in range(5):
    for j in range(i+1):
        print(j+1,end = " ")
    print()
    j = i+1
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

[ ]: