Section 001
CDS 411 *Modeling and Simulation II*

Python fundamentals

Class 3

Python as a calculator

Arithmetic

Basic arithmetic is easy to do in Python

- Addition, subtraction, multiplication, division
- Raising to a power
- Order of operations

```
print(3 * (5 / 8 - 1.25) ** 2)
```

```
## 1.171875
```

The `print()` function is the easiest way to tell Python to show output.

Variable assignment

Just like in algebra or in other programming languages, we can use variable assignment to store values for future use.

```
length_of_bridge = 15
```

To see what's in the variable, we can print it:

```
print(length_of_bridge)
```

```
## 15
```

You can increment variables easily in Python:

```
# What do you think the result will be?
length_of_bridge += 2 * length_of_bridge
```

Variable assignment

Just like in algebra or in other programming languages, we can use variable assignment to store values for future use.

```
length_of_bridge = 15
```

To see what's in the variable, we can print it:

```
print(length_of_bridge)
```

```
## 15
```

You can increment variables easily in Python:

```
# What do you think the result will be?
length_of_bridge += 2 * length_of_bridge
```

```
print(length_of_bridge)
```

The import statement

Arithmetic with mathematical functions

Python has built-in support for mathematical functions like sines, cosines, exponentials, factorials, etc. However, you need to load them in first.

```
import math
```

After running this, the mathematical functions can be accessed by appending a period `.` to the end of the word `math`, and then typing out the correct function name.

```
print(math.factorial(12))          # Evaluates 12!
```

```
## 479001600
```

```
print(5 * math.sin(math.pi / 3))  # Evaluates 5sin(π/3)
```

```
## 4.330127018922193
```

Loading external .py files

The import statement can also be used to load `.py` files. Say you have a file named `my_program.py` **in the same directory as the file or Jupyter notebook you're currently editing.** To load the code from `my_program.py`, used

```
import my_program
```

This will cause the code inside `my_program.py` to be executed once. If any functions were defined in this file, for example a function named `my_great_method()`, you can access it in the same way you use the `math` module:

```
my_program.my_great_method()
```

The Python Standard Library

`math` is not the only built-in module available in Python. A full list of all available modules along with their documentation are available online:
https://docs.python.org/3.6/library/

Examples of modules I use a lot are:

- `re`: Support for regular expressions
- `collections`: Specialized container data types for Python
- `itertools`: Functions creating iterators for efficient looping
- `functools`: Higher-order functions and operations on callable objects
- `pathlib`: Object-oriented filesystem paths
- `shutil`: High-level file operations
- `os`: Miscellaneous operating system interfaces
- `logging`: Logging facility for Python
- `subprocess`: Subprocess management
- `sys`: System-specific parameters and functions

and more!

Data structures

Using lists

- What is a list?
- Lists are not vectors.
- Anything can go into a list.
- How to access and slice elements in a list.
- Nesting lists
- Appending to lists
- Joining lists

Dictionaries

- What is a Python dictionary?
- Key-value pairs
- Nearly anything can be a key or a value
- Lists can go in dictionaries
- Nested dictionaries
- Viewing keys and values
- Accessing with brackets versus `.get()`

Other structures

- Tuples
- Sets

Control flow

Loops

- What are loops and iteration?
- Basic for loops with `range()`
- For loop over list with enumeration
- DON'T USE COUNTER VARIABLES
- Looping over dictionaries

Credits

License