Python Review
Intro to Git
Eric J Nguyen
Python Overview

- Based on Python 3.4.4 specifications
- Assumed that you know Python
- Derived from Dr. Mitra’s CS 303E Notes
Conventions

● Use a plain text editor
  ○ Notepad++, TextWrangler, Sublime Text, etc.
  ○ Don’t use Notepad (Win) or TextEdit (Mac)!

● Spaces, not tabs
  ○ Google “Indent using spaces in <TextEditor>”
  ○ Indents matter in Python

● Unix line endings
  ○ Google “Unix line endings in <TextEditor>”
Hello World

# This is a comment
# This is a basic Hello World program

print("Hello World!")
Variables

- Start with a letter or underscore
- Contains letters, digits, or underscores

\[
x = 1
\]
\[
y = 2
\]
Operators

- **Arithmetic**
  - +    -    *    /    //    %    **

- **Comparison**
  - ==    !=    >    >=    <    <=
User Input

nameStr = input("Name: ")
age = eval(input("Age: "))
Functions

def functionName():
    return 7

def functionName2(arg1, arg2):
    return arg1 + arg2
Conditionals

```python
if (someCondition):
    ..do stuff..
else:
    ..do other stuff..
```
Loops

for i in range(0,10):
    print(i)

break and continue
String

Ordered list of characters

```python
string = "Hello World!"
print(string)
print(string[1])
print(string[1:])
print(string[::2])
```
File Input

```
inFile = open("inputFile.txt","r")

for line in inFile:
    print(line.rstrip("\n"))

inFile.close()
```
File Output

outFile = open("outputFile.txt","w")
outFile.write("Hello World!\n")
outFile.close()

# To Append
outFile = open("outputFile.txt","a")
outFile.write("Hello World 2!\n")
outFile.close()
List

Basic ordered data structure

```
l = ["a", "b", "c", 2, 3, 4]
for i in range(len(l)):
    print(l[i])
for e in l:
    print(e)
```
Dictionary

Key-value store

```python
phoneNums = {}
phoneNums['Eric'] = '512-555-0123'
phoneNums['Daniel'] = '512-555-0124'
phoneNums['Eric'] = '512-555-0000'
```
Python Questions?
Git Overview

● Version Control
  ○ Managing different versions of code
  ○ See the code “history”
  ○ Everything is “local” until you push to “remote”
  ○ GitHub is our “remote”

● Collaboration
  ○ Many people working on the same code at once
  ○ Merge the changes later
  ○ No need to email/Dropbox the “latest version” of code
Git Terms

- **Working Directory**: current folder of actual files
- **Commit**: single set of code changes
  - Uniquely identified by a SHA1 hash (fingerprint)
- **Branch**: ordered set of commits
  - Timeline of code changes
- **Repository**: set of branches
Starting Out

- Create your repository ("repo")
- Add your code
- Make your first commit
- Push your repo
Workflow

- Make your changes
- Commit
- Pull and merge from remote (sometimes)
- Push
Merge Conflicts

- Can’t decide how to resolve two conflicting changes
- Manually resolve the conflict and commit the change
- Push
More Advanced Stuff

- Use command line `git`
  - Feel free to use either GitHub Desktop or command line `git`
Git Questions?