CS 115, Autumn 2021

Lecture 31: files; advanced editors

&&	!alive	alive
!dead		
dead		

In many other languages:

- `&&` is used for `and`
- `!` is used for `not`

Thanks to Marty Stepp and Stuart Reges for parts of these slides

Gas prices question

- Write a program that reads a file `gasprices.txt`
  - Format: *Belgium $/gal US $/gal date …*

```
8.20 3.81 3/21/11 8.08 3.84 3/28/11 ...
```

- The program should print the average gas price over all data in the file for both countries:

```
Belgium average: 8.3
USA average: 3.9
```

Multiple tokens on one line

You can use `read()` to read the whole file into a string and the `split` function to break a file apart

- **str**`.split()` – splits a string on blank space
- **str**`.split(`**other_str**`)` – splits a string on occurrences of the other string

```
>>> f = open("hours.txt")
>>> text = f.read()
'1 2\n45 6\n'

>>> f = text.split()
['1', '2', '45', '6']
```

Looping through a file

- The result of `split` can be used in a `for` ... `in` loop
- A template for reading files in Python:

```
file = open("filename")
text = file.read()
text = text.split()
for line in text:
        statements
```

Gas prices solution

```
def main():
    file = open("gasprices.txt")
    belgium = 0
    usa = 0
    count = 0
    lines = file.read().split()

    for i in range(0, len(lines), 3):
      belgium += float(lines[i])
      usa += float(lines[i + 1])

    print("Belgium average:", (belgium / count), "$/gal")
    print("USA average:", (usa / count), "$/gal")
```

Birthdays

- Write a program that prints the number of people with birthdays in each month of the year.
  - What month has the most birthdays?
  - What we have: a file with a month, day and number of births for each day of the year

```
1 1 8494
1 2 8049
1 3 8177
1 4 10202
1 5 10927
1 6 10669
1 7 10814
1 8 10748
1 9 8645
1 10 8299
1 11 10652
```

Advanced Editors

- So far in this class we have used IDLE
  - Advantages:
    - Comes with Python so nothing else to install
    - Pretty simple to use
  - Disadvantages:
    - It does not have many features
    - It is not convenient for big projects with lots of files

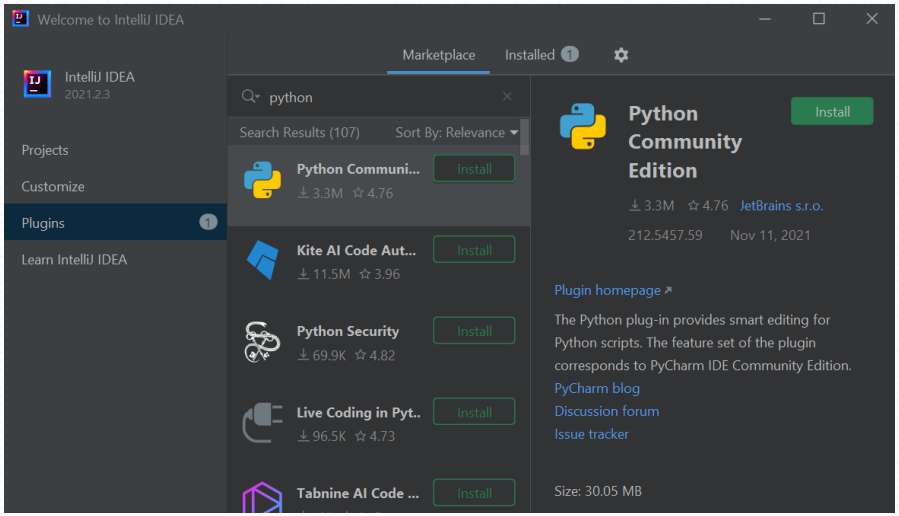
Advanced Editors

- There are a huge number of other editors to choose from. Some common editors:
  - VSCode
  - Atom
  - Pydev
  - Pycharm
  - Sublime Text
  - IntelliJ (used by some other CS 115s)
- You can use whichever you prefer
  - There are a couple common things you need to know to use fancier editors

Using advanced editors

- Some editors come with the ability to run Python by default
  - With many you need to install a plugin
    - Search for "python" to find one

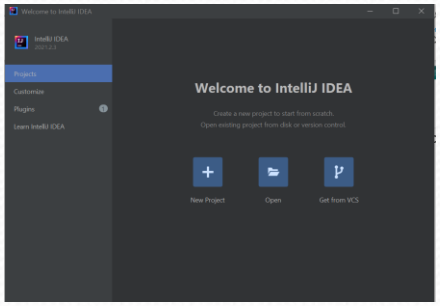
VSCode



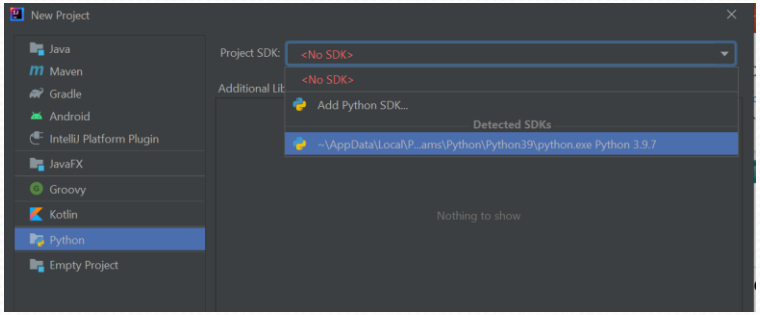
IntelliJ

Using advanced editors

- You need to create a project to be able to run your code in many advanced editors
  - Select "New Project"



  - Select "Python", click the arrow next to Project SDK. You should see an option at the bottom under "Detected SDKs". Select this.
  - Select any name for your project



Using advanced editors

- Once you have a project you will need to create a file to write your code in.
  - Right click on the folder with the same name as your project, select "New" and then select "File".
  - Run your code by pressing the green arrow/triangle in the upper right