Derek Larson

12/08/2023

IT FDN 110: Foundations of Programming-Python

Reflection Document

github.com/derkrylar99/IntroToProg-Python

Lessons Learned: Reflection

"What did I learn in this course that will help me going forward?

With increasing exposure to Python and Unreal Blueprint visual scripting logic in work projects, it became clear that without formal knowledge I would be stuck at the same level continuously struggling to self-learn concepts. Seeing first-hand how script-enabled automation, when applied to repetitive content-porting tasks or asset setup, directly benefits the user + project – especially relating to data stabilization – made me realize the great personal value of object-oriented programming.

Before starting the course, I had a basic understanding of how to create functions and had noticed certain oddities like the ability for variables to become any type, at any time, without declaration. However, I also had no clue how to convert functions to classes or even how to format a string.

The first several modules of the course were fundamental, delivering basic revelations like Python being an interpreted language (shedding light on old confusions) as well as providing a greater understanding of Python's history and relationship to other programming languages. Learning from the basics allowed me to process simple concepts like input and storage in ways that were completely relevant but also consistently overshadowed in the work environment by the urgency of deadline task deliveries.

By the time we came around to Classes and Objects, everything "clicked" and since then I have been actively creating new scripts being directly applied to work tasks, saving massive amounts of time while also providing more stability through validation testing (not exactly the unittest framework, but similar in concept). I am enabled to navigate and extend scripts built explicitly to link with others, can group/consolidate/integrate my older scripts into existing toolsets with Class functionality, and I even understand now why the files necessary to render our data always end with if __name__ == "__main__" : it's where we import and define all variables for scene generation and rendering before executing as the main script file!