Why Python?
(Intro. to Python 1)

Dr. Ryan Krauss

Grand Valley State University

pdm

Why Python?

- free/open-source

pdm

Why Python?

- free/open-source
- clean syntax

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose
- easy to learn

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose
- easy to learn
- "batteries included"

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose
- easy to learn
- "batteries included"
  - many, many libraries available

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose
- easy to learn
- "batteries included"
  - many, many libraries available
  - python-control library will be very valuable for our class

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose
- easy to learn
- "batteries included"
  - many, many libraries available
  - python-control library will be very valuable for our class
  - numpy, scipy, matplotlib, IPython, spyder, ...

pdm

Why Python?

- free/open-source
- clean syntax
- powerful and all purpose
- easy to learn
- "batteries included"
  - many, many libraries available
  - python-control library will be very valuable for our class
  - numpy, scipy, matplotlib, IPython, spyder, ...
- I like it and have a lot of experience using and teaching it

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration
  - transfer functions

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration
  - transfer functions
- process experimental data (.csv or .xlsx)

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration
  - transfer functions
- process experimental data (.csv or .xlsx)
- find frequency response of various systems

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration
  - transfer functions
- process experimental data (.csv or .xlsx)
- find frequency response of various systems
- design feedback control systems based on root locus and Bode

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration
  - transfer functions
- process experimental data (.csv or .xlsx)
- find frequency response of various systems
- design feedback control systems based on root locus and Bode

pdm

What do we need for dynamic systems and control?

- model dynamic systems and simulate their response to various inputs
  - numeric integration
  - transfer functions
- process experimental data (.csv or .xlsx)
- find frequency response of various systems
- design feedback control systems based on root locus and Bode

- Big Idea: Python has open-source libraries for doing all of these things and it is free and fairly easy to learn

pdm

Python vs. C syntax 1

- Python uses dynamic typing, so you do not have to pre-declare your variable types

C

```
int a;
a = 7;
```

Python

```
a = 7
```

pdm

Python vs. C syntax 1

- Python uses dynamic typing, so you do not have to pre-declare your variable types
- semi-colons are not necessary at the end of every line

C

```
int a;
a = 7;
```

Python

```
a = 7
```

pdm

Python vs C. Syntax 2: for loops

C

```
for (int i=0; i<10; i++){
    cout << "i = " << i
}
```

Python

```
for i in range(10):
    print('i = ' + str(i))
```

pdm

Python vs. C: functions

C

```c
int my_C_func(int A, int B){
    int C;
    C = A + B;
    return C;
}
```

Python

```python
def my_Python_func(A, B):
    C = A + B
    return C
```

pdm

C functions: only 1 return value

- C functions can only return one thing

```
// declare D outside of function
int D;

int my_C_func(int A, int B, int* D){
    int C;
    C = A + B;
    D = A*B;
    return C;
}

// call passing in pointer to D
my_C_func(2, 7, &D);
```

pdm

C functions: only 1 return value

- C functions can only return one thing
  - to work around this, you have to pass in a pointer for your other "outputs"

```
// declare D outside of function
int D;

int my_C_func(int A, int B, int* D){
    int C;
    C = A + B;
    D = A*B;
    return C;
}

// call passing in pointer to D
my_C_func(2, 7, &D);
```

pdm

Python functions: multiple return values

Returning multiple values from a Python function is straightforward:

```
def my_Python_func(A, B):
    C = A + B
    D = A*B
    return C, D
```

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops
  - `if/then` statements

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops
  - `if/then` statements
  - defining functions

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops
  - `if/then` statements
  - defining functions
  - plotting

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops
  - `if/then` statements
  - defining functions
  - plotting
  - working with vectors and matrices

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops
  - `if/then` statements
  - defining functions
  - plotting
  - working with vectors and matrices
  - reading data files

pdm

Learning a new language

- what things do you need to learn to use a new language for engineering purposes?
  - basic syntax
  - declaring variables
  - `for` loops
  - `if/then` statements
  - defining functions
  - plotting
  - working with vectors and matrices
  - reading data files
  - simulating dynamic systems

pdm