Functional Programming in Python 3

Introduction to Functional Programming



Gerald Britton
IT Solutions Designer

@GeraldBritton www.linkedin.com/in/geraldbritton

Version Check



This version was created by using:

- Python 3.10
- Visual Studio Code 1.7
- Python extension for Visual Studio Code
  - v2022.16

Version Check



This course is 100% applicable to:

- Python 3.10 and above

Not Applicable



This course is NOT applicable to:

- Python releases prior to 3.10

Programming Paradigms



Imperative

- Machine code, Assembly language

Procedural

- Fortran, COBOL, Algol, Basic, C

Object Oriented

- Java, C++, C#, Smalltalk, Python

Declarative

- SQL, XML (HTML)

Logic

- Prolog, Planner

A Brief History of Functional Programming

1950s

McCarthy - Lisp

1960s

Iverson - APL

1970s

Backus' paper - FP

1970s

Milner - ML

1980s

Haskell

“Do it whenever it is convenient and think hard about the decision when it isn’t convenient”

John Carmack

Functional Programming Principles

- **First-class functions**
- **Pure functions**
- **Immutable variables and objects**
- **Lazy evaluation**
- **Recursion**
- **Pattern matching**
- **OOP included!**

Tools You Will Need

Python 3.10 or higher

- https://www.python.org/downloads/

An integrated development environment

- IDLE (included in Python download)
- PyCharm, Wing IDE, PyDev for Eclipse, Visual Studio
- Many others
- https://wiki.python.org/moin/PythonEditors

Visual Studio Code

- https://code.visualstudio.com/Download

Summary



- Functional programming
- History
- Major concepts
- Motivations
- Many languages perform actions
- FP is about computing results