



# OMIS 30: Intro to Programming (with Python)

Week 4, Class 1

Introduction to Programming

Instructor: Denis Vrdoljak





# FOR/RANGE

## Week 4



# WHILE loop review

```
1 animals = ['Cat', 'Dog', 'Elephant', 'Giraffe']
2 i = 0
3
4 while i < len(animals):
5     print(animals[i] + " found in the barn.")
6     i += 1
```



# WHILE loop review

```
1 animals = ['Cat', 'Dog', 'Elephant', 'Giraffe']
2 i = 0
3
4 while i < len(animals):
5     print(animals[i] + " found in the barn.")
6     i += 1
```

---

Cat found in the barn.  
Dog found in the barn.  
Elephant found in the barn.  
Giraffe found in the barn.

---



# FOR Loops

- For loops iterate through an iterable.
- This can be a list, a string...or a set,



# FOR Loops

```
barn = ['cat', 'dog', 'elephant', 'giraffe', 'pig']
for animal in barn:
    print(animal + " found in the barn")
```

- For loops iterate through an iterable.
- This can be a list, a string...or a set,



# FOR Loops

- For loops iterate through an iterable.
- This can be a list, a string...or a set, a tuple, a dictionary, etc
- For loops stop once the last element in the iterable is used
- break and continue work with for loops, just like with while loops

```
barn = ['cat', 'dog', 'elephant', 'giraffe', 'pig']
for animal in barn:
    print(animal + " found in the barn")
```

output:

```
cat found in the barn
dog found in the barn
elephant found in the barn
giraffe found in the barn
pig found in the barn
```

In [27]:



# FOR Loops

```
for letter in "animal":  
    print(letter)
```



- For loops iterate through an iterable.
- This can be a list, a string...or a set,



# FOR Loops

- For loops iterate through an iterable.
- This can be a list, a string...or a set, a tuple, a dictionary, etc
- For loops stop once the last element in the iterable is used
- break and continue work with for loops, just like with while loops

output:

```
for letter in "animal":  
    print(letter)
```

a  
n  
i  
m  
a  
l

In [29]:



# For Loops vs While Loops

```
# Simple WHILE loop
barn = ['cat','dog','elephant','giraffe','pig']
i=0
# i is a counter that will help us iterate through the list
# when i gets to the end of the list, the loop will stop
while i < len(barn):
    animal = barn[i]
    print(animal + " found in the barn")
    i += 1
#increment counter
```

For loops iterate through an iterable (like a list)

While loops repeat until their boolean equality becomes false



# For Loops vs While Loops

```
for animal in barn:  
    print(animal + " found in the barn")
```

For loops iterate through an iterable (like a list)

While loops repeat until their boolean equality becomes false



# For Loops vs While Loops

```
# Simple WHILE loop
barn = ['cat','dog','elephant','giraffe','pig']
i=0
# i is a counter that will help us iterate through the list
# when i gets to the end of the list, the loop will stop
while i < len(barn):
    animal = barn[i]
    print(animal + " found in the barn")
    i += 1
#increment counter

for animal in barn:
    print(animal + " found in the barn")
```



# Introducing range()

- range() creates a list of numbers.



# Introducing range()

```
Counttofive =[0,1,2,3,4]
for i in Counttofive:
    print(i)
```

0  
1  
2  
3  
4

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

0  
1  
2  
3  
4

- range() creates a list of numbers.  
Basically, it counts for you.



# range(): Counting from 1

```
print (i)
```

```
1  
2  
3  
4  
5
```

```
Counttofive =[1,2,3,4,5]
```

```
for i in Counttofive:
```

```
for i in range(1,6):  
    print (i)
```

```
1  
2  
3  
4  
5
```



# range() in a Nutshell

```
Counttofive =[0,1,2,3,4]  
  
for i in Counttofive:  
    print(i)
```

0  
1  
2  
3  
4

```
Counttofive =[1,2,3,4,5]  
  
for i in Counttofive:  
    print (i)
```

1  
2  
3  
4  
5

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

0  
1  
2  
3  
4

```
for i in range(1,6):  
    print (i)
```

1  
2  
3  
4  
5



# FOR loops with range()

```
: # for loops with range
barn = ['cat','dog','elephant','giraffe','pig']
i=0

# another way to loop is to use range to slice by indexe

for idx in range(len(barn)):
    print(barn[idx] + " found in the barn")

print("\n\n")
```

```
cat found in the barn
dog found in the barn
elephant found in the barn
giraffe found in the barn
pig found in the barn
```

- another way to loop is to use range to slice by index



# Modules and packages

Libraries two important forms.

## Modules

Simply .py file with multiple functions

Importing a module imports the functions in the .py file

## Packages

A package is a folder with relate modules inside it

The package must have a file called `__init__.py` to tell python that a folder is a package

Importing a package imports all the modules in the folder



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```

Objects within the library can be accessed with dot notation:

```
random.choose(mylist)
```

Objects can also be imported as an import from:

```
from random import choose  
choose(mylist)
```

More on the Random library can be found here:

<https://docs.python.org/3/library/random.html>



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```

Objects within the library can be accessed with dot notation:

```
random.choose(mylist)
```



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```

Objects within the library can be accessed with dot notation:

```
random.choose(mylist)
```

Objects can also be imported as an import from:

```
from random import choose  
choose(mylist)
```



# Other syntaxes for imports

Import numpy as np

i.e. np.zeros([10,10])

```
from numpy import *  
zeros([10,10])
```

```
from numpy import zeros  
zeros([10,10])
```

```
from numpy import zeros as z  
z([10,10])
```



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```

Objects within the library can be accessed with dot notation:

```
random.choose(mylist)
```



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```

Objects within the library can be accessed with dot notation:

```
random.choose(mylist)
```

Objects can also be imported as an import from:

```
from random import choose  
choose(mylist)
```



# Import Statement

Importing libraries is done with the “import” statement

Imports must be at the top of the script

Example:

```
import random
```

Objects within the library can be accessed with dot notation:

```
random.choose(mylist)
```

Objects can also be imported as an import from:

```
from random import choose  
choose(mylist)
```

More on the Random library can be found here:

<https://docs.python.org/3/library/random.html>