Lecture 15

Topics to be covered:

Chapter 9:
- For loops
- Counting using the range() function
- While vs. for loops

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for variable in container:
    # Loop body
    # Loop body

# Statements to execute after the for loop
# is complete
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

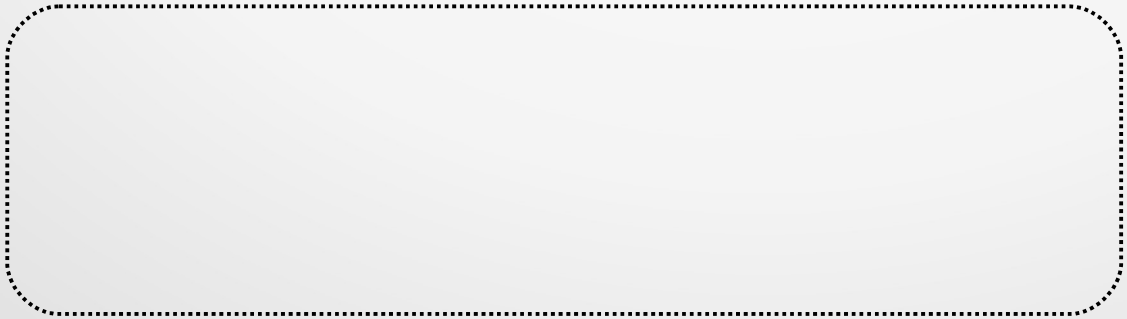
```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```



For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40, 70,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40, 70,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    →print(2*item, end=", ")
print("finished!")
```

20, 40, 70, 86,

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40, 70, 86,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    →print(2*item, end=", ")
print("finished!")
```

20, 40, 70, 86, 112,

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40, 70, 86, 112,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40, 70, 86, 112, 180,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:
    print(2*item, end=", ")
print("finished!")
```

```
20, 40, 70, 86, 112, 180,
```

For loops

Another type of loop in Python

for loop: a counted loop, executes a block of code n times;
Iterates over the elements in a container

```
for item in [10,20,35,43,56,90]:

    print(2*item, end=", ")

print("finished!")
```

20, 40, 70, 86, 112, 180,
finished!

For loops

Another type of loop in Python

Example:

```
myD = { 172: "Friday", 823: "Tuesday", 564: "Monday", 923: "Saturday", 435: "Wednesday", 712: "Sunday", 384: "Thursday"}

for k in myD:
    print("%d corresponds to %s" % (k,myD[k]))
print("finished!")
```

For loops

Another type of loop in Python

Example:

```
myD = { 172: "Friday", 823: "Tuesday", 564: "Monday", 923: "Saturday", 435: "Wednesday", 712: "Sunday", 384: "Thursday"}

for k in myD:
    print("%d corresponds to %s" % (k,myD[k]))
print("finished!")
```

```
172 corresponds to Friday
823 corresponds to Tuesday
564 corresponds to Monday
923 corresponds to Saturday
...
```

`For` loops

Example: Assume that we are given a list of decimal values in a Python list, named `data`. And we are asked to write a program that finds the average of all the values in the list `data`.

`For` loops

Example: Assume that we are given a list of decimal values in a Python list, named `data`. And we are asked to write a program that finds the average of all the values in the list `data`.

Here is an order of operations to do:

- add the values from list `data` one by one to a variable `s`,
- `count` the number of values in the list `data`,
- get the average by dividing `s` by `count`.

For loops

Example: Assume that we are given a list of decimal values in a Python list, named `data`. And we are asked to write a program that finds the average of all the values in the list `data`.

Here is a program for that:

```
s = 0
count = 0
for val in data:
   s += val
   count += 1
average = s / count

print("The average of the values in the list",
data,"is", average)
```

For loops

Example: Assume that we are given a list of decimal values in a Python list, named `data`. And we are asked to write a program that finds the average of all the values in the list `data`.

Here is a program for that:

```
s = 0
count = 0
for val in data:
    s += val
    count += 1
average = s / count

print("The average of the values in the list",
data,"is", average)
```

How can we simplify the program?

hint: I don't need to count the number of values in the list `data`

For loops

Example: Assume that we are given a list of decimal values in a Python list, named `data`. And we are asked to write a program that finds the average of all the values in the list `data`.

Here is a program for that:

```
s = 0
count = 0
for val in data:
    s += val
    count += 1
average = s / count
print("The average of the values in the list",
data,"is", average)
```

How can we simplify the program?

hint: I don't need to count the number of values in the list `data`

```
use len
```

For loops

Example: Assume that we are given a list of decimal values in a Python list, named `data`. And we are asked to write a program that finds the average of all the values in the list `data`.

Here is a program for that:

```
s = 0
for val in data:
   s += val
average = s / len(data)

print("The average of the values in the list",
data,"is", average)
```

`For` loops

In-class work:

Do **exercise 1** from the handout

For loops

Example: Now let's consider a small database with names associated with the e-mail addresses:

```
contactInfo = {
"Mark Huggard" : "Mhuggard@org.com",
"Alice True" : "ATrue@org2.com",
"Hunter O'Brien" : "HOBrien@org.com",
"Jane Cole" : "JaneCole@org6.com",
"Frank Dove" : "FrankDove@org7.com"
}
```

For loops

Example: Now let's consider a small database with names associated with the e-mail addresses:

```
contactInfo = {
"Mark Huggard" : "Mhuggard@org.com",
"Alice True" : "ATrue@org2.com",
"Hunter O'Brien" : "HOBrien@org.com",
"Jane Cole" : "JaneCole@org6.com",
"Frank Dove" : "FrankDove@org7.com"
}
```

Now, let's write the code fragment to:
1) extract all e-mail addresses from `contactInfo` and store them in a Python list, named `emails`, then
2) display alphabetically sorted list of e-mails.

For loops

Example: Now let's consider a small database with names associated with the e-mail addresses:

```
contactInfo = {
"Mark Huggard" : "Mhuggard@org.com",
"Alice True" : "ATrue@org2.com",
"Hunter O'Brien" : "HOBrien@org.com",
"Jane Cole" : "JaneCole@org6.com",
"Frank Dove" : "FrankDove@org7.com"
}

emails = []
for k in contactInfo: # extract all e-mail addresses
   emails.append(contactInfo[k])
emails.sort() # sort alphabetically
print(emails) # display
```

Counting using the `range()` function

`range()` function

Range	Generated sequence	Explanation
range(5)	0 1 2 3 4	every integer from 0 to 4
range(0, 5)	0 1 2 3 4	every integer from 0 to 4
range(3, 7)	3 4 5 6	every integer from 3 to 6
range(10, 13)	10 11 12	every integer from 10 to 12
range(0, 5, 1)	0 1 2 3 4	every 1 integers from 0 to 4
range(0, 5, 2)	0 2 4	every 2 integers from 0 to 4
range(5, 0, -1)	5 4 3 2 1	every 1 integer from 5 to 1
range(5, 0, -2)	5 3 1	every 2 integers from 5 to 1

Counting using the `range()` function

Example: Prices of various items change with time. Let's write the program that will allow us to calculate the price of the smart phone few years later, if we are given this year price, and each year's inflation rate in percent.

```
price = float(input("Enter phone's price:"))
n = int(input("Enter the number of years:"))
rate = float(input("Enter the inflation
rate:"))
rate_decimal = rate/100
```

Counting using the `range()` function

Example: Prices of various items change with time. Let's write the program that will allow us to calculate the price of the smart phone few years later, if we are given this year price, and each year's inflation rate in percent.

```
price = float(input("Enter phone's price:"))
n = int(input("Enter the number of years:"))
rate = float(input("Enter the inflation
rate:"))
rate_decimal = rate/100

newPrice = price
for i in range(0,n):
  newPrice = newPrice + newPrice*rate_decimal

print("The phone's price will be $%f" %
newPrice)
```

Counting using the `range()` function

In-class activity

Do the exercises 2-3 from the handout

While vs. for loops

Consider the following scenarios:

(a) iterate until the user enters a letter ‘p’ or a letter ‘q’

(b) iterate 50 times

(c) iterate until x is less than 19

(d) find the product of all values in a list

What looping mechanism to use? while or for?

While vs. for loops

Consider the following scenarios:

(a) iterate until the user enters a letter ‘p’ or a letter ‘q’
while loop

(b) iterate 50 times
for loop

(c) iterate until x is less than 19
while loop

(d) find the product of all values in a list
for loop

What looping mechanism to use? while or for?

While vs. for loops

As a guide line:

- Use a for loop when the number of iterations is computable before entering the loop
  ***examples**: counting down from X to 0, printing a string N times, etc.*
- Use a for loop when accessing the elements of a container
  ***examples**: when adding 1 to every element in a list, or printing the key of every entry in a dict, etc.*
- Use a while loop when the number of iterations is not computable before entering the loop
  ***example**: when iterating until a user enters a particular character.*

While vs. for loops

Example: Given a string **`myString`**, let's write a program to count the number of occurrences of letter 'a' in it, without using built-in function `count()`.

While vs. for loops

Example: Given a string **`myString`**, let's write a program to count the number of occurrences of letter 'a' in it, without using built-in function `count()`.

1) What looping mechanism to use (while or for)?

While vs. for loops

Example: Given a string `myString`, let's write a program to count the number of occurrences of letter 'a' in it, without using built-in function `count()`.

1) What looping mechanism to use (while or for)? for loop

While vs. for loops

Example: Given a string **`myString`**, let's write a program to count the number of occurrences of letter 'a' in it, without using built-in function `count()`.

1) What looping mechanism to use (while or for)? for loop

2) let's write the code:

```
counter = 0
for ch in myString:
   if ch == 'a': counter += 1
# at the end of loop's executions, variable
# counter will have the number of occurrences
# of letter 'a' in myString
```

While vs. for loops

In-class activity

Do the exercises 4-5 from the handout