Basic Programming in Python

3. Chapter: Lists and Loops

Lecturer: Daniel Weinhardt

Summer Term 2024

Slides were created by Nohayr Muhammad Abdelmoneim
Thank you very much for sharing!

Overview

- Tips
- Lists
- Introduction to loops
- For loops
- While loops
- Nested loops

Tips

Autocomplete (Tab) get Documentation

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

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

```
help(str.replace)
```

help(methodname)

```
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.
```

```
str.replace
```

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

Ctrl + mouse cursor
or
Shift + Tab

```
str.replace
```

```
Signature: str.replace(self, old, new, co
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
```

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
```

Example

Until now you heard of if statements
How do you check whether each letter in a string is a digit?

Example

Until now you heard of if statements
How do you 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
```

Loops - Introduction

- 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 may be executed more than one time depending on certain factors/conditions

Loops - Types of loops

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

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

```
Var = ["Hello", "World", "Python", "is", "Awesome"]

for i in Var:
    print(i)
```

✓ 0.0s

```
Var = ["Hello", "World", "Python", "is", "Awesome"]
end_token = "Awesome"
condition = True
i = 0
while condition:
    print(Var[i])
    condition = Var[i] != end_token
    i += 1
```

✓ 0.0s

Output for each option:

```
Hello
World
Python
is
Awesome
```

Loops - More looping strategies in for-loops

- *range(Int)*
  Picks one integer at a time given by a stop criterium

```
Var = ["January", "February", "March"]
length_var = len(Var)
for i in range(length_var):
    print(Var[i])
```

✓ 0.0s

- *enumerate(Var)*
  Picks one element at a time and its respective index

```
Var = ["January", "February", "March"]
length_var = len(Var)
for i, e in enumerate(Var):
    print(i, e)
```

✓ 0.0s

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

- *zip(Var1, Var2)*
  Pairs to sequences of equal length and picks the elements at the same indexes

```
Var1 = ["January", "February", "March"]
Var2 = ["Januar", "Februar", "März"]
for e1, e2 in zip(Var1, Var2):
    print(e1, e2)
```

✓ 0.0s

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

Loops - the range-statement

- *range(end)*
  Starts automatically at 0 and ends at *end*

```
for i in range(5):
    print(i)
```
✓ 0.0s
```
0
1
2
3
4
```

- *range(start, end)*
  Starts at *start* and ends at *end*

```
for i in range(2,5):
    print(i)
```
✓ 0.0s
```
2
3
4
```

- *range(start, end, step)*
  Start at *start*, ends at *end* and has a step size of *step*

```
for i in range(2,10,3):
    print(i)
```
✓ 0.0s
```
2
5
8
```

Loops - Example revisited

Can we check more easily whether an element of a string is a digit than with consecutive, hardcoded if-statements?

Using only if-statements:

```
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
```

Using *for in Var*:

```
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(Int)*:

```
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.

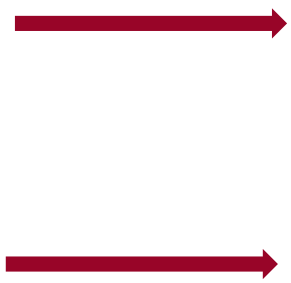
```
i=0
while i<6:
    print(i)
    i+=1
```

```
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
```

QUESTIONS?