Hands-on Python Foundations

Day 2:

More useful scripts

Outline

Today's topics

- Lists
- Dictionaries
- Tuples and sets
- Exceptions
- Files
- Requests
- APIs
- Command line overview
- Script arguments

Interactive Scenarios – Day 1

1. Data structures: lists, dictionaries + more

- https://learning.oreilly.com/scenarios/hands-on-python-foundations/9780137904648X004/

2. Exceptions and file handling

- https://learning.oreilly.com/scenarios/hands-on-python-foundations/9780137904648X005/

3. Requests and APIs

- https://learning.oreilly.com/scenarios/hands-on-python-foundations/9780137904648X006/

Schedule

0:00 - Review homework, Q&A

0:20 - **Data structures** - Interactive scenario #4

1:15 - *Break* (15 mins)

1:30 - **Exceptions and file handling** - Interactive scenario #5

2:30 - *Break* (15 mins)

2:45 - **Requests and APIs** - Interactive scenario #6

3:30 - Command line Python

3:50 - Intro to homework (5 min)

3:55 - Quiz (5 min)

Questions and breaks

- I'll answer attendee chat throughout class
- Q&A widget during the breaks

- 2 Breaks (15 mins each)

- Email more in-depth questions at arianne.dee.studios@gmail.com

Poll #1

- How did you feel about the material from last class?
  - It was too much, I was (and still am) overwhelmed
  - It was a lot, but I reviewed it and am following along
  - It was a good pace and I feel prepared for this class
  - It was basic, looking forward to this class more
  - I didn't attend last class

Poll #2 (multi-choice)

- Did you do any work to prep for this week?
  - I attempted some homework
  - I completed the homework
  - I went through some interactive scenarios
  - I watched some videos
  - I did some practice problems
  - I used Stack Overflow/Google/documentation
  - I looked at other resources (put useful links in chat)
  - Other (say in chat)

Poll #3 (multi-choice)

- Which APIs are you looking forward to using?
  - ❑ N/A, I'm not doing the project
  - ❑ Weather
  - ❑ Joke/quote/comic of the day
  - ❑ Todo list/reminders/events
  - ❑ News
  - ❑ Sports
  - ❑ Stocks/finance
  - ❑ Other (say in chat!)

Automated daily email

Good morning, Arianne 👋

Today's weather ☁️

Today there will be light snow.
High: 5.9 °C (42.6 °F)
Low: -0.9 °C (30.3 °F)

☀️ 7:48 AM
🌙 5:16 PM

Joke of the day 🤣

Little Johnny comes home from his first day of school. His mother asks, "So, what did you learn at school today?" Little Johnny replies, "NOT ENOUGH. They want me to come back tomorrow!"

Quote of the day 💬

When you recover or discover something that nourishes your soul and brings joy, care enough about yourself to make room for it in your life.
~ Jean Shinoda Bolen

Next game 🏀

In 2 days @ 7:30 PM ET
Home: Toronto Raptors
Away: Atlanta Hawks

Project

- Find some free APIs to retrieve data from
  - weather
  - quote/joke of the day
  - calendar, reminders, todos
  - sports, stocks, events

- Send email every morning

- Basic – plain text email
- Advanced – HTML (formatted) email

Homework review

Homework – week 1

1. Copy your **greeting.py** file into **homework/homework_1.py**

2. Replace the multiple print statements with a ***multi-line f-string***; this will be your email message content

3. Move the temperature conversion to a function

4. Add content for the APIs that you want to use, with data:
   - Hardcoded
   - Input from the user
   - Random

Homework – week 1

2. Work on **problem_3_collatz.py**

Supplemental Videos Lessons

- **Week 1 review**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_01_02/

- **Work with dates and times**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_01_05/

- **Use regular expressions** (string pattern matching)
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_01_06/

Supplemental Videos Lessons

- **Understanding Git and GitHub**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_03_04/
- **Debugging**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_06_01/
- **Testing**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_06_02/

Supplemental Videos Lessons

- **Learn PyCharm shortcuts**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_04_04/
- **Scraping websites**
  - https://learning.oreilly.com/videos/next-level-python/9780136904083/9780136904083-NLP1_01_07_03/
- **Intro to data analysis**
  - https://learning.oreilly.com/videos/introduction-to-python/9780135707333/9780135707333-INPY_01_06_00/

Review

Covered Keywords – Week 1

False	await	**else**	**import**	pass
None	**break**	except	**in**	raise
True	class	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	**elif**	**if**	**or**	yield

Types

Name	type	Example
Integer	`int`	1
Float	`float`	1.5
String	`str`	'1'
Boolean	`bool`	**True, False**
None	`NoneType`	**None**

Covered Keywords - Types

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Variables

```
my_variable_1 = 'hello'
del my_variable_1
```

- Only letters, numbers, underscores
- Can't start with a number

- Value can change type
- If it doesn't exist, you get a NameError

Covered Keywords - Variables

False	await	**else**	**import**	pass
None	**break**	except	**in**	raise
True	class	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	**elif**	**if**	**or**	yield

Importing modules

- Lots of built-in modules you can use

```
import math
math.sqrt(25)
```

- Can import specific function or variable

```
from math import pi, sqrt
math.sqrt(25) * pi
```

Covered Keywords - Imports

False	await	**else**	**import**	pass
None	**break**	except	**in**	raise
True	class	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	**elif**	**if**	**or**	yield

Functions

```
def add(n1, n2):
    return n1 + n2
```

- Your arguments can have default values

```
def greet(name='world'):
    print('Hello, ' + name)
```

- If you don't specify a **return** value, it returns **None**

Covered Keywords - Functions

False	await	**else**	**import**	pass
None	**break**	except	**in**	raise
True	class	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	**elif**	**if**	**or**	yield

Conditions

```
True and not True
```

- **and** – Both must be True
- **or** – One must be True
- **not** – Negates the value (True → False, False → True)

```
'i' in 'team'
```

- **in** – Checks if an item is contained in a string or data structure

Covered Keywords - Conditions

False	await	**else**	**import**	pass
None	**break**	except	**in**	raise
True	class	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	**elif**	**if**	**or**	yield

Conditionals

```
if x <= 0:
   print('freezing')
elif x >= 100:
   print('boiling')
else:
   print('liquid')
```

- Must start with 1 **if**
- Can have 0, 1, or many **elif**
- Can have 0 or 1 **else**

Covered Keywords - Conditionals

False	await	else	**import**	pass
None	**break**	except	**in**	raise
True	**class**	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	elif	if	**or**	yield

While-loops

```
x = 0
while x < 10:
    print(x)
    x += 1
```

- Loop until condition (x < 10) is False

- If **break** reached, stop looping
- If **continue** reached, stop the current loop but keep looping

For-loops

```
for i in range(10):
    print(i)
```

- Loop for each item in a sequence
- The range(n) function returns a sequence 0, 1, 2, ..., n-1

- If **break** reached, stop looping
- If **continue** reached, stop the current loop but keep looping

Covered Keywords - Loops

False	await	**else**	**import**	pass
None	**break**	except	**in**	raise
True	class	finally	**is**	**return**
and	**continue**	**for**	lambda	try
as	**def**	**from**	nonlocal	**while**
assert	**del**	global	**not**	with
async	**elif**	**if**	**or**	yield

Week 2 topics

Today's topics

- ❑ Lists
- ❑ Dictionaries
- ❑ Tuples and sets
- ❑ Exceptions
- ❑ Files
- ❑ Requests
- ❑ APIs
- ❑ Command line overview
- ❑ Script arguments

Covered Keywords – Week 2

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Covered Keywords – Week 3

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Data structures

Today's topics

- ❑ **Lists** ←
- ❑ Dictionaries
- ❑ Tuples and sets
- ❑ Exceptions
- ❑ Files
- ❑ Requests
- ❑ APIs
- ❑ Command line overview
- ❑ Script arguments

Interactive Scenario # 4

4. Data structures: lists, dictionaries, tuples and sets

- https://learning.oreilly.com/scenarios/hands-on-python-foundations/9780137904648X004/

Lists

- Create a list
```
my_list = ['a', 'b', 'c']
```
- Get an item by index
```
my_list[0]
```
- Get a sublist (slice)
```
my_list[0:2]
```
- Update a list item
```
my_list[0] = 'A'
```
- Add an item to the end
```
my_list.append('d')
```

Lists

- Ordered
  - Access items by index
- Can have duplicates
- Mutable (can be modified)

Today's topics

- ☑ Lists
- ☐ **Dictionaries** ←
- ☐ Tuples and sets
- ☐ Exceptions
- ☐ Files
- ☐ Requests
- ☐ APIs
- ☐ Command line overview
- ☐ Script arguments

Dictionaries

- Create a dict
```
my_dict = {'a': 1, 'b': 2}
```
- Get an item by key
```
my_dict['a']
```
- Get an item (with default)
```
my_dict.get('a', None)
```
- Update a value
```
my_dict['a'] = 0
```
- Add an item
```
my_dict['c'] = 3
```

Dictionaries

- Not ordered
  - Access items by key

- Cannot have duplicate keys
- Can have duplicate values

- Mutable (can be modified)

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☐ **Tuples and sets** ←
- ☐ Exceptions
- ☐ Files
- ☐ Requests
- ☐ APIs
- ☐ Command line overview
- ☐ Script arguments

Tuples

- Create a tuple

```
my_tuple = ('a', 'b', 'c')
```

- Get an item by index

```
my_tuple['a']
```

Tuples

- Ordered
  - Access items by index
- Can have duplicates
- Immutable (cannot be modified)

Sets

- Create a set

```
my_set = {'a', 'b', 'c'}
```

- Add an item

```
my_set.add('d')
```

- Get and remove an item

```
my_set.pop()
```

Sets

- Not ordered
  - Access items by key

- Cannot have duplicate keys
- Can have duplicate values

- Mutable (can be modified)

Data structures

Name	type	New object	Get item	Main features
List	`list`	[1, 2, 3]	a_list[0]	Ordered Duplicates
Dictionary	`dict`	{'a': 1, 'b': 2, 'c': 3}	a_dict['a']	No order No duplicate keys
Tuple	`tuple`	(1, 2, 3)	a_tuple[0]	Ordered Duplicates Immutable
Set	`set`	{1, 2, 3}	a_set.pop()	No duplicates No order

problem_4_wordle.py

Write a function that returns how similar a word is to a guess

- 💚 – letter in correct position
- 💛 – letter in wrong position
- 🖤 – letter not in word

test_wordle.py

- pip install pytest
- Set PyCharm test runner to pytest
  - Settings > Tools > Integrated Tools > Default test runner > pytest
- Try to make all tests pass

Covered Keywords – Tests

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Testing functions

```
def a_function(n):
    pass

assert a_function(1) == 1
```

- Can create function "stubs"
  - Create function signature (first line)
  - Use **pass** in body to create empty function

- Use **assert** keyword with **pytest** to test functions

Class setup

- Create a GitHub account
  - https://github.com/

- React or post comments on GitHub Issue #3 with the APIs you will be using for your project

Q&A and 15 min break

Interactive Scenario #5

5. Exceptions and file handling

- https://learning.oreilly.com/scenarios/hands-on-python-foundations/9780137904648X005/

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☐ **Exceptions** ←
- ☐ Files
- ☐ Requests
- ☐ APIs
- ☐ Command line overview
- ☐ Script arguments

Covered Keywords – Exceptions

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Exceptions

```
try:
    temp = int(a_str)
except ValueError:
    print('Invalid number')
```

- Can have multiple except cases for different exception types

More exceptions

```
try:
    temp = int(a_str)
except ValueError as e:
    raise CustomError(e)
else:
    print('Valid number')
finally:
    print('Always gets here')
```

- Retrieve error with **as**
- Throw errors with **raise**

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☑ Exceptions
- ☐ **Files** ←
- ☐ Requests
- ☐ APIs
- ☐ Command line overview
- ☐ Script arguments

Covered Keywords – Files

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Working with files

```
with open('filename.txt', 'r') as file:
    file.read()
```

- Use **with** so that file is closed when outside code block
- Different modes:
  - r - read
  - w - write
  - a - write by appending to end of file
  - t - text files
  - b - binary files (e.g. images)

problem_5_wordle.py

- Handle errors in word length
- Choose the first word from a text file
- Remove the word from the file after a game ends

15 min break

Interactive Scenario #6

6. Requests and APIs

- https://learning.oreilly.com/scenarios/hands-on-python-foundations/9780137904648X006/

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☑ Exceptions
- ☑ Files
- ☐ **Requests** ←
- ☐ APIs
- ☐ Command line overview
- ☐ Script arguments

Requests

```
import requests
response = requests.get('http://test.com')
if response.status_code == 200:
    print(response.text)
```

- Status codes:
  - 2XX – Success
  - 4XX – Client error (you did something wrong)
  - 5XX – Server error (something's wrong but out of your control)
- **response.text** gets content as a string
- **response.json()** gets content as Python dict/list

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☑ Exceptions
- ☑ Files
- ☑ Requests
- ☐ **APIs** ←
- ☐ Command line overview
- ☐ Script arguments

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☑ Exceptions
- ☑ Files
- ☑ Requests
- ☑ APIs
- ☐ **Command line overview** ←
- ☐ Script arguments

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☑ Exceptions
- ☑ Files
- ☑ Requests
- ☑ APIs
- ☑ Command line overview
- ☐ **Script arguments** ←

problem_6_wordle.py

- Check to make sure that guesses are valid words

Command Line

Common actions

Action	Mac/Linux	Windows
List the folders and files in current location	ls	dir
Display your current location	pwd	pwd
Change directory/folder in the file system.	cd <*pathname*>	cd <*pathname*>
Move one level up (one folder) from current location	cd ..	cd ..

Work-along

1. Open up a terminal in PyCharm or VS Code
2. Print the current directory
   - **`pwd`**
3. List the files and folders located there
   - **`ls`** or **`dir`**
4. Navigate into the **examples/week_2** folder in the codebase
   - **`cd examples/week_2`**
5. Run example_20_args.py
   - **`python3 example_20_args.py abc 123`**

Other common actions

Action	Mac/Linux	Windows
Copy a file to another folder	cp	copy
Move a file to another folder	mv	move
Create a new directory (folder)	mkdir	mkdir or md
Delete a file or directory	rm	del
Display the contents of a file	cat *<filename>*	type *<filename>*
Clear the window	clear	cls
Show the manual for a command	man *<command>*	help *<command>*

Adding args to our scripts

- **example_23_args.py**
  - Retrieve argument list via **`sys.argv`**

- **example_24_click.py**
  - Use the Click package to create command line applications
  - Supports
    - typed arguments
    - options
    - flags
    - input

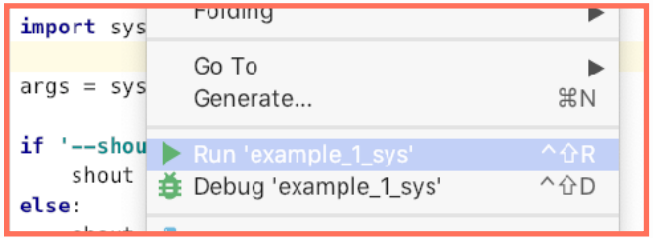
Click

- External package
  - `pip install click`

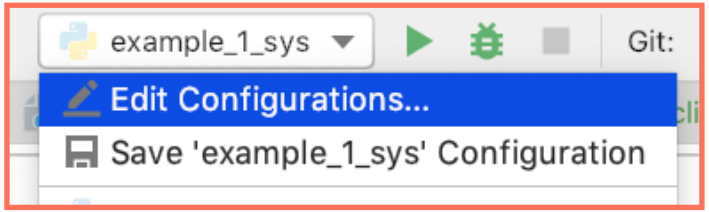
- "**C**ommand **L**ine **I**nterface **C**reation **K**it"
- Create beautiful interfaces with as little code as necessary
- Can package your tool to be installed on other computers
- Supports command groups:
  - `pip install` vs `pip uninstall`

- Documentation: https://click.palletsprojects.com/en/7.x/

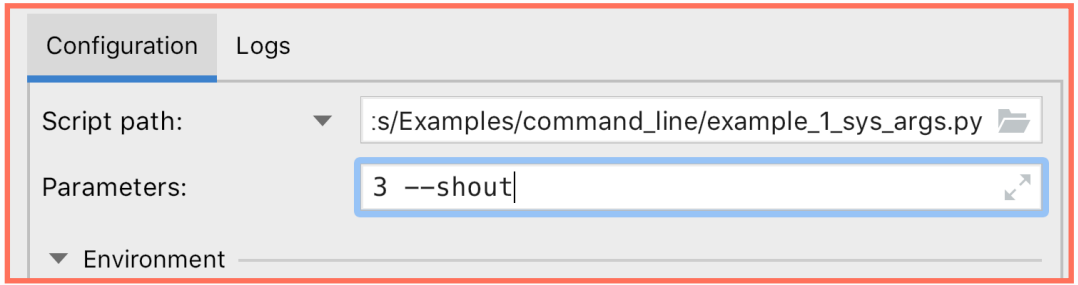
In PyCharm's Run



1. Right click and run the file



2. In top bar, click dropdown and select Edit Configurations…



3. Add arguments to the "Parameters" field

Today's topics

- ☑ Lists
- ☑ Dictionaries
- ☑ Tuples and sets
- ☑ Exceptions
- ☑ Files
- ☑ Requests
- ☑ APIs
- ☑ Command line overview
- ☑ Script arguments

Next week's topics

- ❑ Pip
- ❑ Virtual environments
- ❑ Setting up projects
- ❑ Classes
- ❑ Dunder methods
- ❑ Modules and packages
- ❑ Code hosting

Covered Keywords – Week 3

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Course Project

Automated daily email

Good morning, Arianne 👋

Today's weather ☁️

Today there will be light snow.
High: 5.9 °C (42.6 °F)
Low: -0.9 °C (30.3 °F)

☀️ 7:48 AM
🌙 5:16 PM

Joke of the day 🤣

Little Johnny comes home from his first day of school. His mother asks, "So, what did you learn at school today?" Little Johnny replies, "NOT ENOUGH. They want me to come back tomorrow!"

Quote of the day 💬

When you recover or discover something that nourishes your soul and brings joy, care enough about yourself to make room for it in your life.
~ Jean Shinoda Bolen

Next game 🏀

In 2 days @ 7:30 PM ET
Home: Toronto Raptors
Away: Atlanta Hawks

Homework

- Create email content text

Week 2 homework

1. Copy your code from **homework_1.py** to **homework_2.py**
2. Accept the name as a script argument instead of input
3. Retrieve weather (and more) data from API
4. Retrieve some data from a file (todo/reminder/other)
5. Create a **new Gmail account** to send your emails from

Week 3 homework

- Create a virtual environment for the project
- Send a text email (automated)
- Refactor into modules and functions (and classes)
- Optional: Send an HTML formatted email
- Optional: Host code on Python Anywhere

Quiz

- Run **quizzes/quiz2.py** and answer it in the console

- There are links to relevant video lessons you can review if you get a question wrong

More practice

- Improve your practice problem from week 1
  - Handle invalid inputs
  - Retrieve data from files
- More in **practice/week_2/** folder
- You may need to do more advanced searches or follow simple tutorials

Video references for week 1

- **Running Python** – Intro to Python lesson 1.1
- **First code** – Intro to Python lesson 1.2
- **Variables** – Intro to Python lesson 2.2
- **Types** – Intro to Python lesson 2.1
- **Strings** – Intro to Python lesson 2.7
- **Functions** – Intro to Python lesson 2.6
- **Conditions** – Intro to Python lesson 3.2
- **Conditionals (if else)** – Intro to Python lesson 3.3
- **While loops** – Intro to Python lesson 4.1
- **For loops** – Intro to Python lesson 4.4

Video references for week 2

- **Lists** – Intro to Python lesson 4.3
- **Dictionaries** – Next Level Python lesson 1.3
- **Sets, tuples** – Intro to Python lesson 5.2
- **Exceptions** – Next Level Python lesson 1.4
- **Reading files** – Next Level Python lesson 2.1
- **Writing files** – Next Level Python lesson 2.2
- **CSV files** – Next Level Python lesson 2.3
- **Requests** – Next Level Python lesson 7.1
- **API requests** – Next Level Python lesson 7.5
- **Command line** – Next Level Python lesson 3.1

Supplemental videos

- **Week 1 review** – Next Level Python lesson 1.2
- **Dates and times** – Next Level Python lesson 1.5
- **Regular expressions** – Next Level Python lesson 1.6
- **HTML overview** – Next Level Python lesson 7.2
- **Scraping websites** – Next Level Python lesson 7.3
- **Git and GitHub**– Next Level Python lesson 3.4
- **Debugging** – Next Level Python lesson 6.1
- **Testing** – Next Level Python lesson 6.2
- **Intro to data analysis** – Intro to Python lesson 6