Programming, Problem Solving, and Algorithms

CPSC203, 2022 W2

Announcements

- **Test 2 is this week**
- **Lab3 is this week**
  - It's about Dataclasses!
- **Problem of the Week 3 is due this week (extended from last week)**
  - Assigned Pandas videos from Thursday should help!
- **Problem of the Week 4 is also due this week**
  - Dataclasses practice

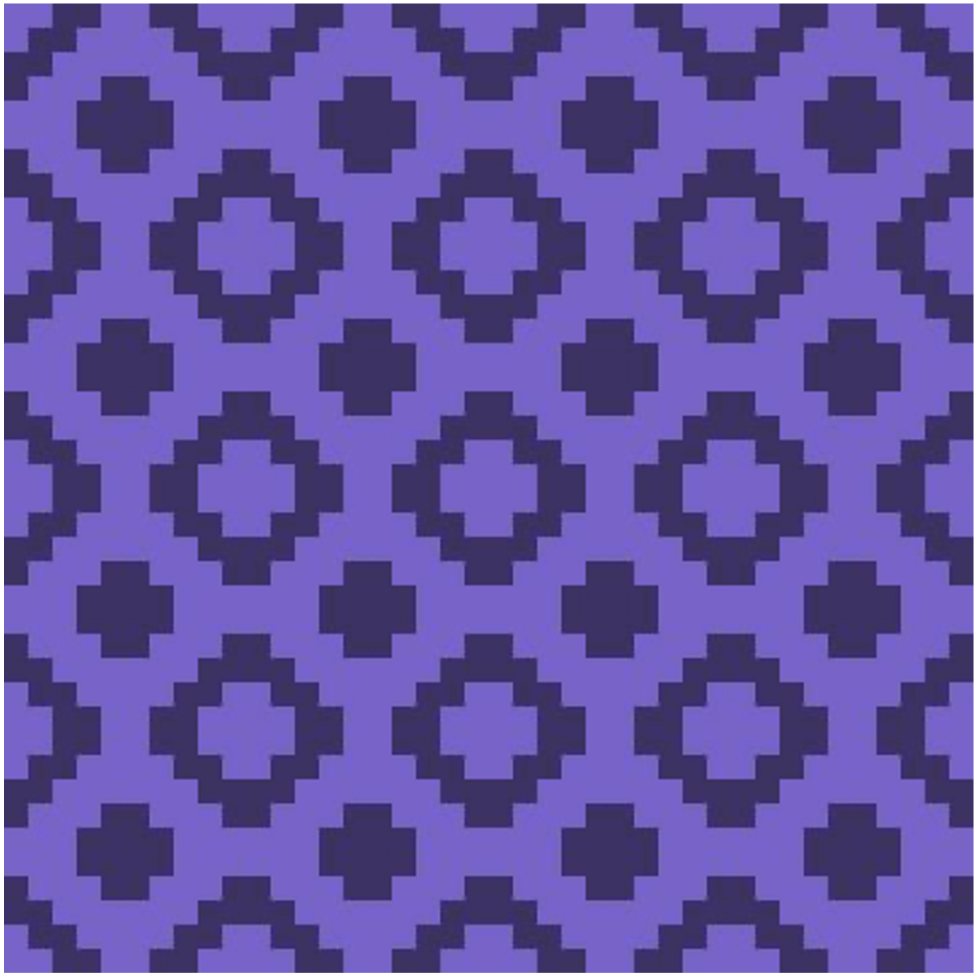
Today's Plan...

1. Announcements! (5 mins)
2. Implement member functions of our Stitcher (45 mins)
3. Break (5 mins)
4. Installing packages using Conda (10 mins)
5. Demo of Pandas (10 mins)

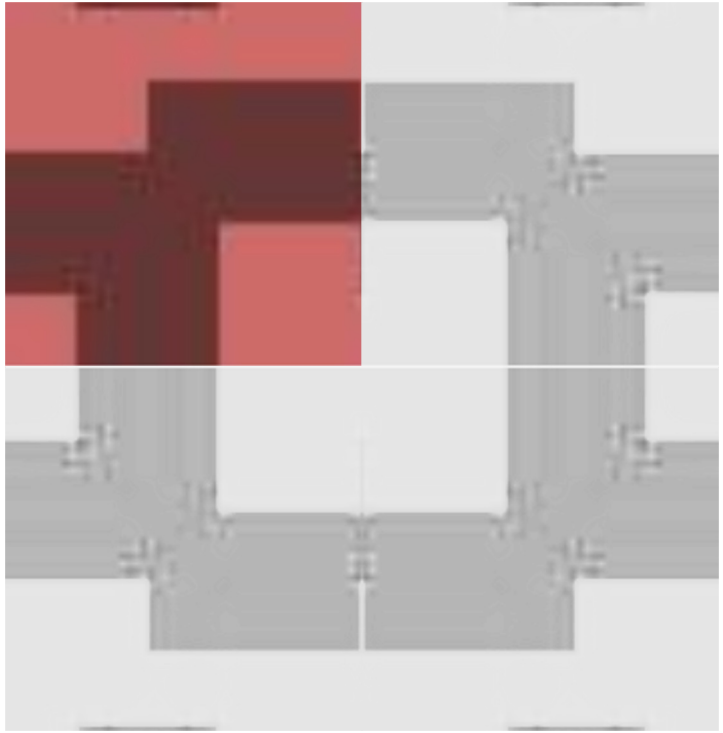
Slides from the Assigned Videos

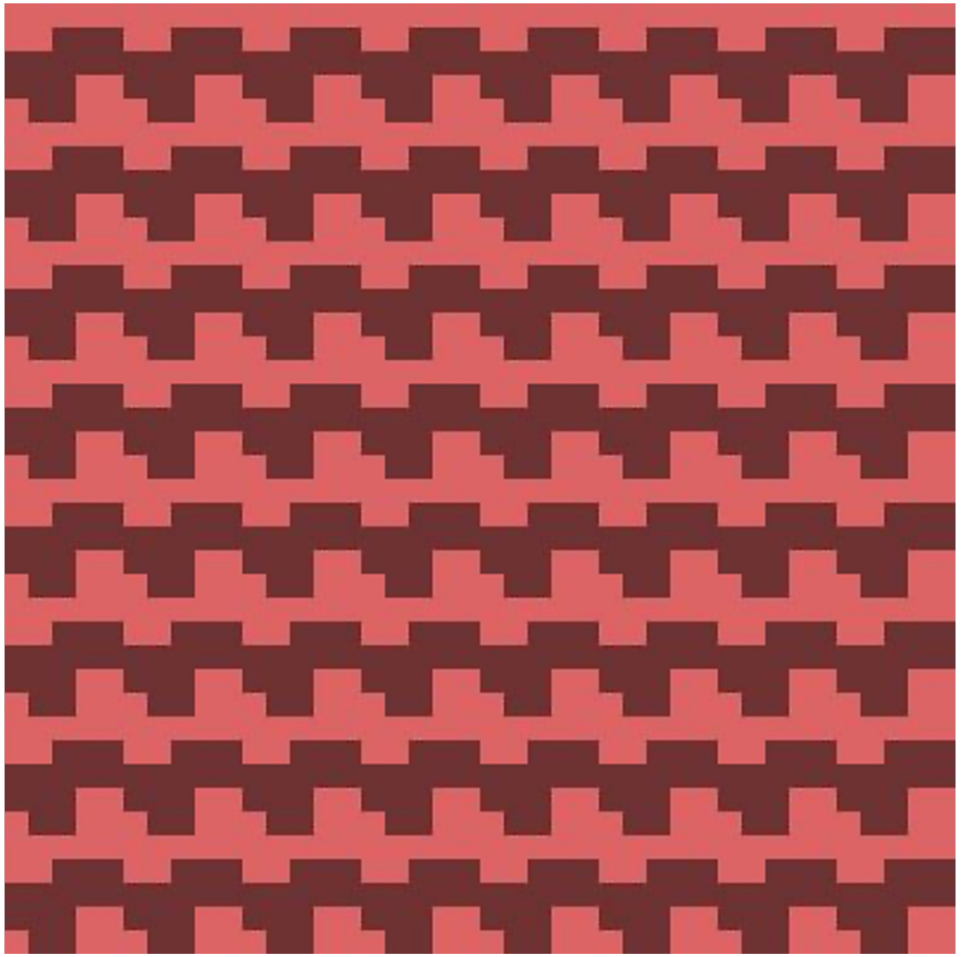
Adding Functionality

Creating blocks is an arduous task. We'd like a way to make new blocks out of old ones! How many different kinds of blocks are found in this image? How are they related to one another?



Example block...



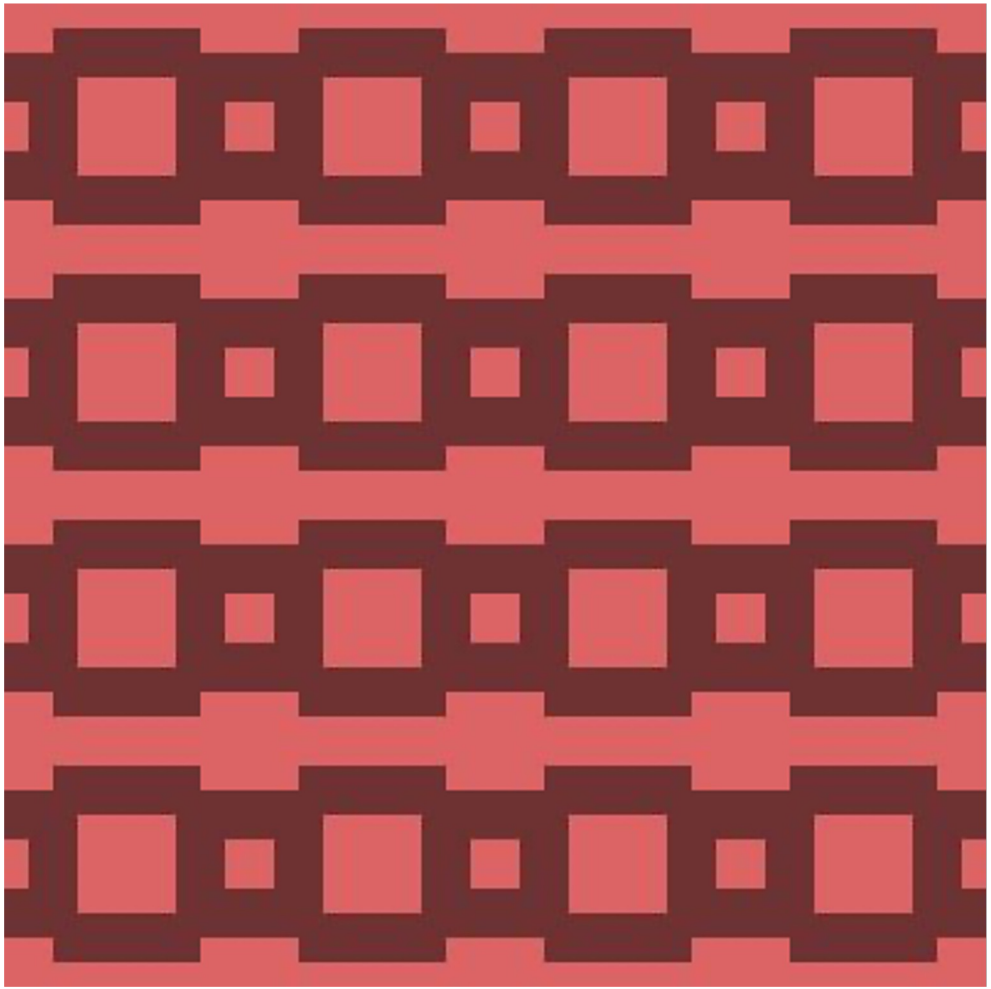


A Block Pattern

We can use that one block to create a surprising pattern!

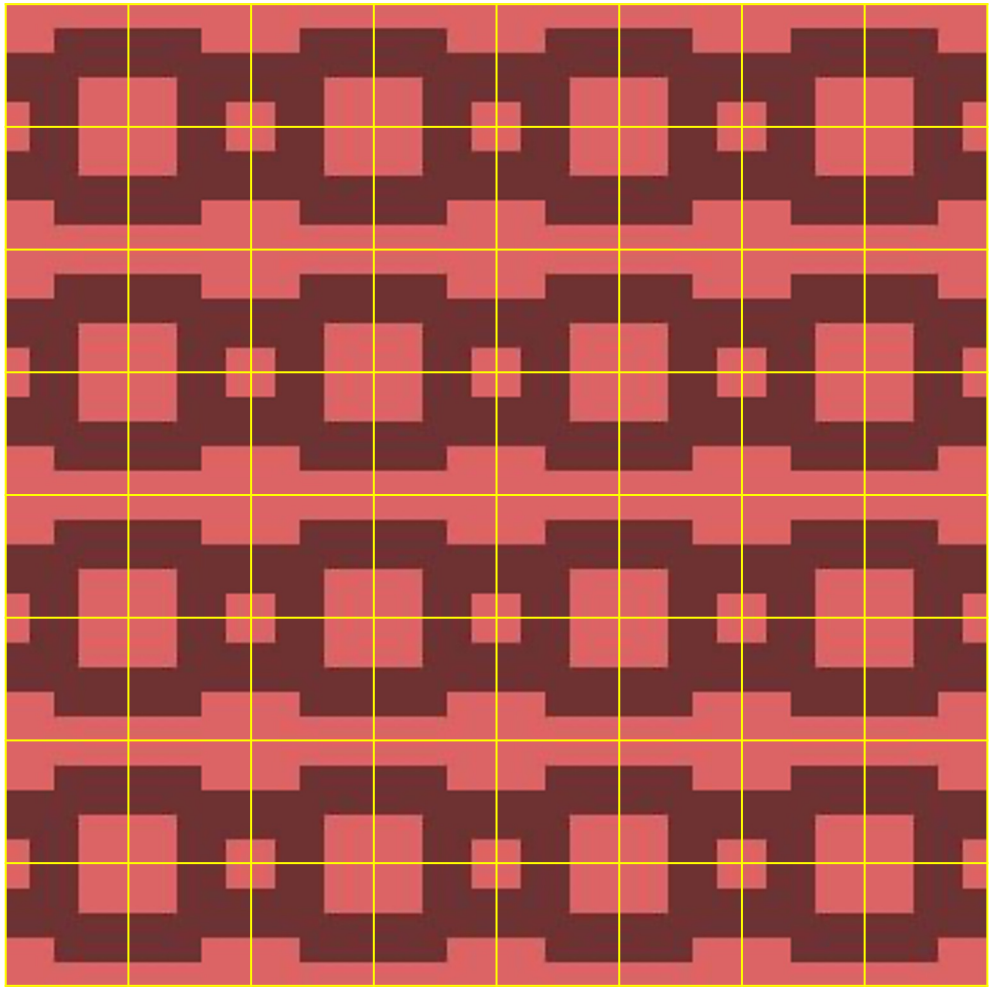
TODOS:

1. Write flippy member functions
2. Use them to create patterns



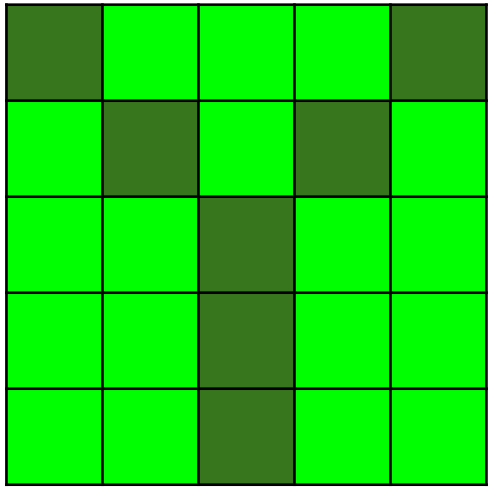
A Block Pattern

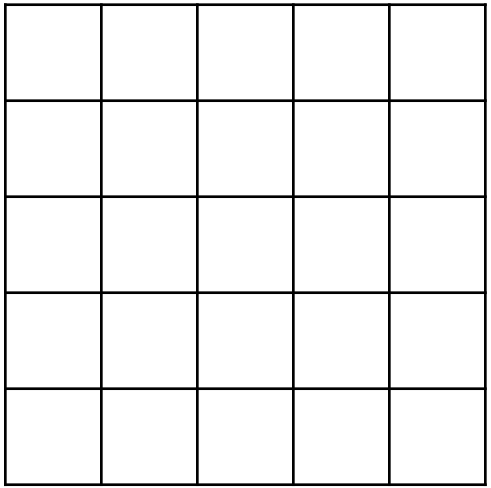
Positions: Row Column



Flip Vertical

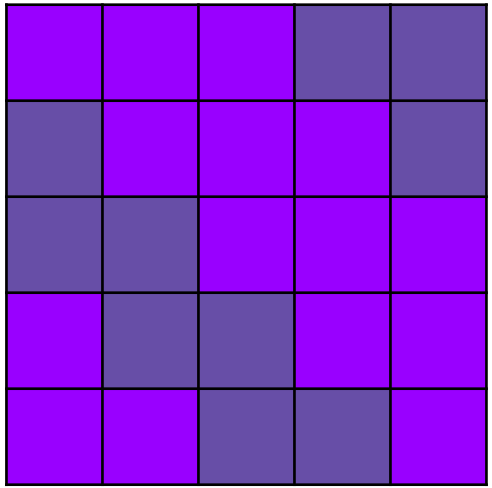
Suppose we want to perform a vertical reflection of a block. Sketch the resulting block. Describe how you would accomplish the flipped block, in terms of the block representation in our code (list of rows).

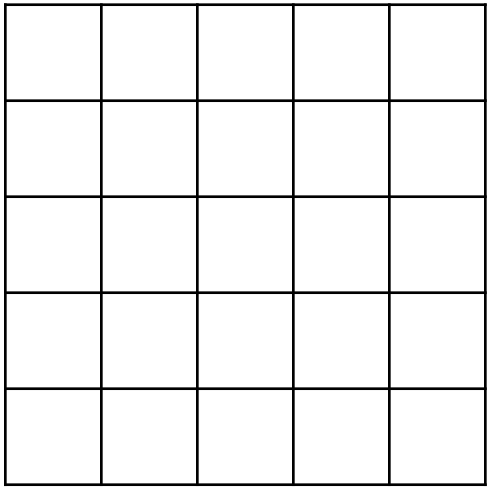




Flip Horizontal

Suppose we want to create a new block which is just the horizontal reflection of a given block. Sketch the new block. Describe how you would accomplish the flipped block, in terms of the block representation in our code.





-

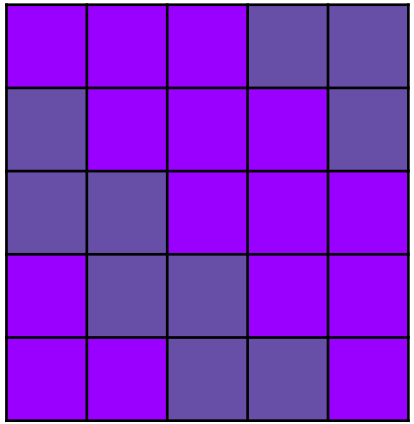
Flip Horizontal

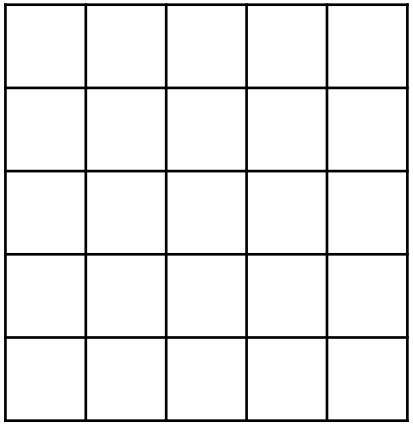
Review the code we've written and make 3 observations:

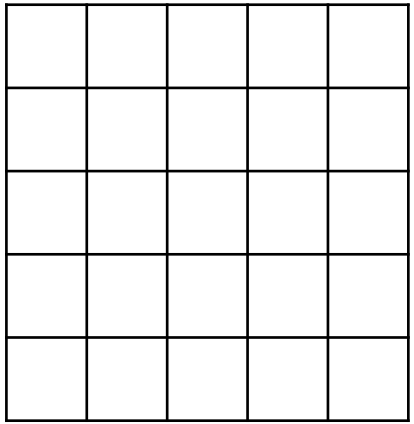
1. Line ___: _____
2. Line ___: _____
3. Line ___: _____

Rotate 180

Suppose we want to create a new block which is a 180 degree rotation of a given block. Sketch the new block. Describe how you would accomplish the flipped block.







1.

2.

Rotate 180

Review the code we've written and make 3 observations:

1. Line ___: _____
2. Line ___: _____
3. Line ___: _____

Implement member functions of Stitcher

Break

Pandas and data frames

```
import pandas
```

Imports the pandas library. We will almost always use an abbreviation...

Instead of saying `pandas.read_csv('file.csv')`

we can say

This function returns a DataFrame containing the data from `file.csv`

CSV files

To implement `df = pd.read_csv('file.csv')`

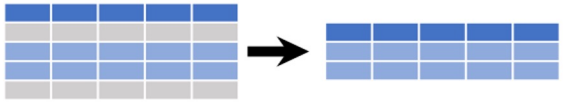
`file.csv` must have field names in row 1, and data beginning in row 2.

bill_week.csv saved

```
1  ,week,title,artist,rank,last_week,peak_pos,weeks_on_chart
2  0,2019-09-21,Truth Hurts,Lizzo,1,1,1,19
3  1,2019-09-21,Senorita,Shawn Mendes & Camila Cabello,2,2,1,12
4  2,2019-09-21,Goodbyes,Post Malone Featuring Young Thug,3,10,3,10
5  3,2019-09-21,Circles,Post Malone,4,7,4,2
6  4,2019-09-21,Bad Guy,Billie Eilish,5,3,1,24
7  5,2019-09-21,Ran$om,Lil Tecca,6,4,4,15
8  6,2019-09-21,No Guidance,Chris Brown Featuring Drake,7,6,6,14
```

Selecting Rows

Subset Observations (Rows)



`df[df.Length > 7]`
Extract rows that meet logical criteria.

`df.drop_duplicates()`
Remove duplicate rows (only considers columns).

`df.head(n)`
Select first n rows.

`df.tail(n)`
Select last n rows.

`df.sample(frac=0.5)`
Randomly select fraction of rows.

`df.sample(n=10)`
Randomly select n rows.

`df.iloc[10:20]`
Select rows by position.

`df.nlargest(n, 'value')`
Select and order top n entries.

`df.nsmallest(n, 'value')`
Select and order bottom n entries.

Logic in Python (and pandas)			
<	Less than	!=	Not equal to
>	Greater than	`df.column.isin(values)`	Group membership
==	Equals	`pd.isnull(obj)`	Is NaN
<=	Less than or equals	`pd.notnull(obj)`	Is not NaN
>=	Greater than or equals	`&,|,~,^,df.any(),df.all()`	Logical and, or, not, xor, any, all

```
df.nlargest(10,'last_week')
```

Returns top 10 hits from last week.

```
df[ df['weeks_on_chart'] > 10 ]
```

Returns all songs that have been on the charts for more than 10 weeks.

Adding a column

```
df['gradient'] = df['last_week'] - df['rank']
```

Adds a column to the DataFrame containing the difference for every row.

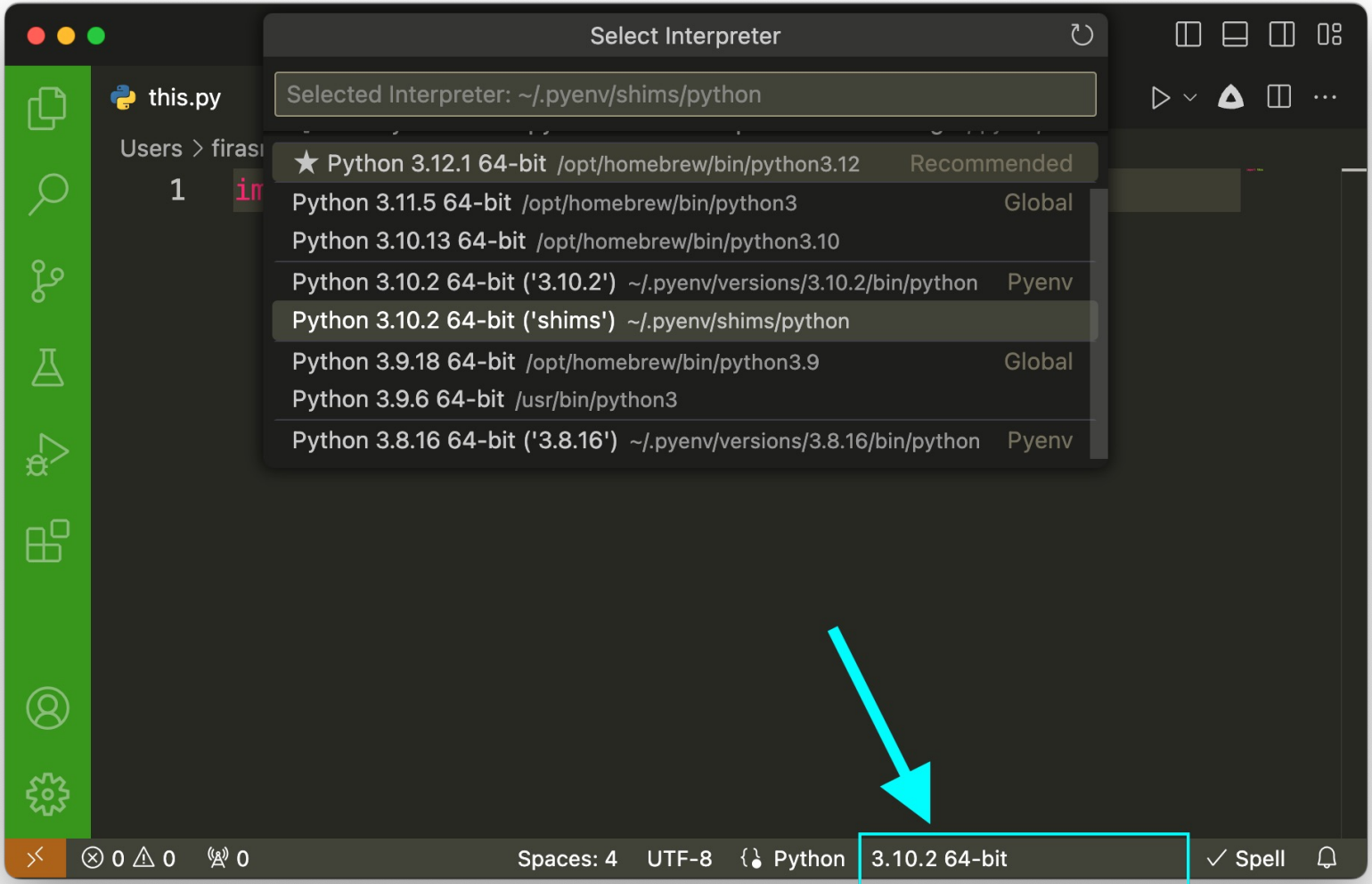
```
df[ df['gradient'] > 10 ]
```

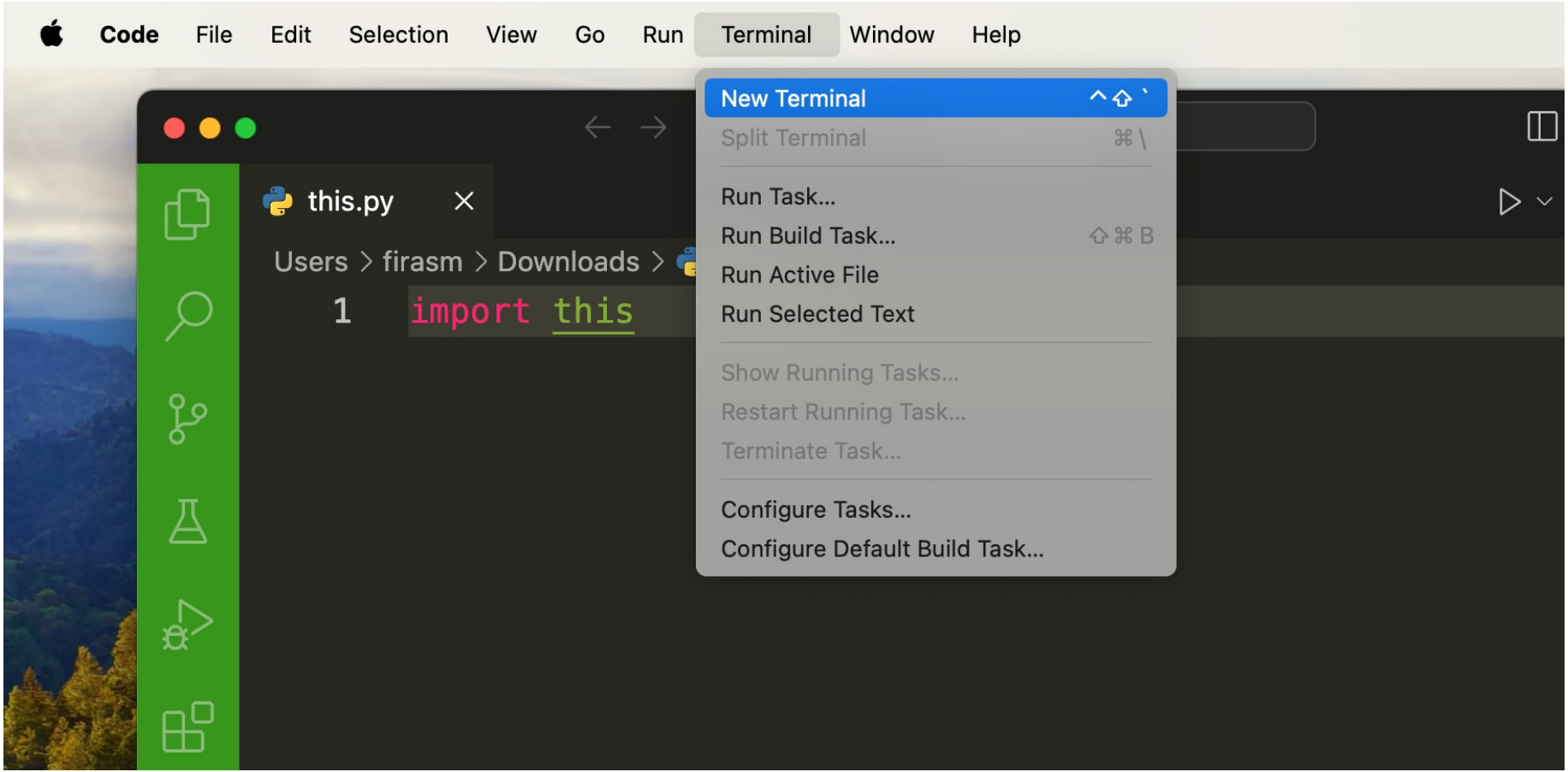
Returns all songs that have moved more than 10 spaces in the last week..

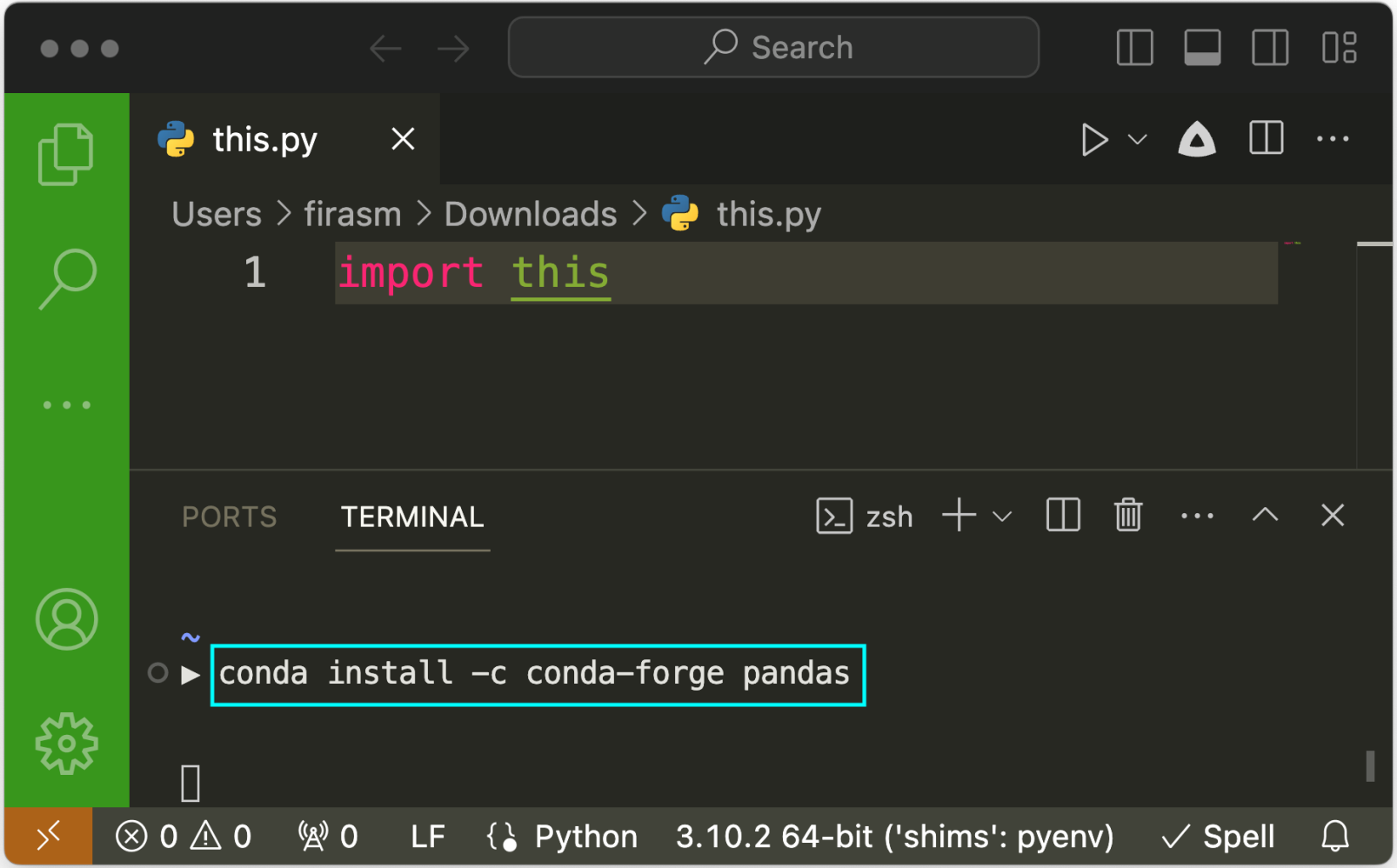
Installing packages using conda

CONDA CHEAT SHEET

Command line package and environment manager


Select Interpreter
Selected Interpreter: ~/.pyenv/shims/python
★ Python 3.12.1 64-bit /opt/homebrew/bin/python3.12 Recommended
Python 3.11.5 64-bit /opt/homebrew/bin/python3 Global
Python 3.10.13 64-bit /opt/homebrew/bin/python3.10
Python 3.10.2 64-bit ('3.10.2') ~/.pyenv/versions/3.10.2/bin/python Pyenv
Python 3.10.2 64-bit ('shims') ~/.pyenv/shims/python
Python 3.9.18 64-bit /opt/homebrew/bin/python3.9 Global
Python 3.9.6 64-bit /usr/bin/python3
Python 3.8.16 64-bit ('3.8.16') ~/.pyenv/versions/3.8.16/bin/python Pyenv
this.py
Users > firas
1
Spaces: 4
UTF-8
Python
3.10.2 64-bit
Spell


Code File Edit Selection View Go Run Terminal Window Help
New Terminal
Split Terminal
Run Task...
Run Build Task...
Run Active File
Run Selected Text
Show Running Tasks...
Restart Running Task...
Terminate Task...
Configure Tasks...
Configure Default Build Task...
this.py
Users > firasm > Downloads >
1 import this


Search
this.py
Users > firasm > Downloads > this.py
1 import this
PORTS
TERMINAL
zsh
~
conda install -c conda-forge pandas
0 0
0
LF
Python
3.10.2 64-bit ('shims': pyenv)
Spell

Demo of Pandas