SI 506: Programming I
Fall 2019
Lecture 18

Anthony Whyte <arwhyte@umich.edu>
Lecturer III, School of Information
715 N. University Ave, Ann Arbor, MI 48109
Roumanis Square, 2nd floor ("the loft")

UMSI

preliminaries

UMSI

Exercises

paths

lectures/lecture_18/

lecture_18_exercise.py

lecture_18_exercise_solution.py

un-americas_country_areas_dev.csv

un-americas_country_areas.csv

un-americas_regions_country_areas.csv

world_bank_americas_economies.csv

Data sets

UN regional, country/area data, and World Bank data

Countries/areas comprising the Americas

- Northern America
- Caribbean
- Central America
- South America

UMSI

Contract

Deliverables

1. Process four *.csv files of UN/World Bank country/area data. Retain special characters.
2. Return counts of country/areas by region (data structure specified).
3. Write counts to *.json file.
4. Combine individual country/area data (data structure specified, exclude null values).
5. Write combined country/area data to *.json file.

Deliverables

region counts


```
{
  "regions": {
    "americas": 57
  },
  "subregions": {
    "latin_america_and_the_caribbean": 52,
    "northern_america": 5
  },
  "intermediate_regions": {
    "caribbean": 28,
    "central_america": 8,
    "south_america": 16
  }
}
```

Deliverables

new record (semi-structured)


```
[
  . . .
  {
    "country_area": "Brazil",
    "iso_alpha3_code": "BRA",
    "locations": {
      "region": "Americas",
      "subregion": "Latin America and the Caribbean",
      "intermediate_region": "South America"
    },
    "development_status": {
      "un_classification": "Developing",
      "world_bank_classification": "Upper middle income economy"
    }
  },
  . . .
]
```

Deliverables

new record (semi-structured)

```
[
  . . .
  {
    "country_area": "Canada",
    "iso_alpha3_code": "CAN",
    "locations": {
      "region": "Americas",
      "subregion": "Northern America"
    },
    "development_status": {
      "un_classification": "Developed",
      "world_bank_classification": "High income economy"
    }
  },
  . . .
]
```

no intermediate region

Deliverables

new record (semi-structured)

```
[
  . . .
  {
    "country_area": "Saint Barthélemy",
    "iso_alpha3_code": "BLM",
    "locations": {
      "region": "Americas",
      "subregion": "Latin America and the Caribbean",
      "intermediate_region": "Caribbean"
    },
    "development_status": {
      "un_classification": "Developing"
    }
  },
  . . .
]
```

retain special characters

no World Bank economic classification

UMSI

Data set 1

UN ‘Americas’ countries/areas

file name: un-americas_country_areas.csv
file type: comma-delimited text file (.csv)
delimiter: comma (',')
row count: 58
header row: yes
row values: country/area, ISO Alpha3 code
character set: UTF-8
special characters: yes
description: countries/areas comprising the Americas.
source: UN Statistics Division M49

UMSI

Data set II

UN ‘Americas’ countries/territories development status

file name: un-americas_country_areas_dev_status.csv
file type: comma-delimited text file (.csv)
delimiter: comma (',')
row count: 58
header row: yes
row values: country/area, development status
character set: UTF-8
special characters: yes
description: countries/areas comprising the Americas classified as either developing or developed.
source: UN Statistics Division M49

UMSI

Data set III

UN ‘Americas’ regions and countries/areas

file name: un-americas_regions_country_areas.csv
file type: comma-delimited text file (.csv)
delimiter: comma (',')
row count: 58
header row: yes
row values: region, subregion, intermediate region, country/area
character set: UTF-8
special characters: yes
description: regional affiliations for countries/areas comprising the Americas. Not all country/areas assigned to an intermediate region.
source: UN Statistics Division M49

UMSI

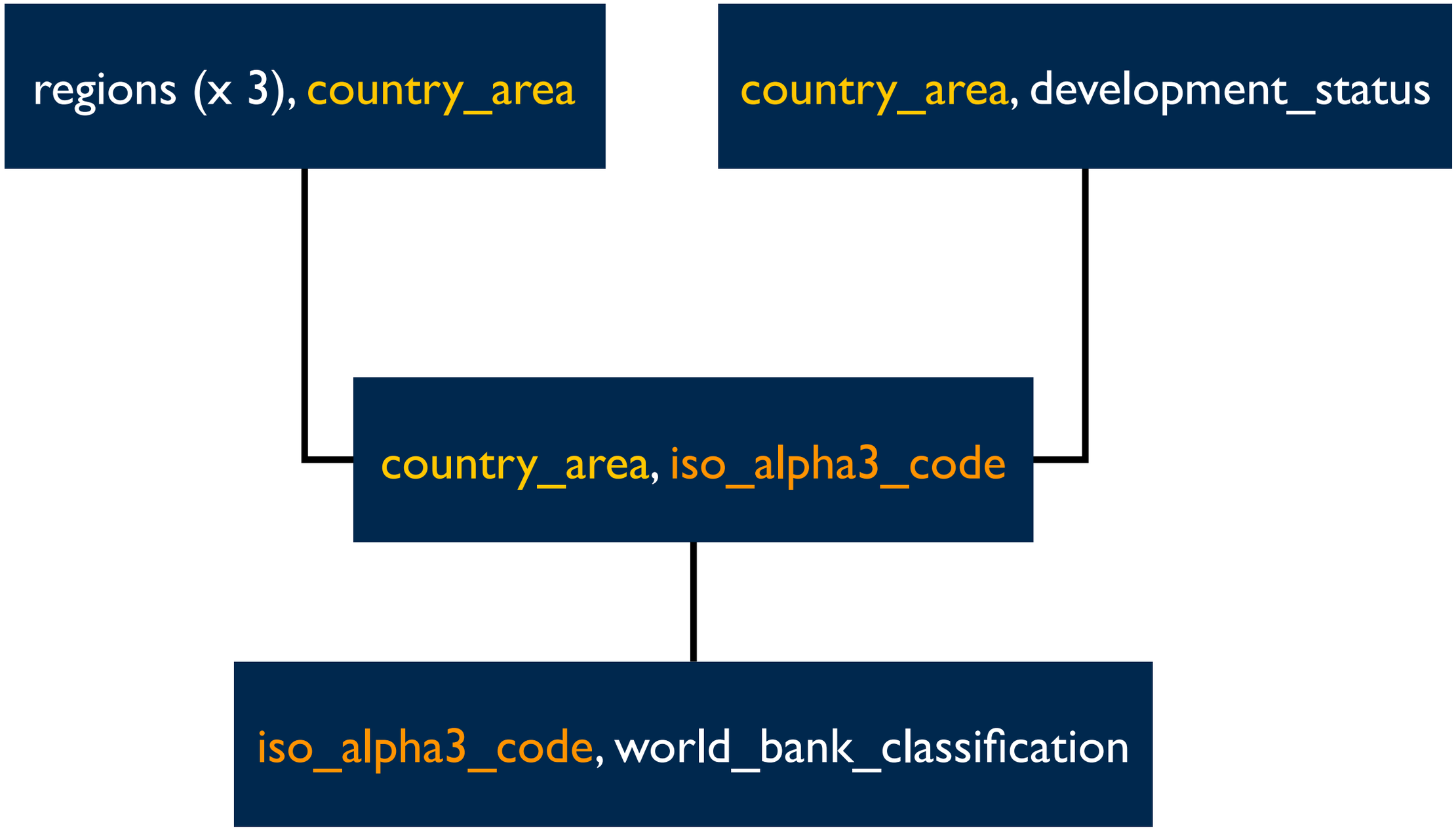
Data set IV

World Bank country/area economy classification

file name: world_bank-americas_economies.csv
file type: comma-delimited text file (.csv)
delimiter: comma (',')
row count: 58
header row: yes
row values: ISO Alpha 3 code, economic classification
character set: UTF-8
special characters: yes
description: national economies in the Americas classified as high income, upper middle income, lower middle income, or low income.
source: UN Statistics Division M49

UMSI

Data
connections



UMSI

lecture_18_exercise.py

script anatomy

load

```
import csv
import json
```

entry

```
CONSTANTS
```

```
main()
```

process, delegate tasks

call

```
create_record(<5 parameters>)
format_keys(string)
get_region_count(data, region)
get_country_area_regions(regions, headers, country_area)
get_dev_status(dev_status, country_area)
get_economic_status(ec_status, iso_code)
read_csv(filename, seq_type=SEQ_TYPE[0])
write_json(filename, data)
```

```
if __name__ == '__main__':
    main() # call main method
```

CONSTANTS

putting a tuple to good use

```
INTERMED_REGIONS = ('Caribbean',
                    'Central America',
                    'South America’
                   )
```

```
ca_count = get_region_count(regions, INTERMED_REGIONS[1])
```


UMSI

Why create a function?

enhance modularity, reusability, avoid duplicate code

```
for country_area in country_areas:

    . . .

    for region in regions:
        if region[-1] == country_area_name:
            locations = {region_headers[0]: region[0],
                         region_headers[1]: region[1]}
            if region[2]:
                locations[region_headers[2]] = region[2]
            new_record['locations'] = locations
            break
```

Yuck!

UMSI

Function: get regional affiliations

returns regional affiliations

```
def get_country_area_locations(regions, headers, country_area):
    """Return country or area UN regional associations.
    WARN: N. America has no intermediate regions defined."""
    for region in regions:
        if region[-1] == country_area:
            locations = {headers[0]: region[0],
                         headers[1]: region[1]}
            if region[2]:
                locations[headers[2]] = region[2]

            return locations

    return None  # render explicit what is implicit
```

no break statement required

exercise

finis

directors cut

UMSI

VS Code

opening files from a script

1. use absolute paths

```
import os

this_dir = os.path.dirname(os.path.abspath(__file__))
```

os.path figures out the current folder that points to the file you're running for any OS you're on.

```
file_to_open = os.path.join(this_dir, ‘somefile.csv’)
```

2. change the `python.terminal.executeInFileDir`
   1. Search for “`python.terminal.executeInFileDir`”
   2. Check the check box (change vales from False to True
   3. Close settings window

UMSI

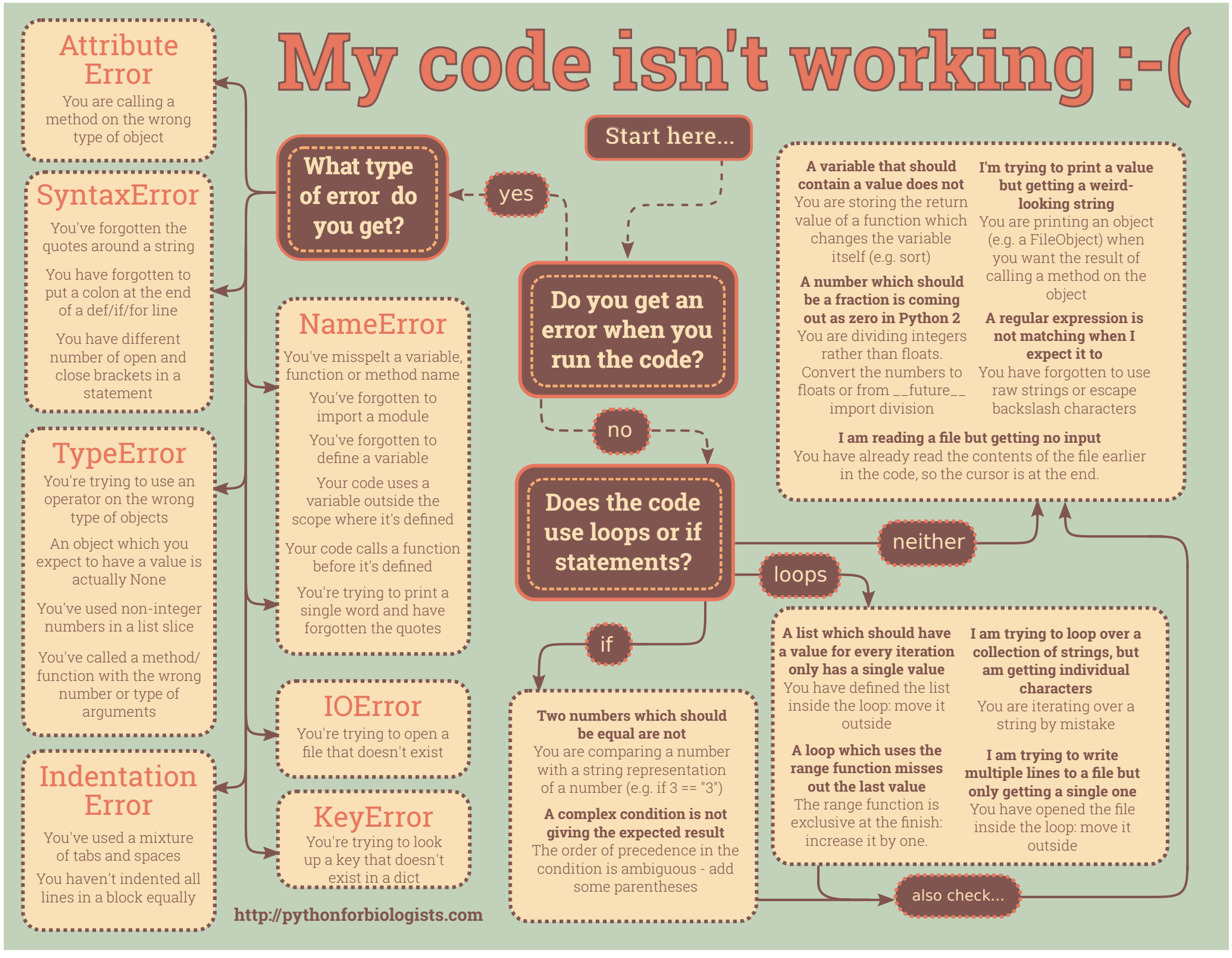
Tuples

data type characteristics

- sequence: siblings: list, string
- ordered: access items via indexing, slicing
- immutable: item reassignment not permitted
- comparable: comparison operator friendly
- hashable: can be used as dictionary keys

When your code misbehaves

debug flowchart



UMSI

Slide deck revisions

errata: corrections and other changes

Slide no(s).	Fix ver.	Description
	v1p1	