



Welcome



alphacloudone.com



Agenda

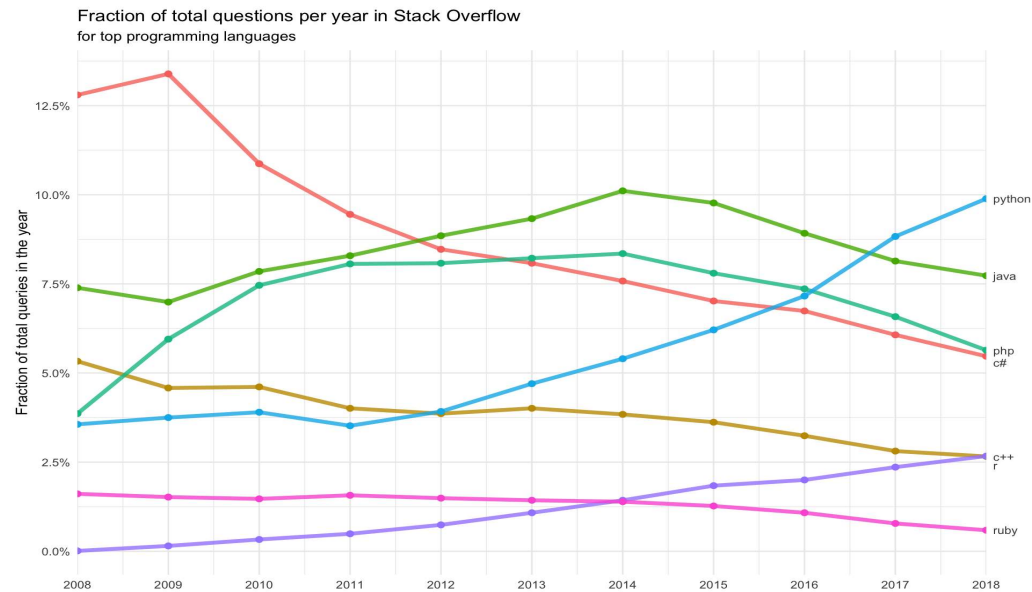
- Why Python
- Your Instructor
- Course overview
- Getting the most out of the Course
- How to get help
- Where to find the course materials
- The Jupyter Notebook
- Objects and Data Types
- Exercises
- Q&A





Bootcamp

Why Python





Bootcamp

Your Instructor

Chris Pflum

- Cloud Engineer, Cloud Architect, PM, Instructor
- 20 Years Software Development and IT Experience
- C, C++, SAS, Java, JavaScript, Python
- Machine learning engineer- supervised and unsupervised models. Regression, Categorization, Neural Nets
- AWS Big Data and Developer Associate certifications
- Originally from Ohio, currently reside in Tampa, Florida
- BA and MBA from The Ohio State University
- On Twitter- **@Chris_Pflum**, or **chrispflum** on LinkedIn



alphacloudone.com



High-level Course Overview

- Jupyter Notebook
- Comments, readability
- Print vs output
- Objects and data types
- Comparison operators
- Conditional statements
- Loops
- Other operators
- Methods/functions
- Class Objects
- User Input
- Scope- LEGB
- Final Milestone Project





Getting the Most out of this Course

1. Complete the Coding Exercises
2. Ask questions
3. Leverage the Notebooks





How to get help

1. Use the chat
2. Q&A at end of each session
3. Python.org
4. Google
5. Stack Overflow





Course Materials

These slides and sample notebooks can be found at:

[Github.com/alphacloud/](https://github.com/alphacloud/)



alphacloudone.com



Thank You



alphacloudone.com



Three types of tools for coding Python

- Text Editor
- IDE
- Notebook environment





Notebooks



- Use blocks of code, in cells
- Great for learning, get see input and output right next to each other
- Support in-line markdown notes
 - (great for including explanatory text)
- Support images, visualizations, and more
- Specialized file format, eg ipynb for Jupyter





Bootcamp

Readability

 Class & Inheritance in Java :	 Class & Inheritance in Python :
<pre>class Animal{ private String name; public Animal(String name){ this.name = name; } public void saySomething(){ System.out.println("I am " + name); } } class Dog extends Animal{ public Dog(String name) { super(name); } public void saySomething(){ System.out.println("I can bark"); } } public class Main { public static void main(String[] args) { Dog dog = new Dog("Chiwawa"); dog.saySomething(); } }</pre>	<pre>class Animal(): def __init__(self, name): self.name = name def saySomething(self): print "I am " + self.name class Dog(Animal): def saySomething(self): print "I am " + self.name\ + ", and I can bark" dog = Dog("Chiwawa") dog.saySomething()</pre>



Variables

- Names- can't start with number, no spaces, no symbols
- Best practice- lowercase (unless global)
- Avoid Python keywords
- Variables can be dynamically assigned to different data types
- Compared to static variable names in C++, Java, etc
- Can use `type()` function to determine current type of variable





Bootcamp

Variables

- Integers (int) whole numbers 2, 5, 11
- Float (float) numbers with a decimal point 3.14
- Boolean (bool) Logical value True or False
- String (str) text, or an ordered sequence of characters "Hello World"
- List (list) ordered sequence of objects [2, 3.14, "Hello World"]
- Dictionary (dict) unordered Key-Value pairs {"car": "BMW", "beverage": "milk"}
- Tuple (tup) immutable sequence of objects (10, "a", 3.14)
- Set (set) unordered collection of unique objects





Bootcamp

Variables

- Integers (int) whole numbers 2, 5, 11
- Float (float) numbers with a decimal point 3.14
- Boolean (bool) Logical value True or False
- String (str) text, or an ordered sequence of characters "Hello World"
- List (list) ordered sequence of objects [2, 3.14, "Hello World"]
- Dictionary (dict) unordered Key-Value pairs {"car": "BMW", "beverage": "milk"}
- Tuple (tup) immutable sequence of objects (10, "a", 3.14)
- Set (set) unordered collection of unique objects

