USDA Forest Service
U.S. DEPARTMENT OF AGRICULTURE

Introduction to Geospatial Scripting

Monica Vermillion, Monica.Vermillion@usda.gov

September | 2021

GTAC Geospatial Technology and Applications Center

Housekeeping

Make sure you have registered for Part 2

Keep video off and stay muted

If you have a question:

- Raise hand in Teams
- Respond in chat box
- Q & A at end of presentation

Closed captions are available

Personal Intro

Monica Vermillion (she/her)

- Remote Sensing Specialist and Trainer

Geospatial Technology and Applications Center (GTAC)

- The University of Alabama, Boise State University
- Cheatgrass mapping, lidar data collection, Google Earth Engine
- Rock climbing, skiing, whitewater kayaking

Course Agenda

Lecture 1: Fundamentals of Scripting

Break to do Exercise 1&2

Continue Lecture 1

Break for Exercise 3 & Lunch

Lecture 2: Geospatial Applications

Afternoon Exercise (Specific to Script)

Wrap-up discussion

Purpose of this Training

- Get you started in coding!
- Understand the benefits of scripting
- Teach the fundamentals of coding
- Make it less scary
- Provide reference material

Software and Course Data

Required Software (*choose 1*):

- RStudio
- Python 2 (Installed via ArcMap)
- Python 3 (Installed via ArcPro)
- Active GEE account

Course Data:

- Python: Course data folder
- R: Course data folder

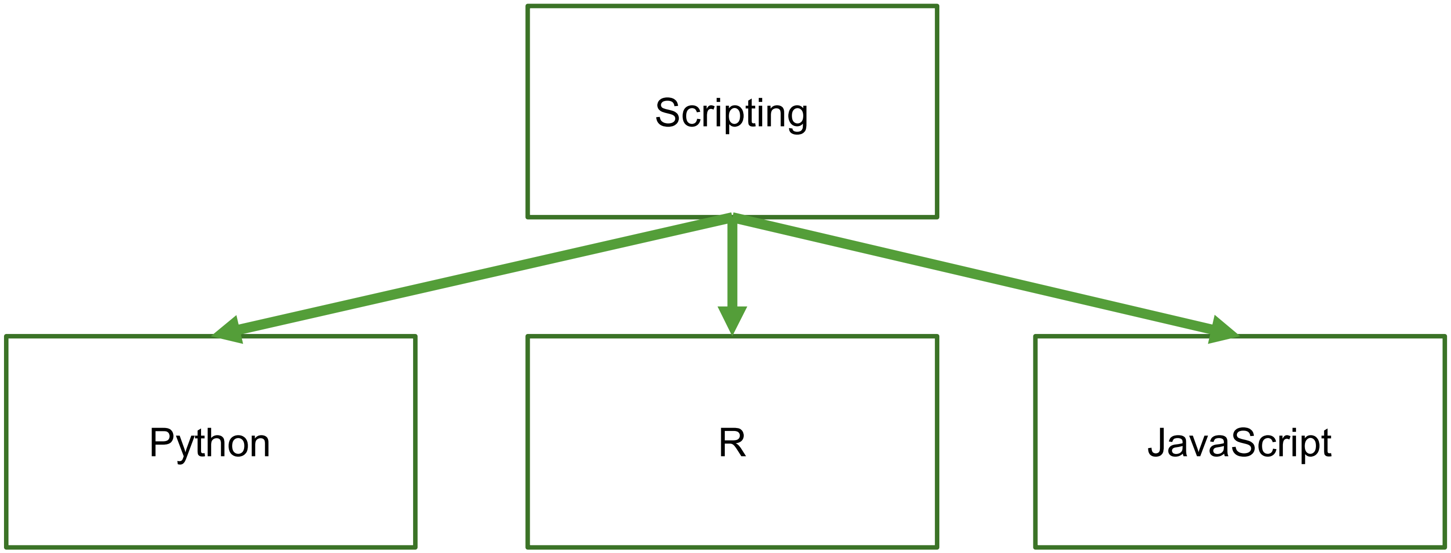
Which language do I use and why?

We are focusing on a specific geospatial application:

- **Python** – broad applications, integrates well with ArcMap and/or ArcPro
- **JavaScript** – gives access to Google Earth Engine, ideal for working with remote sensing data and imagery, especially time-series data
- **R** – powerful statistical capabilities, local data

Language Specific

- Differences in applications



Scripting

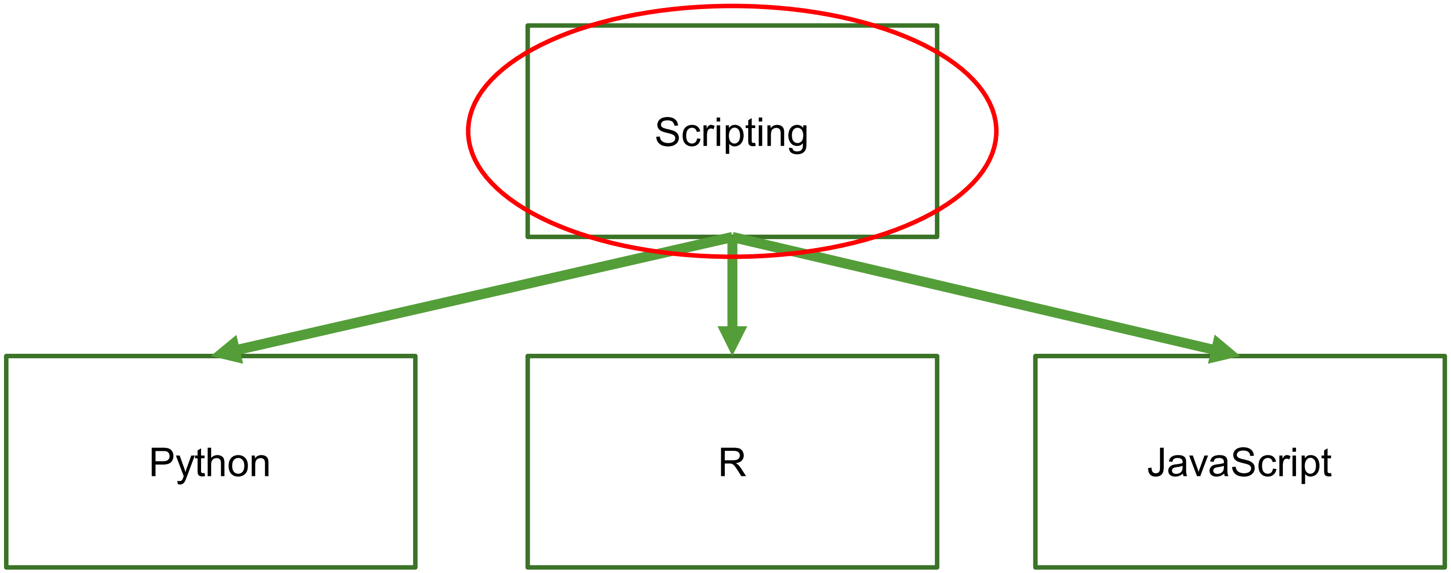
Python

R

JavaScript

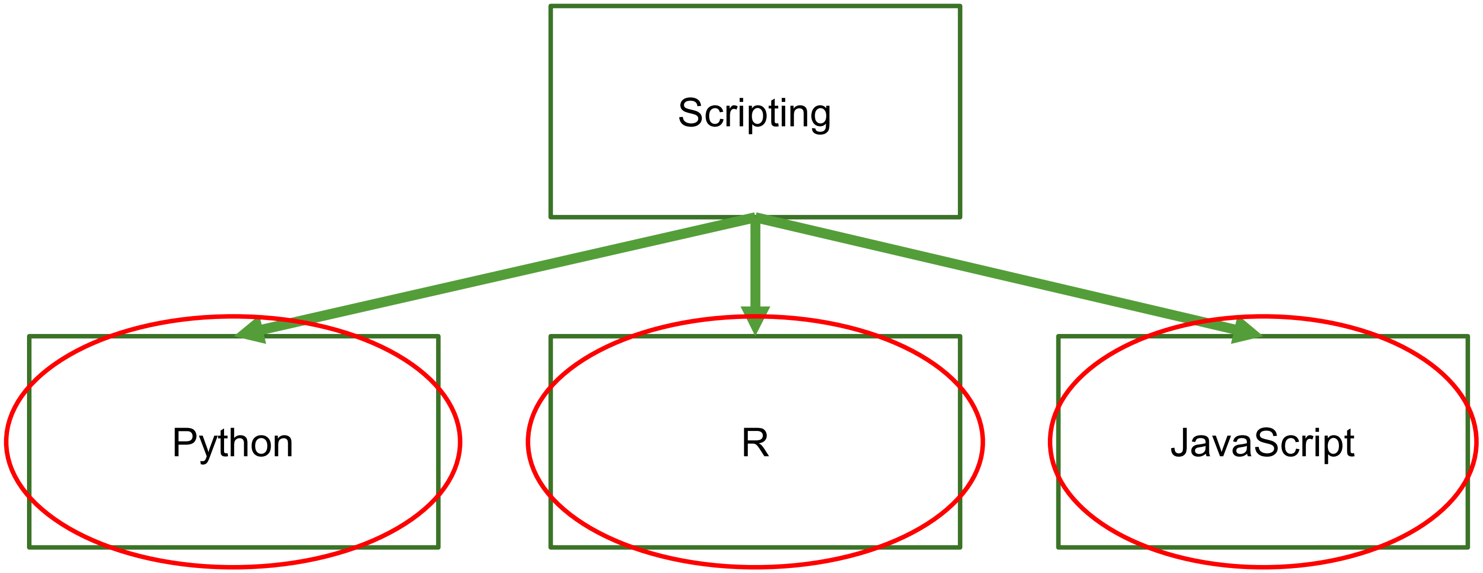
Language Specific

- Differences in applications



Language Specific

- Differences in applications



Python 2

- ArcPy
- Works well through ArcMap
  - Access to any ArcMap tools for automating hundreds of geoprocessing tasks
  - Python is so much more than ArcPy
- GTAC training:
  - *Geospatial Scripting in Python*
  - *Intermediate Geospatial Scripting in Python*

Python 3

- ArcPy
- Works well through ArcPro
  - Access to any ArcMap tools for automating hundreds of geoprocessing tasks
  - Python is so much more than ArcPy
- GTAC training:
  - *Geospatial Scripting in Python*
  - *Intermediate Geospatial Scripting in Python*

JavaScript

- Google Earth Engine (GEE)
  - Meant for massively parallel cloud computing Processing remote sensing data over a huge geographic extent
  - GEE has its own unique set of operators and methods, distinct from JavaScript
- GTAC Training:
  - *Geospatial Scripting in Google Earth Engine Code Editor*

R

- Rstudio
  - Created for statistical computing and graphical visualization
- GTAC Trainings:
  - *Geospatial Scripting in R*

Applications

- There are no inherent advantages to any language
- It is your application that will choose the language for you
  - e.g: Calculating a spectral index over hundreds of Landsat scenes > JavaScript (GEE)
  - e.g: Modeling forest characteristics based on lidar data > R
  - e.g: Processing hundreds of vector files> Python

USDA Forest Service
U.S. DEPARTMENT OF AGRICULTURE

Fundamental Concepts in Scripting

Morning Presentation, 2021

What is Scripting?

- How we relay instructions to a machine
- What is a programming language?
  - “A system of precisely defined symbols and rules devised for writing computer programs”

Why Scripting?

- Quick processing of large datasets
- Automate a workflow
- Reuse a script
- Minimize human error

Why Scripting?

- Quick processing of large datasets
- Automate a workflow
- Reuse a script
- Minimize human error

The downside?

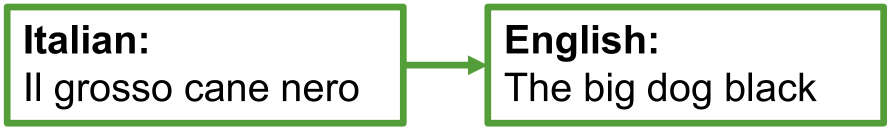
- Learning how to code takes **time**
- It is intimidating and seems like only really smart people code

What do I need to know to start scripting?

- Syntax – the format of the code
- Vocabulary – the commands you can use in the code

What is syntax?

Example of syntax in a natural language:

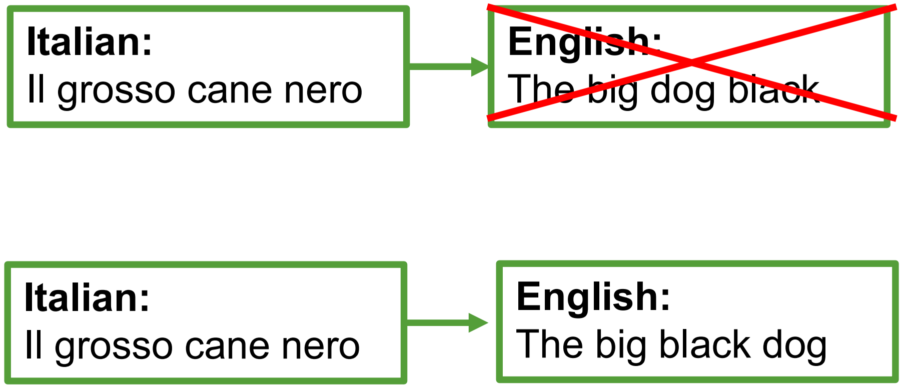


Italian:
Il grosso cane nero

English:
The big dog black

What is syntax?

Example of syntax in a natural language:



Italian:
Il grosso cane nero

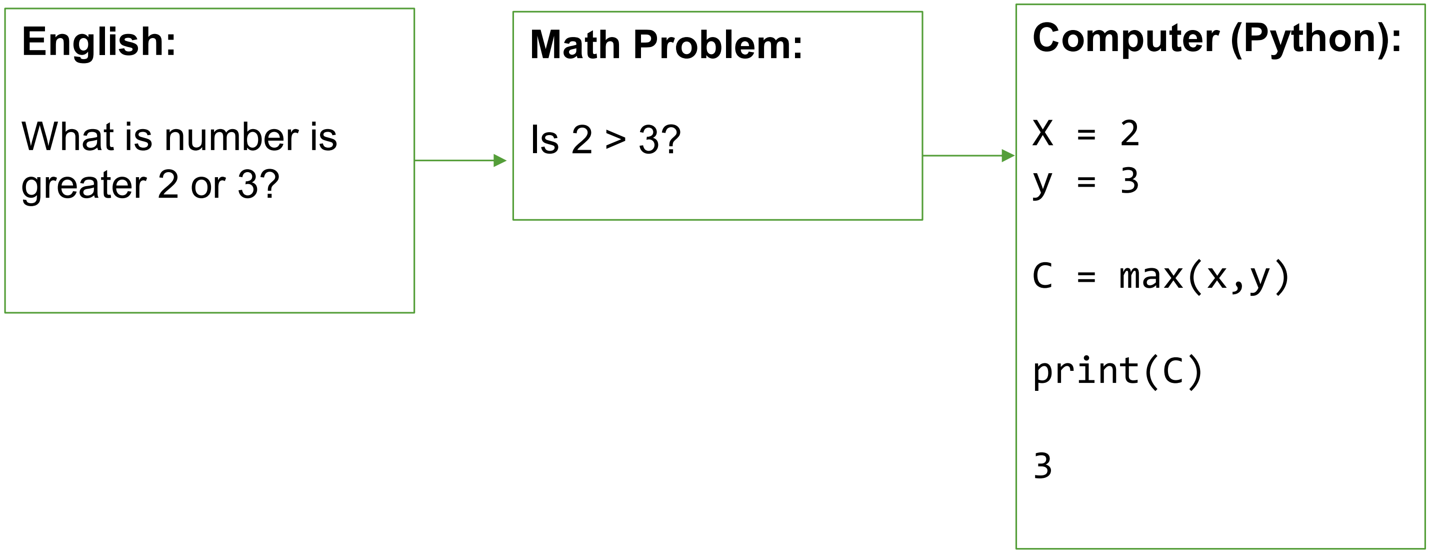
~~**English:**
The big dog black~~

Italian:
Il grosso cane nero

English:
The big black dog

What is syntax?

Example of syntax in coding language



English:

What is number is greater 2 or 3?

Math Problem:

Is 2 > 3?

Computer (Python):

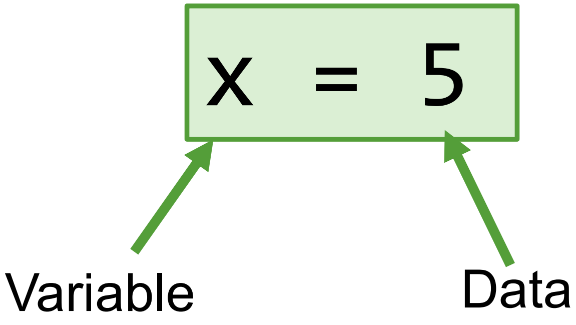
```
X = 2
y = 3

C = max(x,y)

print(C)

3
```

Getting Started

- Variables and Variable Types
- Statements
- Operators
- Conditional Statements
- Functions

Variables

- A variable is a name for a value
- Data is assigned a common name
- Different languages have different variable types and different syntax to declare variables



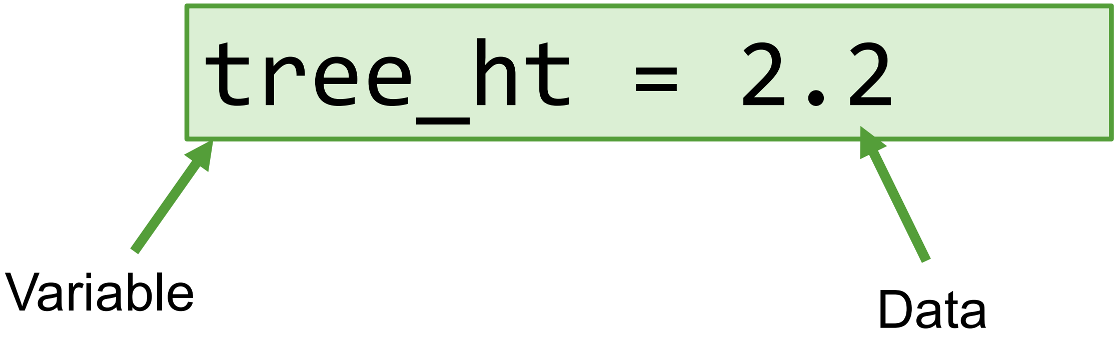
```
x = 5
```

Variable

Data

Variables

- A variable is a name for a value
- Data is assigned a common name
- Different languages have different variable types and different syntax to declare variables



```
tree_ht = 2.2
```

Variable

Data

Variables

- Variable names should be descriptive
- Variable naming rules

```
variable1 ≠ 1variable
```

- Case sensitivity

```
Variable ≠ variable
```

- Space sensitivity

```
variable1 ≠ variable 1
```

How do we declare a variable?

- A statement is an instruction
- Each of the statements below demonstrate how to declare the variable “X”, and assign it a value of 2

Language	Variable	Value	Code
Python	X	2	`X = 2`
JavaScript	X	2	`var X = ee.Number(2);`
R	X	2	`X <- 2`

Variable Types

- Integer

```
numPeople = 10
```

- Floating Point

```
biomass_g = 112.58
```

- Strings

```
personName = ‘Wren’
```

- List

```
x = [1,2,3,4], petNames = [‘Charlie’, ‘Mocha’, ‘Rudy’]
```

- Many more!

Statements

- A statement is an instruction
- Each of the statements below demonstrate how to declare the variable “X”, and assign it a value of 5

Language	Variable Name	Value	Code
Python	X	5	`X = 5`
JavaScript	X	5	`var X = ee.Number(5);`
R	X	5	`X <- 5`

Statements

Print statements output information

Language	Statement	Output

Python	`X=5`
	`print X`
		5

JavaScript	`var X = ee.Number(5);`
	`print(X);`
		5

R	`X <- 5`
	`print (X)`
		5

Operators

- Operators instruct the machine to perform a specific action
  - Arithmetic
    - +,-,*,/
  - Relational
    - =,>,<,≤,≥,≠
  - Logical
    - And, Or, Not

Operators

What do you expect the statement to return?

Language	Operator	Statement	Returns

Python	Equals	`X=5`
		`print X==5`	

JavaScript	Equals	`var X = ee.Number(5);`
		`print(X.eq(5))`	

R	Equals	`X <- 5`
		`print (X==5)`	

Operators

What do you expect the statement to return?

Language	Operator	Statement	Returns

Python	Equals	`X=5`
		`print X==5`
			True

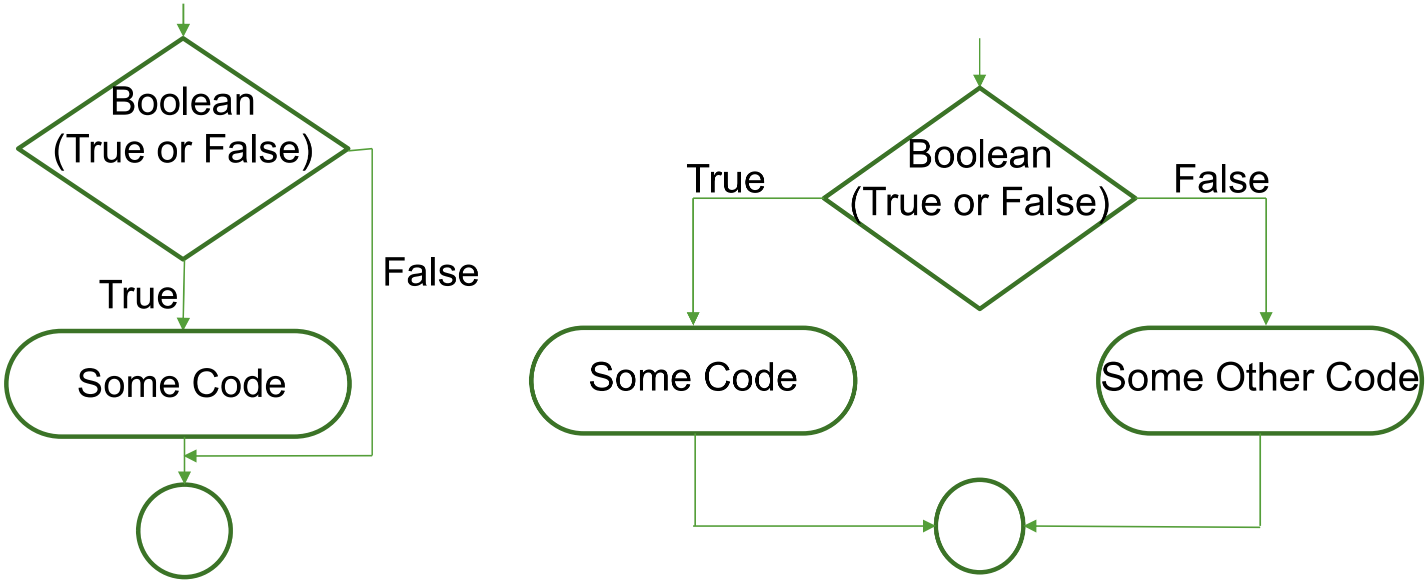
JavaScript	Equals	`var X = ee.Number(5);`
		`print(X.eq(5))`
			1

R	Equals	`X <- 5`
		`print (X==5)`
			True

Do Exercises 1 & 2

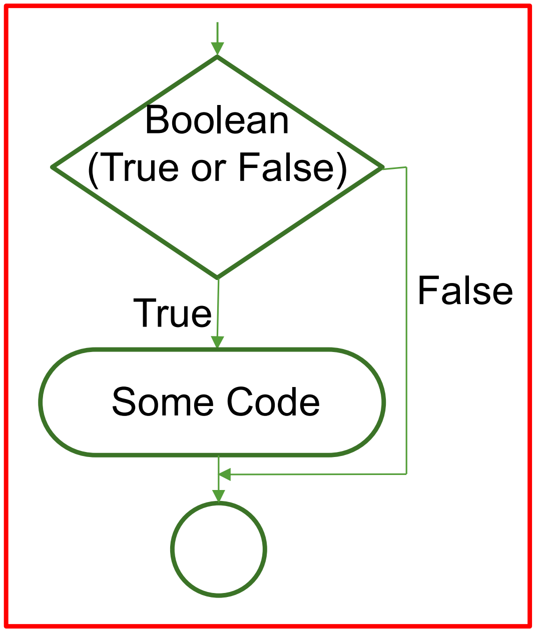
Conditional Statements

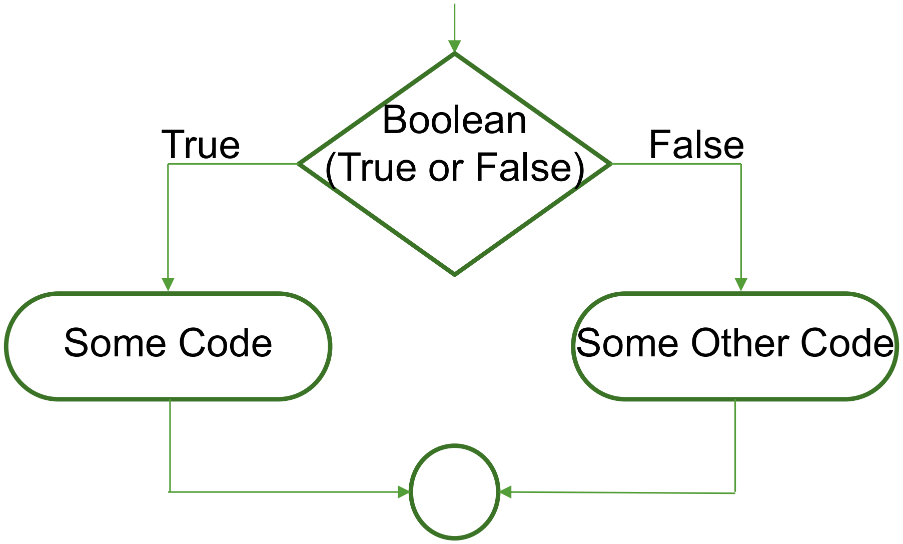
- Execute code only if a condition is met
- Visualizing a conditional statement:



Conditional Statements

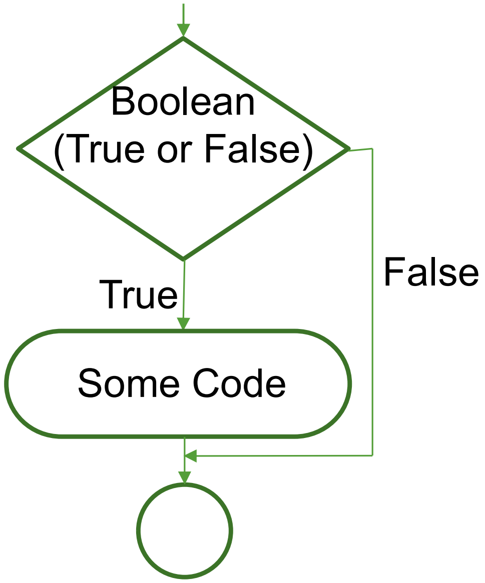
- Execute code only if a condition is met
- Visualizing a conditional statement:

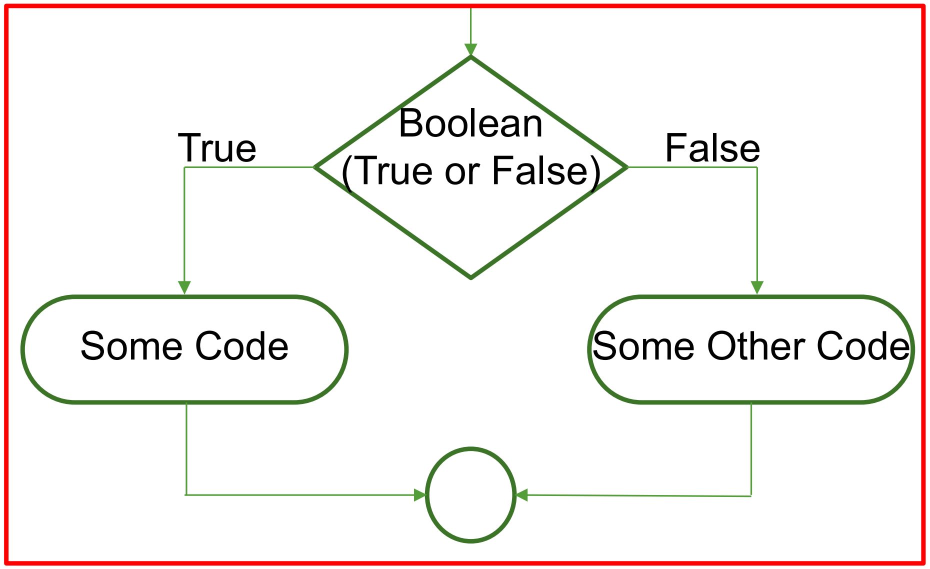




Conditional Statements

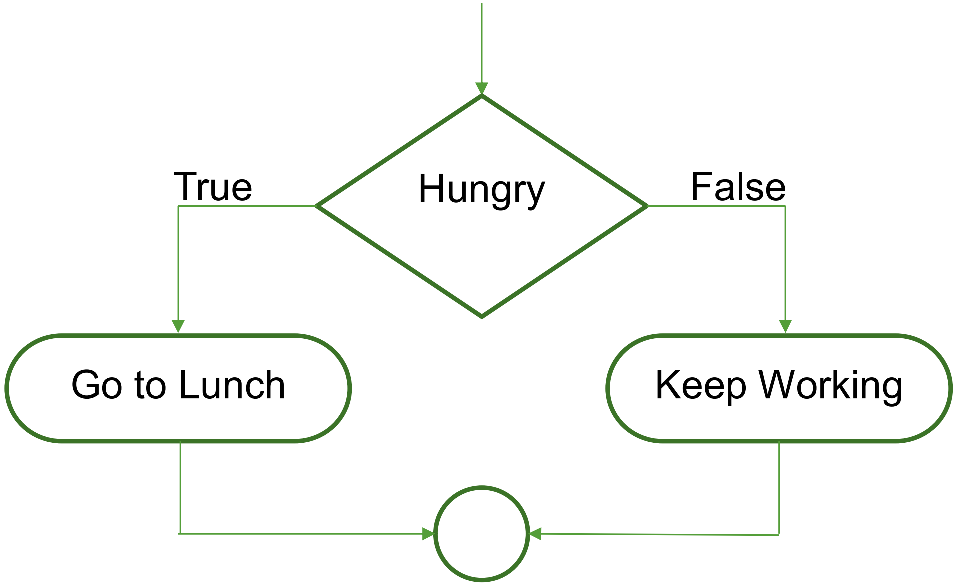
- Execute code only if a condition is met
- Visualizing a conditional statement:





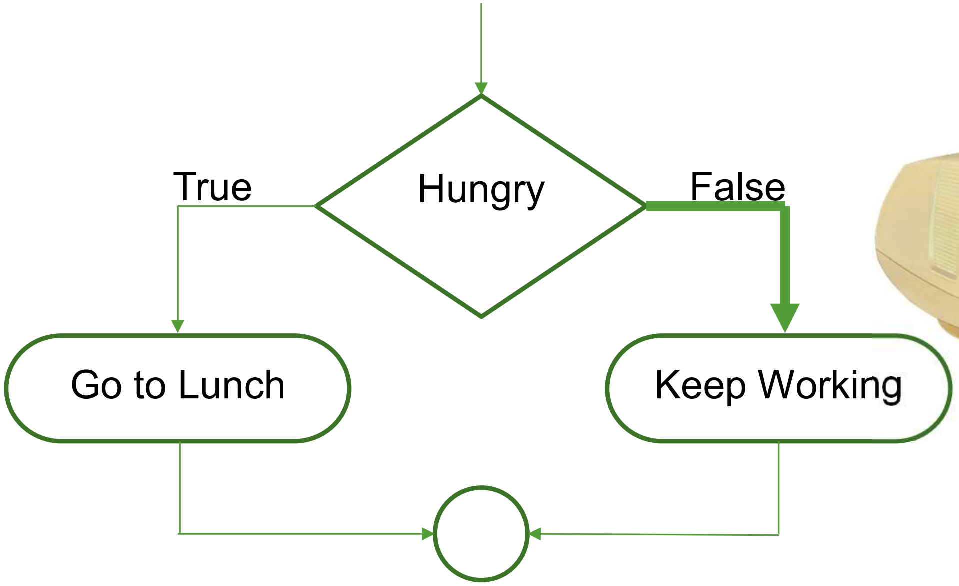
Conditional Statements

Visualizing a conditional statement:



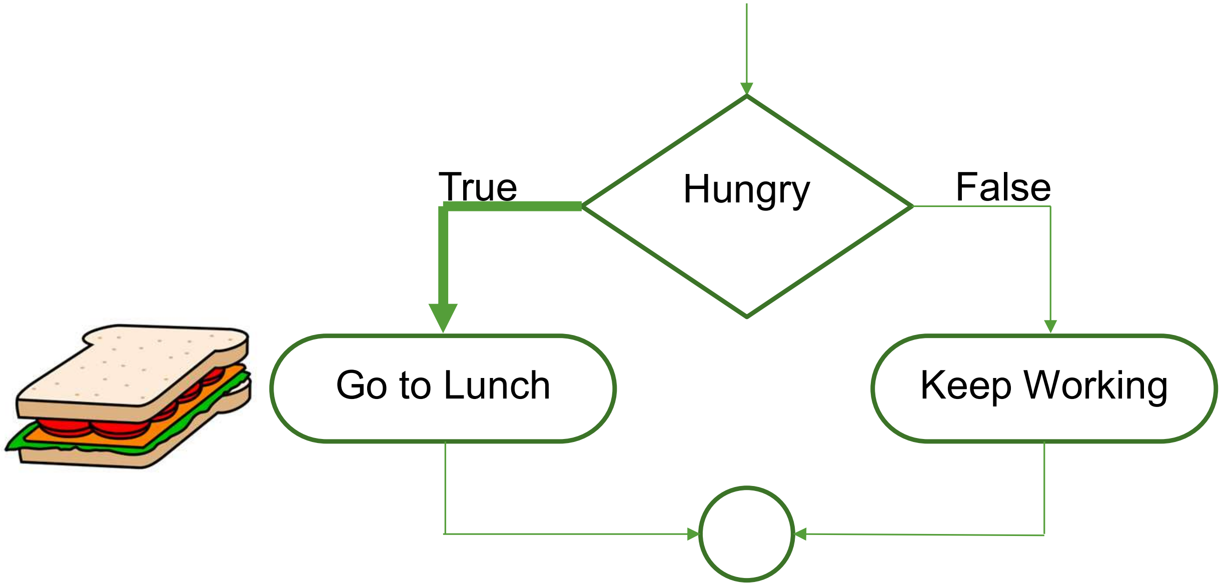
Conditional Statements

- Visualizing a conditional statement:



Conditional Statements

- Visualizing a conditional statement:



Conditional Statements

- Writing a conditional statement
  - Syntax is important and differs among languages.

Writing a Conditional Statement

Language	Statement (x=5)	Returns

Python	if x == 5:
	    print(“x is 5”)
	else:
	    print(“x is not 5”)
		x is 5

JavaScript	if (x.eq(5)){
	    print(“x is 5”)
	    }
	else{
	    print(“x is not 5”)
	    }
		x is 5

R	if (x == 5){
	    print(“x is 5”)
	    }else{
	    print(“x is not 5”)}
		x is 5

Functions

- Apply the same code over and over
- Advantages-
  - Simplify script maintenance
  - Complicated code only needs to be written once
- Best practice

Functions

- Example: wrapping up equations
  - Complicated equation:

$$Skewness = \frac{\sum_{i=1}^{N}\left(Y_i - \overline{Y}\right)^3}{(N-1)s^3}$$

Functions

- Example: wrapping up equations
  - Complicated equation:

$$\left\{ Skewness = \frac{\sum_{i=1}^{N}\left(Y_i - \overline{Y}\right)^3}{(N-1)s^3} \right\}$$

Functions

- Example: wrapping up equations
  - Complicated equation:

$$\text{Skew}(Y,N,s) = \left\{ \frac{\sum_{i=1}^{N}\left(Y_i - \overline{Y}\right)^3}{(N-1)s^3} \right\}$$

Functions

- Syntax

Language	Function Syntax

Python	`def <function name>(<arguments>):`
	`    <function code>`
	`    return <variable to output>`

JavaScript	`function <function name>(<arguments>){`
	`    <function code>`
	`    return(<variable to output>)`
	`}`

R	`<function name> <- function(<arguments>){`
	`    <function code>`
	`    return(<variable to output>)`
	`}`

Functions

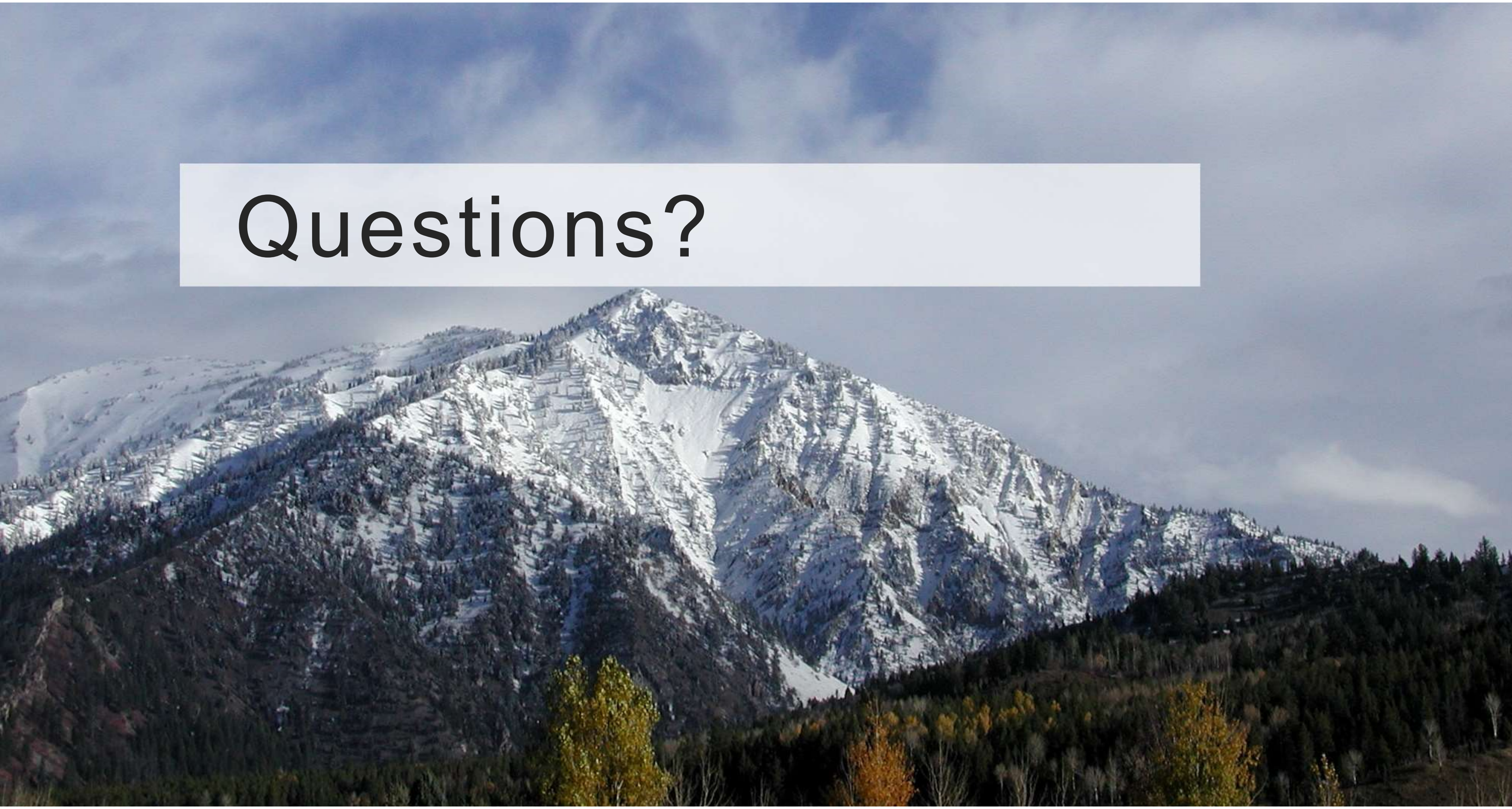
- Variable Scope
- Global and local variables
  - Global variables exist everywhere in the script
  - Local variable exist **only inside the function they're declared in**
- A variable created inside of a function has no meaning in the script anywhere outside of that function

Wrap Up

- Variables and Variable types
- Statements
- Operators
- Conditional Statements
- Functions

Exercise 3 & Lunch

- Morning exercises will give you an opportunity to practice all concepts



Questions?

Schedule

- ~~Morning Presentation~~
  - ~~Fundamental Concepts in Programming~~
- Morning Exercises
- Afternoon Presentation
  - Geospatial applications of Python, R, and JavaScript
- Afternoon Exercises
- Wrap-up Discussion