Block 1: Programming Mindset & Python Setup

Python Module for Incoming ISE & OR PhD Students

Will Kirschenman

August 7, 2025 | 9:00 AM - 9:50 AM

North Carolina State University

NC STATE UNIVERSITY

Welcome to Block 1!

- **Goal:** Build the foundation for your PhD journey in Python
- **Duration:** 50 minutes of interactive learning
- **Format:** Presentation + hands-on notebook exercises

What We'll Cover
Programming mindset · Python ecosystem · Google Colab · Python fundamentals

Session Learning Objectives

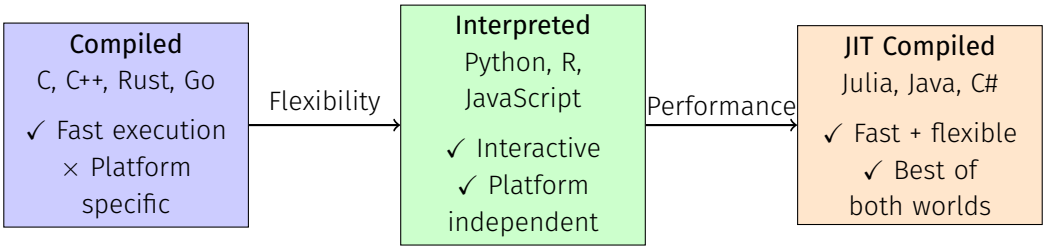
By the end of Block 1, you will:

1. Understand **why programming matters** in OR/ISE
2. Know **where Python fits** in the programming landscape
3. Be comfortable with **Google Colab** as your development environment today
4. Have **hands-on experience** with Python basics
5. Be **ready for Block 2** (NumPy & Pandas)

Interactive Approach
We'll switch between slides and the Jupyter notebook throughout!

Programming Paradigms

The Programming Landscape

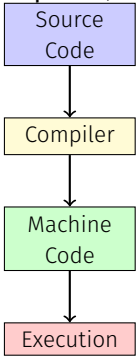


For OR/ISE Research

Python offers an excellent balance of ease-of-use, ecosystem, and performance for most research tasks.

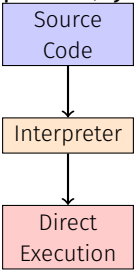
Programming Language Execution Models

Compiled (C++)



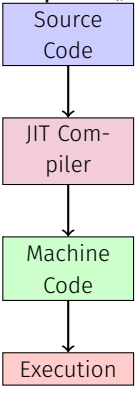
Recipe book: Translate once, use many times

Interpreted (Python)



Live translator: Translate and execute together

JIT Compiled (Julia)



Smart chef: Compile as needed during execution

Why Python for OR/ISE?

Python's Dominance in Scientific Computing

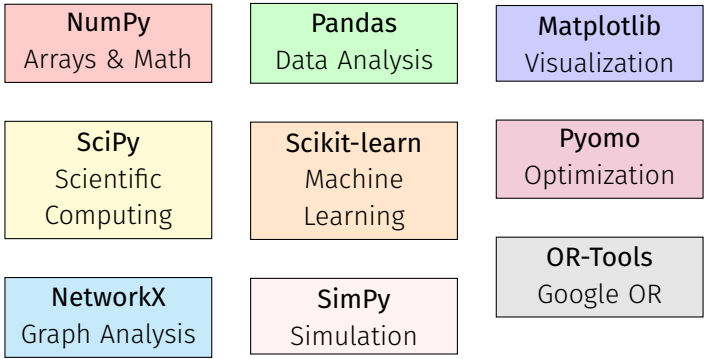
Language Comparison for OR/ISE:

Feature	Python	R	MATLAB	Julia
Learning	★★★	★★	★★★	★★
Data Science	★★★	★★★	★★	★★
Optimization	★★★	★★	★★★	★★
ML/AI	★★★	★★★	★★	★★
Community	★★★	★★	★★	★
Industry	★★★	★★	★★	★

Python's Advantages

- Gentle learning curve
- Massive ecosystem
- Industry standard
- Active research community
- Free and open source

Your Python Research Toolkit



What This Means for You

One language, endless possibilities: from data cleaning to optimization to machine learning!

Python's Expressiveness: A Quick Example

Optimization Problem: Minimize $f(x) = x^2 - 4x + 4$

In Python (3 lines):

```python
from scipy.optimize import
    minimize_scalar

f = lambda x: x**2 - 4*x + 4
result = minimize_scalar(f)
```

In C++ (50+ lines):

- Include headers
- Define objective function
- Implement optimization algorithm
- Handle numerical precision
- Error checking
- Memory management

The Python Advantage

Focus on **solving problems**, not wrestling with implementation details!

Google Colab

Essential Colab Shortcuts

Master these shortcuts for efficient coding:

Action	Shortcut
Run cell, go to next	`Shift + Enter`
Run cell, stay current	`Ctrl + Enter`
Add cell above	`A` (command mode)
Add cell below	`B` (command mode)
Delete cell	`DD` (command mode)
Cut cell	`X` (command mode)
Copy cell	`C` (command mode)
Paste cell below	`V` (command mode)
Change to code cell	`Y` (command mode)
Change to markdown	`M` (command mode)
Enter command mode	`ESC`
Comment/uncomment	`Ctrl/Cmd + /`
Show all shortcuts	`H` (command mode)

Command vs Edit Mode

Command mode (blue bar): Navigate & manipulate cells · **Edit mode** (green bar): Edit content

Python Fundamentals

Python's Philosophy: Simple and Readable

"Python is executable pseudocode"

Variables (No Declaration Needed!)

- `x = 42` (integer)
- `pi = 3.14159` (float)
- `name = "OR"` (string)
- `is_awesome = True` (boolean)

Collections

- `numbers = [1, 2, 3, 4]`
- `mixed = [42, "Python", 3.14]`
- `algorithms = ["Simplex", "IP"]`

Python vs. Other Languages

Java: `int x = 42;`

C++: `std::vector<int> numbers;`

Python:
```
x = 42
numbers = [1, 2, 3]
```

Less typing, more thinking!

Control Flow: Making Decisions

Python uses indentation instead of brackets!

If Statements

```
score = 85

if score >= 90:
    grade = "A"
    print("Excellent!")
elif score >= 80:
    grade = "B"
    print("Good job!")
else:
    grade = "C"
    print("Keep trying!")
```

For Loops

```
algorithms = ["Simplex", "IP", "DP"]

for algorithm in algorithms:
    print(f"Learning {algorithm}")

# With enumeration
for i, algo in enumerate(algorithms):
    print(f"{i+1}. {algo}")
```

Key Insight

Indentation makes Python code naturally readable and organized!

Functions: Building Reusable Code

Functions are essential for code organization:

Basic Function

```python
import math

def calculate_eoq(demand, order_cost,
                  holding_cost):
    """Calculate Economic Order Quantity
        """
    eoq = math.sqrt(2 * demand *
                    order_cost /
                        holding_cost)
    return eoq

# Usage
result = calculate_eoq(1000, 600, 10)
print(f"EOQ={result:.2f}")
```

Research Applications

- **Objective functions** for optimization
- **Data processing** pipelines
- **Simulation** components
- **Algorithm** implementations
- **Reproducible** experiments

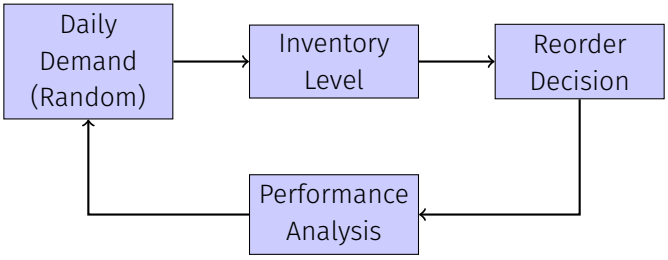
Best Practice

Always include docstrings to document what your functions do!

Practical Application

Putting It All Together: OR/ISE Example

Let's see Python in action with a simple inventory simulation

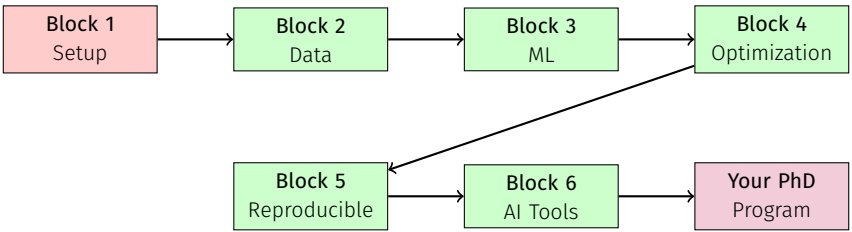


What You'll Build in the Notebook

- Simulate inventory system with Python functions
- Use loops and conditions for decision logic
- Visualize results with matplotlib
- Analyze performance metrics

From Simple to Sophisticated

Today's foundation → Your PhD program



Next Steps

What We Accomplished in Block 1 ✓

- ✓ **Understanding** why programming matters in OR/ISE
- ✓ **Positioned Python** in the programming landscape
- ✓ **Set up** Google Colab for development
- ✓ **Hands-on practice** with Python fundamentals
- ✓ **Built** a practical inventory simulation

Ready for Block 2!
You now have the foundation to dive into data manipulation with NumPy & Pandas

Preview: Block 2 (10:00 AM - 10:50 AM)

Essential Data Wrangling with NumPy & Pandas

What's Coming

- **NumPy arrays** for fast computation
- **Pandas DataFrames** for data analysis
- **Real dataset** manipulation
- **Data cleaning** techniques
- **PhD student dataset** (with humor!)

Quick Prep

- Save your Block 1 notebook
- Stretch and grab coffee
- Think about data you work with
- Get ready for more hands-on coding!

10-minute break, then we continue to Block 2!

Resources for Continued Learning

Essential Resources

- **Python Official Tutorial**
  docs.python.org/3/tutorial/
- **Python Tutor**
  pythontutor.com (visualize code)
- **Practice Python**
  practicepython.org
- **Google Colab Guide**
  colab.research.google.com

Before Block 2

- Re-run all notebook cells
- Try modifying the inventory simulation
- Think about data in your research

Remember: Save your notebook to Google Drive!
File → Save a copy in Drive

Questions?

See you in 10 minutes for Block 2!