* Course overview
  + High-level programming language
* Introduction
* Getting the Most from This Co..
* What is Python?
  + High level programming language
  + Dynamic type system
  + Readability
  + Interpreted language
  + Multi-paradigm
* Why Should Developers Care?
  + Beautiful is better than ugly
  + Explicit is better than implicit
  + Simple is better than complex
  + Simple Is Hard
  + Complex is better than complicated
  + Readability counts
  + Maintenance is Expensive
* What Makes Python Different?
  + Extensible design
    - Small and powerful
    - Embeddable scripting language
  + Community Involved Design
    - PEP(Python Enhancement Proposal)
  + Emphasizing Fun
  + Culture and Community
* Summary
* Introduction
* Linux Scripting and Administra…
  + Files, configuration, processes, applications, testing
  + Ex)
    - import json
    - with open(‘input.json’, ‘r’) as input:
    - obj = json.load(input)
    - print(‘Hello, ’ + obj[‘name’])
* Web Development
  + API(flask, bottle, pyramid)
  + Website(Django, turbogers, web2py)
  + App(plone, djano-cms, mezzanine)
  + Ex)
    - from flask import Flask
    - app = Flask(\_\_name\_\_)
    - @app.route(“/”)
    - def hello():
    - return “Hello World!”
  + Have route with parameter
  + Ex)
    - @app.route(“/user/<username>”)
    - def show\_user(username):
    - return “User: %s” % username
* Application Scripting
  + Blender is a free open source 3D modeling tool
* Data Science
  + Big Data and machine learning
  + Big Data
    - Kilobytes, megabytes, gigabyte, terabytes, petabytes, exabytes
  + Machine Learning
    - Algorithms that process data and find its own connection
    - Spam determination
    - Network intrusion detection
    - Optical character recognition
    - Computer vision
* Summary
* Introduction
* The Basics
  + Python.org
  + Installing python
  + Command-line installs
    - choco install python for windows
    - brew install python3 for mac
  + working with python
    - python
      * executing python file
      * rudimentary REPL(read, eval, print, loop)
    - pip
      * install 3rd party libraries(like npm, nuget, etc)
    - ipython
      * executable
      * robust interactive shell
* Learning and Documentation
  + Python: getting started
  + Python Fundamentals
* Working with the Code of Othe…
  + Pypi, where people published code to be used by others
  + Use pip to download the code
  + Pip
    - Install and uninstall packages
    - Track and installing and uninstalling dependencies
    - Package groups
    - Versions
  + Pip comes install by default
  + If you need to install pip
    - Curl <https://bootstrap.pypa.io/get-pip.py> | python
  + Install package with pip
    - pip install [SomePackage]
* Introduction
* Python 2 vs Python 3
  + Python 2
    - Pip, python, ipython
  + Python 3
    - Not backward compatible
    - Pip3, python3, ipython3
  + Default to python 3
  + Python 2.x is legacy
* Executing Python Code
  + Interpreter
    - Executing python file using python exe
  + REPL
    - Read, evaluate, print, loop
    - Call out to python code within interactive REPL
  + Natively
    - Compile and run(py2exe, pyinstaller, etc)
* Summary