Week 3 Notes

Day 1

Python

Data Analytics and Python –

*Programs* *and* *algorithms* – algorithm is a series of steps to solve a problem, a program is a series of instructions telling a computer how to solve a problem

*Python* – high level language, meaning mor natural language, takes less time to write, shorter and easier to read, and more likely to be correct and error free when code is ran

Two ways computers process program code –

*Interpreted* *languages* – interpreters process programs a little bit at a time, reading lines and performing computations

*Compiled* *Languages* – see slides for info

Python is an interpreted language – see slides for info

*Types of languages* – natural are mostly spoken; formal are designed by people like programming languages

**Vocab**

*VALUE* – number or string that can be stored in a variable or computed in an expression

*VARIABLE* – name that refers to a value

*Variable* Name – name given to variable

*Str* – python data type that holds a string of characters

*Operators* – special symbols that represent a simple computation

*Data* Type – set of values

*Comment* – information in a program that explains what certain code does

*Input* – command in a program that prompts the user to put in an answer

*Output* – result of the program

Graphical user interface, text

Description automatically generated

Different ways to code – Integrated Dev Environment IDE

Command line interpreter

*IDE* – *REPL* – Reads evaluates prints loops

*Python* *in* *VS* *Code* – integrated dev environment is an app that combines all the functions or REPL as well as an editor with which you can create and modify code to then submit to the interpreter for execution. Notable Features : syntax highlighting, context sensitive help, code completion, debugging

*Debugging* – process of detecting and removing existing and potential errors from software code, its an essential skill, both a science and an art, needs to be done as you code thinking of terms such as what could go wrong and how can I prevent it from happening, complexity is ruins efficiency, issues could be large or small

Tips – regularly test and review codes, chunk code into smaller sections and ensure each section works, explain your code aloud to yourself barney style, work backwards to try finding the issue, take a break and come back to it when you run out of ideas, ask someone to help look

**Common** **Errors**

*Syntax* – mistake in code like spelling punctuation spacing and indentation

*Runtime* – occur during running or execution, might run indefinitely or crash

*Semantic* – code is typically grammatically correct, but its not coming out with the expected results

*Logical* – do not accomplish intended goal, calculations are off

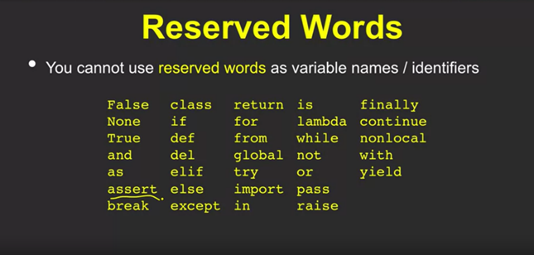
Jupyter Notebook – open source web application used to create and share documents that contain : live code, equations, visualizations, text; browser based tool for interactive authoring of documents combining text, mathematics, code, computations, and rich media output

Notebook Documents: a representation of ----on video

Python Data Types and Variables - - - **alt+/ comments out selected code len() prints length .index returns index number**

Variables – assigned data values; requirements – has to start with a letter or \_, can only contain alpha numeric characters and underscores, case sensitive, caps used to notate global variables

Reserved words – cannot be assigned values



Assigning multiple values at once a, b, c, d, e = a, b, c, d, e

Global Variables and scope – global variables hold their value throughout the lifetime of the program

Table

Description automatically generated

Casting Data types – assigning data types –

a = 55

a = float(a)

Day 3

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generatedTable

Description automatically generated

Timeline

Description automatically generated

Complex data types

List – values separated by commas and enclosed in square brackets[]; have strict order but can be modified, have any number of duplicates, contain any data types

Diagram

Description automatically generatedCopying List – one could use the ‘=’ by assigning the old list a new list, .copy() returns a shallow copy of the list and doesn’t modify the original; deepcopy() is the safest option for making a true independent copy of the compound list object

Clear and delete lists – clear(), pop(), remove(), del

Tuple cannot be changed, storm ult items of any type, empty tuple ()Timeline

Description automatically generated

Graphical user interface, application

Description automatically generated

Any returns true if any bool = true. Like the or function

All is like the and, will return true if all is true

Sorted is sort, does not rearrange the tuple itself

Sets – unique collection that is unordered, unchangeable and unindexed , must be the same data type, cannot have duplicates uses {}

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Table

Description automatically generatedChart, box and whisker chart

Description automatically generatedDictionaries – collection that is ordered changeable does not allow duplicate,