

Comparing Methods of Python Education

Dr Kevin McCullen
Assistant Professor
Computer Science

Goals

Make quantitative comparisons between different methods of Python education delivery.

Compare a MOOC (Massive Online Open Course), a video series, and a traditional Introduction to Programming Course.

Method

We evaluated the material covered, the (approximate) amount of time required to complete the courses, and the course expectations.

Course expectations were measured using assignments and quizzes.

Defining CS1

Used the 2013 ACM Model Curriculum to define content for a CS1 course

SDF Sections:

Fundamental Programming Concepts:

- Basic Syntax / Semantics
- Variables and primitive data types
- Expressions and assignments
- Simple I/O including with files
- Conditionals and iteration
- Functions and parameter passing
- Recursion

Fundamental Data Structures:

- Arrays
- Records/Structs
- Strings and string processing

Development Methods:

- Program comprehension
- Program correctness
- Modern programming environments

PL Sections:

Object-Oriented:

- Classes, methods, constructors
- Inheritance

Event-Driven/Reactive Programming:

- Event and event handlers

Course Topics

CS1 Topic	MOOC	VIDEO	TRADITIONAL
What is Programming		X	X
Python Setup (Python, Idle, etc)	X	X	X
Variables	X	X	X
Expressions	X	X	X
Conditionals	X	X	X
Boolean Expressions	X	X	X
For Loops	X	X	X
While Loops	X	X	X
Lists	X	X	X
Functions	X	X	X
Parameters and Arguments	X	X	X
Scope	X	X	X
Strings	X	X	X
Tuples	X	X	X
Sets	X		X
Dictionaries	X	X	X
File I/O	X	X	X
Exceptions	X	X	X
Classes and Objects	X	X	X
Mutable vs Immutable	X	X	X
Python Libraries	X	X	X
Stacks and Queues		X	
Graphs and Trees		X	
Event Driven Programming		X	X
Turtle Graphics		X	X
Lambda Expressions/Map	X		
Decorators	X		
Generators	X		
GUI development	X		

References

- ACM, Computer Science Curriculum 2013
- Downey, A., Think Python, O'Reilly Media, 2015
- The Great Courses, *How to Program: Computer Science Concepts and Python Exercises*
- Gries, P., et al., Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2013
- Udemy, *Complete Python Bootcamp: Go from zero to hero in Python 3*

Conclusions

While neither the MOOC or Videos covered the material in the same depth as the traditional course, the MOOC compared favorably with the traditional course in content covered and expectations (quizzes, assignments, and projects).

MOOC

14 core sections, with “advanced” and “bonus” material sections. A total of 103 short videos (1 minute to 18 minutes in length – the maximum for TED talks).

Approximately 14 hours for all of the material. Short coding exercises, quizzes, and Jupyter notebooks containing examples and exercises. Three coding projects, two milestones projects, and a capstone.

Video Series

24 sessions of approximately 30 minutes each. Each lecture included exercises where you can pause the video and attempt code examples in PyCharm or Idle.

A small number of the code examples did not match the instructor description or contained errors.

Traditional

Three credit introduction to programming course with a required one credit laboratory. A total of 42 sessions over 14 weeks, approximately 50 hours of contact time. However, quizzes and exams are included in that total.

Covers material in Think Python, with some material from Practical Programming with Python.

The lecture section has ten homeworks, eleven quizzes, and three exams. The laboratory adds ten short programming assignments, ten longer projects, and two in-class practical exams.