I hope you're ready to get started! Below is the outline for this course—each section will include some mix of videos, text explanations, Python examples, and practice questions. The mix of formats will keep each bit of information interesting and digestible. Feel free to post in the forums if you have questions!

1. **Introduction and Efficiency**
    - Course Introduction
    - Syntax
    - Efficiency
    - Notation of Efficiency
2. **List-Based Collections**
    - Lists/Arrays
    - Linked Lists
    - Stacks
    - Queues
3. **Searching and Sorting**
    - Binary Search
    - Recursion
    - Bubble Sort
    - Merge Sort
    - Quick Sort
4. **Maps and Hashing**
    - Maps
    - Hashing
    - Collisions
    - Hashing Conventions

5. **Trees**
    - Trees
    - Tree Traversal
    - Binary Trees
    - Binary Search Trees
    - Heaps
    - Self-Balancing Trees
6. **Graphs**
    - Graphs
    - Graph Properties
    - Graph Representation
    - Graph Traversal
    - Graph Paths
7. **Case Studies in Algorithms**
    - Shortest Path Problem
    - Knapsack Problem
    - Traveling Salesman Problem
8. **Technical Interview Tips**
    - Mock Interview Breakdown
    - Additional Tips
    - Practice with Pramp
    - Next Steps

Please check off the following topics to confirm that you already understand them! You're only expected to understand the basics of how to manipulate them—details about usage will be explained later if needed. If anything looks unfamiliar, you can check out one of our introductory Python classes. This course will also default to Python 2 where applicable, though it's very similar to Python 3!

Comments

```
# This is a one-line Python comment - code blocks are so useful!
"""This type of comment is used to document the purpose of functions
and classes."""
```

Declaration/Initialization

```
# Remember values, not variables, have data types.
# A variable can be reassigned to contain a different data type.
answer = 42
answer = "The answer is 42."
```

Data Types

```
boolean = True
number = 1.1
string = "Strings can be declared with single or double quotes."
list = ["Lists can have", 1, 2, 3, 4, "or more types together!"]
tuple = ("Tuples", "can have", "more than", 2, "elements!")
dictionary = {'one': 1, 'two': 2, 'three': 3}
variable_with_zero_data = None
```

Simple Logging

```
print "Printed!"
```

Conditionals

```
if cake == "delicious":
    return "Yes please!"
elif cake == "okay":
    return "I'll have a small piece."
else:
    return "No, thank you."
```

Loops

```
for item in list:
    print item

while (total < max_val):
    total += values[i]
    i += 2
```

Functions

```
def divide(dividend, divisor):
    quotient = dividend / divisor
    remainder = dividend % divisor
    return quotient, remainder

def calculate_stuff(x, y):
    (q, r) = divide(x,y)
    print q, r
```

Classes

```
class Person(object):
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def birthday(self):
        self.age += 1
```

You shouldn't need to run Python code outside the classroom, so don't worry if you don't have a development environment set up!

Start Quiz

NEXT

Below are some examples of functions in Python. Look at each and take note of the time efficiency. Then, in the quiz, enter those values using the correct notation. Use approximations wherever possible!

```
"""input manatees: a list of "manatees", where one manatee is
represented by a dictionary
a single manatee has properties like "name", "age", et cetera
n = the number of elements in "manatees"
m = the number of properties per "manatee" (i.e. the number of
keys in a manatee dictionary)"""

def example1(manatees):
    for manatee in manatees:
        print manatee['name']

def example2(manatees):
    print manatees[0]['name']
    print manatees[0]['age']

def example3(manatees):
    for manatee in manatees:
        for manatee_property in manatee:
            print manatee_property, ": ",
manatee[manatee_property]

def example4(manatees):
    oldest_manatee = "No manatees here!"
    for manatee1 in manatees:
        for manatee2 in manatees:
            if manatee1['age'] < manatee2['age']:
                oldest_manatee = manatee2['name']
            else:
                oldest_manatee = manatee1['name']

    print oldest_manatee
```