14.Loops

May 5, 2020

Any Doubt

loop

[1]:
```
var = 1
while var <= 10:
    var += 1
    if var % 2:
        continue
    if var % 5 == 0:
        break
    print("Hello World Times: ", var)

else:
    print("Checking Your Knowledge about loops")
```

```
Hello World Times:  2
Hello World Times:  4
Hello World Times:  6
Hello World Times:  8
```

10 % (guide, assignment, doubt solving) –> 90 %

[2]:
```
while True:
    break
    print("hello")
```

[3]:
```
while True:
    print("hello")
    break
```

```
hello
```

[2]:
```
c = 1
while c <= 10:
    if c % 2 == 0:
        break
    print("Hello")
    c += 1
```

```
else:
    print("no else part if you break a loop")
```

```
Hello
```

[4]:
```
from random import randint
```

[6]:
```
randint(1, 10)
```

[6]:
```
3
```

```
userGuess > 50

Boundary Value Analysis

    49, 50, 51
```

Programming –>

```
Don't speak Show me your code
```

[12]:
```
import os
os.system('notepad')
```

[12]:
```
0
```

```
PEP --> Python Enhancement Proposals

PEP Sections

    PEP 8 --> Python Coding Style
```

[21]:
```
%%writefile guess_game.py
"""
Guess Game
this is a fun to guess a number between 1-50 correctly
"""
from random import randint
# random is module to generate random numbers
import os
# this is to clear screen usin cls command
os.system('cls') # to clear output screen
print("\n\n\n\n")
COMGUESS = randint(1, 50)
# a random number is guessed
CHANCES = 5
# initial 5 chances
while CHANCES: # False --> 0
    print(f"\nYou have left {CHANCES} chances.")
```

```
    USERGUESS = int(input("Your Guess[1, 50]: "))
    if USERGUESS > 50 or USERGUESS < 1:
        print("\nWarning!! Your Guess is Out of Range Please Guess b/w 1-50.")
        continue
    if USERGUESS > COMGUESS:
        print("\nHint: Your Guess is High Think a Lower Number.")
    elif USERGUESS < COMGUESS:
        print("\nHint: Your Guess is Low Think a Higher Number.")
    else:
        print("\n\nWholaa!!! You such a Master!! You have guessed Correctly.")
        break
    CHANCES -= 1
else:
    print("\n\nYou such a Looser")
    print("\nComputer Guess Was : ", COMGUESS)
```

```
Overwriting guess_game.py
```

```
python -m pip install pylint
```

[10]:
```
pwd
```

[10]:
```
'C:\\Batches\\Batch_7pm_online'
```

Sequences

```
string
list
tuple
dict
set
frozenset
```

[22]:
```
print("Hello World")
```

```
Hello World
```

[24]:
```
print(*"Hello World")
print("H", "e", "l", "l", "o", " ", "W", "o", "r", "l", "d")
```

```
H e l l o   W o r l d
H e l l o   W o r l d
```

[25]:
```
print(*"Hello World", sep='\n')
```

```
H
e
l
l
o
```

```
W
o
r
l
d
```

[27]:
```
print(*[1, 2, 'java'], sep='\n')
```

```
1
2
java
```

[28]:
```
print(*{'name':'sachin', 'age':23}, sep='\n')
```

```
name
age
```

[29]:
```
print(*{'name':'sachin', 'age':23}.values(), sep='\n')
```

```
sachin
23
```

[30]:
```
print(*{'name':'sachin', 'age':23}.items(), sep='\n')
```

```
('name', 'sachin')
('age', 23)
```

[34]:
```
print(*{1, 2, 1,23, 2,32, 3,  2,32}, sep='\n')
```

```
32
1
2
3
23
```

[35]:
```
lang = [ 'java', 'c', 'c++', 'ruby', 'perl', 'python']
```

[36]:
```
print(*lang, sep='\n')
```

```
java
c
c++
ruby
perl
python
```

[37]:
```
lang = [ 'java', 'c', 'c++', 'ruby', 'perl', 'python']

i = 0
while i < len(lang):
    print(lang[i])
    i += 1
else:
    print("iteration sucessfull")
```

```
java
c
c++
ruby
perl
python
iteration sucessfull
```

[39]:
```
lang = [ 'java', 'c', 'c++', 'ruby', 'perl', 'python']

i = 0
while i < len(lang):
    print(lang[i], end=' ')
    i += 1
else:
    print("\niteration sucessfull")
```

```
java c c++ ruby perl python
iteration sucessfull
```

For

Syntax

```
for value in iterable:
    st-1
    st-2
    st-3
    st-4
    ...
    st-n
```

[40]:
```
s = "python"
```

[42]:
```
for char in s:
    print(char)
else:
    print("okay all good")
```

```
p
y
t
h
o
n
okay all good
```

[43]:
```
for char in s:
    if char in [ 'p', 't']:
        continue
    print(char)
else:
    print("okay all good")
```

```
y
h
o
n
okay all good
```

[44]:
```
for char in s:
    if char in [ 'a', 'e', 'i', 'o', 'u']:
        break
    print(char)
else:
    print("okay all good")
```

```
p
y
t
h
```

[45]:
```
lang = [ 'java', 'c', 'c++', 'ruby', 'perl', 'python']
```

[46]:
```
for item in lang:
    print(item)
else:
    print("ok all good")
```

```
java
c
c++
ruby
perl
python
ok all good
```

to search item in list

[47]:
```
print(lang)
```

```
['java', 'c', 'c++', 'ruby', 'perl', 'python']
```

[49]:
```
"    hello          ".strip()
```

[49]:
```
'hello'
```

[52]:
```
key = input("language: ").strip().lower()

for item in lang:
    if item == key:
        print("Item Found in List.")
        break
else:
    print("Item not Found")
```

```
language:  JaVa
Item Found in List.
```

[53]:
```
key = "".join(input("language: ").strip().lower().split())
for item in lang:
    if item == key:
        print("Item Found in List.")
        break
else:
    print("Item not Found")
```

```
language:         j       A         V        a
Item Found in List.
```

[54]:
```
lang = [ 'java', 'c', 'c++', 'ruby', 'perl', 'python']
item = input().strip().lower()
if item in lang:
    print("Found")
else:
    print("Not Found")
```

```
java
Found
```

[55]:
```
d =  { 'name': 'sachin', 'age': 24, 'color': 'fair', 'country': 'India',
     'blood_group': 'B+ive'}
```

[57]:
```
for some in d:
    print(some)
else:
    print("\nTell me output")
```

```
name
age
color
country
blood_group

Tell me output
```

[58]:
```
for some in d.values():
    print(some)
else:
    print("\nTell me output")
```

```
sachin
24
fair
India
B+ive

Tell me output
```

[59]:
```
for some in d.items():
    print(some)
else:
    print("\nTell me output")
```

```
('name', 'sachin')
('age', 24)
('color', 'fair')
('country', 'India')
('blood_group', 'B+ive')

Tell me output
```

[61]:
```
print(d)
```

```
{'name': 'sachin', 'age': 24, 'color': 'fair', 'country': 'India',
'blood_group': 'B+ive'}
```

[65]:
```
for key in d:
    print(f"{key:>10} = {d[key]:<20} ")
else:
    print("\nTell me output")
```

```
      name = sachin
       age = 24
     color = fair
   country = India
```

```
blood_group = B+ive

Tell me output
```

[66]:
```
for pair in d.items():
    print(f"{pair[0]:>10} = {pair[1]:<20} ")
else:
    print("\nTell me output")
```

```
      name = sachin
       age = 24
     color = fair
   country = India
blood_group = B+ive

Tell me output
```

[76]:
```
l = [ ('java', 'sachin'), ('python', 'vijay'), ('c', 'tanvi'), ] #('a','b','c')]
```

[77]:
```
a,b = ('b','a') # tuple unpacking
print(a, b)
```

```
b a
```

[78]:
```
for lang, name in l:
    print(f"{name} Knows {lang} Very Well.\n")
```

```
sachin Knows java Very Well.

vijay Knows python Very Well.

tanvi Knows c Very Well.
```

[79]:
```
a,b = (1, 2, 3 )
```

```
      ␣
↪---------------------------------------------------------------------------

       ValueError                                Traceback (most recent call␣
↪last)

       <ipython-input-79-b71a25f0f0c9> in <module>
   ----> 1 a,b = (1, 2, 3 )


       ValueError: too many values to unpack (expected 2)
```

[ ]:

[80]:
```
inventory = [

    [ 'iphone11 pro', 35000, 75000],

    [ 'samsung s20', 30000, 70000],

    [ 'one plus', 25000, 50000]

]
```

\ is used to line skip

[84]:
```
for phone, buy_price, sale_price in inventory:
    s =f"We bought {phone} at price {buy_price} and by selling it at price␣
 ↪{sale_price} \
by earning profit of {sale_price-buy_price}."
    print(s, end='\n\n\n')
```

```
We bought iphone11 pro at price 35000 and by selling it at price 75000 by
earning profit of 40000.


We bought samsung s20 at price 30000 and by selling it at price 70000 by earning
profit of 40000.


We bought one plus at price 25000 and by selling it at price 50000 by earning
profit of 25000.
```

some other built-in iterables objects are :

- **range**
- **enumerate**
- **zip**

[85]:
```
s = "hello \
world"
```

[86]:
```
print(s)
```

```
hello world
```

range() it's a number line slicing

syntax:

```
range(end_value) --> will generate numbers starting from 0 to end_value - 1
eg. range(5) --> 0, 1, 2, 3, 4

range(start, end) --> will generate numbers starting from start value till end value - 1
eg. range(1, 10) --> 1, 2, 3, 4, 5, 6, 7, 8, 9,

range(start, end, step) --> will generate number from start to end - 1 have jump of step

eg. range(1, 10, 3) --> 1, 4, 7,
```

[87]:
```
x = range(5) #
print(x)
```

```
range(0, 5)
```

[88]:
```
print(*range(5))
```

```
0 1 2 3 4
```

[89]:
```
print(*range(1, 10))
```

```
1 2 3 4 5 6 7 8 9
```

[90]:
```
print(*range(1, 10, 3))
```

```
1 4 7
```

[92]:
```
num = int(input("Enter a number: ")) # 5
print(*range(num, num*10+1, num), sep='\n')
```

```
Enter a number: 5
5
10
15
20
25
30
35
40
45
50
```

[93]:
```
print(*range(10, 1, -2), sep='\n')
```

```
10
8
6
```

```
4
2
```

[94]:
```
"sachin "* 10
```

[94]:
```
'sachin sachin sachin sachin sachin sachin sachin sachin sachin sachin '
```

[100]:
```
nrows = int(input("Enter number of rows: "))
for var in range(1, nrows+1):
    print(f"{var:>2}. {'*'*var}")
else:
    print("\n\nwhat do you say about it ?")
```

```
Enter number of rows: 10
 1. *
 2. **
 3. ***
 4. ****
 5. *****
 6. ******
 7. *******
 8. ********
 9. *********
10. **********


what do you say about it ?
```

[101]:
```
for var in range(5, 55, 5):
    print(var)
```

```
5
10
15
20
25
30
35
40
45
50
```

enumerate

[103]:
```
print(*enumerate('python'), sep='\n')
```

```
(0, 'p')
(1, 'y')
(2, 't')
```

```
(3, 'h')
(4, 'o')
(5, 'n')
```

[105]:
```
print(*enumerate(['java', 'c', 'c++', 'ruby']), sep='\n')
```

```
(0, 'java')
(1, 'c')
(2, 'c++')
(3, 'ruby')
```

[106]:
```
lang = [ 'java', 'c', 'ruby', 'python', 'perl', 'c++']
for i, item in enumerate(lang):
    print(i, item)
```

```
0 java
1 c
2 ruby
3 python
4 perl
5 c++
```

[108]:
```
d = { 'name':'sachin', 'age': 24, 'country': 'india'}
for i, pair in enumerate(d.items()):
    print(f"{i+1}. {pair[0]:>10} = {pair[1]}")
```

```
1.       name = sachin
2.        age = 24
3.    country = india
```

[109]:
```
d = { 'name':'sachin', 'age': 24, 'country': 'india'}
for i, (key, value) in enumerate(d.items()):
    print(f"{i+1}. {key:>10} = {value}")
```

```
1.       name = sachin
2.        age = 24
3.    country = india
```

Functions in Python

[ ]:
```

```