Basic Programming in Python

3. Session: Loops

Nohayr Muhammad Abdelmoneim

Summer Term 2023

May 8th, 2023

Overview

- Recap and tips
- Lists
- Introduction to loops
- For loops
- While loops
- Nested loops

Tips

- Directory dir

```
In [1]: import math
        dir(math)

Out[1]: ['__doc__',
         '__file__',
         '__loader__',
         '__name__',
         '__package__',
         '__spec__',
         'acos',
         'acosh',
         'asin',
         'asinh',
         'atan',
         'atan2',
         'atanh',
         'ceil',
         'comb',
         'copysign',
         'cos',
         'cosh',
         'degrees',
         'dist',
         'e',
         'erf',
         'erfc',
         'exp',
```

- help

```
help(str.replace)

Help on method_descriptor:

replace(self, old, new, count=-1, /)
    Return a copy with all occurrences of substring old replaced by new.

      count
        Maximum number of occurrences to replace.
        -1 (the default value) means replace all occurrences.

    If the optional argument count is given, only the first count occurrences are
    replaced.
```

Cont. Tips

- Autocomplete (Tab)

```
x="Hello"
x.
```

```
capitalize
casefold
center
count
encode
endswith
expandtabs
find
format
format_map
```

- Documentation (Shift+tab)

```
str.replace
```

```
Signature: str.replace(self, old, new, count=-1, /)
Docstring:
Return a copy with all occurrences of substring old replaced by new.
```

```
str.replace
```

```
Signature: str.replace(self, old, new, count=-1, /)
Docstring:
Return a copy with all occurrences of substring old replaced by new.

  count
    Maximum number of occurrences to replace.
    -1 (the default value) means replace all occurrences.

If the optional argument count is given, only the first count occurrences are
replaced.
Type:      method_descriptor
```

Recap on Lists

- Lists are type of collections in Python, they are used to store multiple items in one variable.
- Elements of a list can be of different types.
- The indexing of elements starts with 0.
- You can also use the len() function on lists.

```
list1=["Red","Yellow","Green"]
list2=[2,6,7,10]
list3=[1.5,"Car",True,12,"T"]
print(list1,len(list1))
print(list2,len(list2))
print(list3,len(list3))
```

```
['Red', 'Yellow', 'Green'] 3
[2, 6, 7, 10] 4
[1.5, 'Car', True, 12, 'T'] 5
```

Loops

- Sequential execution:
  All statements are executed once, in order, no exceptions.
- Conditional execution:
  Some statements may or may not be executed depending on a certain condition.
- Loops:
  Some statements maybe executed more than one time depending on certain factors/conditions

Loops

Types of loops:

- For in :
  Executes certain statement(s) for each element in a given variable/object.

- For in range:
  Executes certain statement(s) repeatedly for a given number of iterations in a given range

- While loops:
  Executes certain statement(s) repeatedly until the condition is no longer true

Example

To check whether each letter in a string is a digit

```
str1="ab2r3"
if str1[0].isdigit():
    print(str1[0]," is a digit")
if str1[1].isdigit():
    print(str1[1]," is a digit")
if str1[2].isdigit():
    print(str1[2]," is a digit")
if str1[3].isdigit():
    print(str1[3]," is a digit")
if str1[4].isdigit():
    print(str1[4]," is a digit")
```

```
2  is a digit
3  is a digit
```

For loop

- For in

```
x=["January", "February", "March"]
for i in x:
    print(i)
```

```
January
February
March
```

- Enumerate

```
x=["January", "February", "March"]
for i,n in enumerate(x):
    print(i,n)
```

```
0 January
1 February
2 March
```

- Zip

```
x=["January", "February", "March"]
y=["Januar", "Februar", "März"]
for l,m in zip(x,y):
    print(l,m)
```

```
January Januar
February Februar
March März
```

Range

- End

```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

- Start, end

```
for i in range(2,8):
    print(i)
```

```
2
3
4
5
6
7
```

- Start, end, step

```
for i in range(2,10,2):
    print(i)
```

```
2
4
6
8
```

Example revisited

To check whether each letter in a string is a digit

- Using for in

```
str1="ab2r3"
for i in str1:
    if i.isdigit():
        print(i," is a digit")
```

```
2  is a digit
3  is a digit
```

- Using for in range

```
str1="ab2r3"
for i in range(len(str1)):
    if str1[i].isdigit():
        print(str1[i]," is a digit")
```

```
2  is a digit
3  is a digit
```

While loop

- While loops are not limited by a certain number or elements, but rather on a condition.

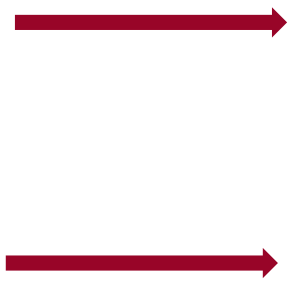
```
i=0
while i<6:
    print(i)
    i+=1
```

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

```
answer=input("Enter your answer ")
while answer!="no":
    print(answer)
    answer=input("Enter your answer ")
```

```
Enter your answer yes
yes
Enter your answer car
car
Enter your answer blue
blue
Enter your answer no
```

Important notes on While loop



```
str1="ab2r3"
i=0
while i<len(str1):
    if str1[i].isdigit():
        print(str1[i]," is a digit")
    i+=1
```

```
2  is a digit
3  is a digit
```

- Initialization value for condition

```
str1="ab2r3"
while i<len(str1):
    if str1[i].isdigit():
        print(str1[i]," is a digit")
    i+=1
```

- Changing value for condition

```
str1="ab2r3"
i=0
while i<len(str1):
    if str1[i].isdigit():
        print(str1[i]," is a digit")
```

Important notes on While loop

```
i=0
while i<6:
    print(i)
    i+=1
```

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

```
answer=input("Enter your answer ")
while answer!="no":
    print(answer)
    answer=input("Enter your answer ")
```

```
Enter your answer yes
yes
Enter your answer car
car
Enter your answer blue
blue
Enter your answer no
```

Nested loops

- Nested loops is using a loop within another loop
- Each iteration of the outer loop executes all iterations of the inner loop

```
list1=["Red","Yellow","Green"]
list2=[1,2,3,4]
for i in list1:
    for j in list2:
        print(i,j)
```

```
Red 1
Red 2
Red 3
Red 4
Yellow 1
Yellow 2
Yellow 3
Yellow 4
Green 1
Green 2
Green 3
Green 4
```

```
list1=["Red","Yellow","Green"]
list2=[1,2,3,4]
for i in list1:
    print(i)
    for j in list2:
        print(j)
```

```
Red
1
2
3
4
Yellow
1
2
3
4
Green
1
2
3
4
```

Examples on Nested loops

```
list1=["a2b3","abc4","2r34"]
for i in list1:
    for j in i:
        if j.isdigit():
            print(j,"is a digit")
```

```
2 is a digit
3 is a digit
4 is a digit
2 is a digit
3 is a digit
4 is a digit
```

```
list1=["a2b3","abc4","2r34"]
for i in list1:
    for j in i:
        if j.isdigit():
            print(j,"in",i,"is a digit")
```

```
2 in a2b3 is a digit
3 in a2b3 is a digit
4 in abc4 is a digit
2 in 2r34 is a digit
3 in 2r34 is a digit
4 in 2r34 is a digit
```

QUESTIONS?