Improving Your Python Programming Experience

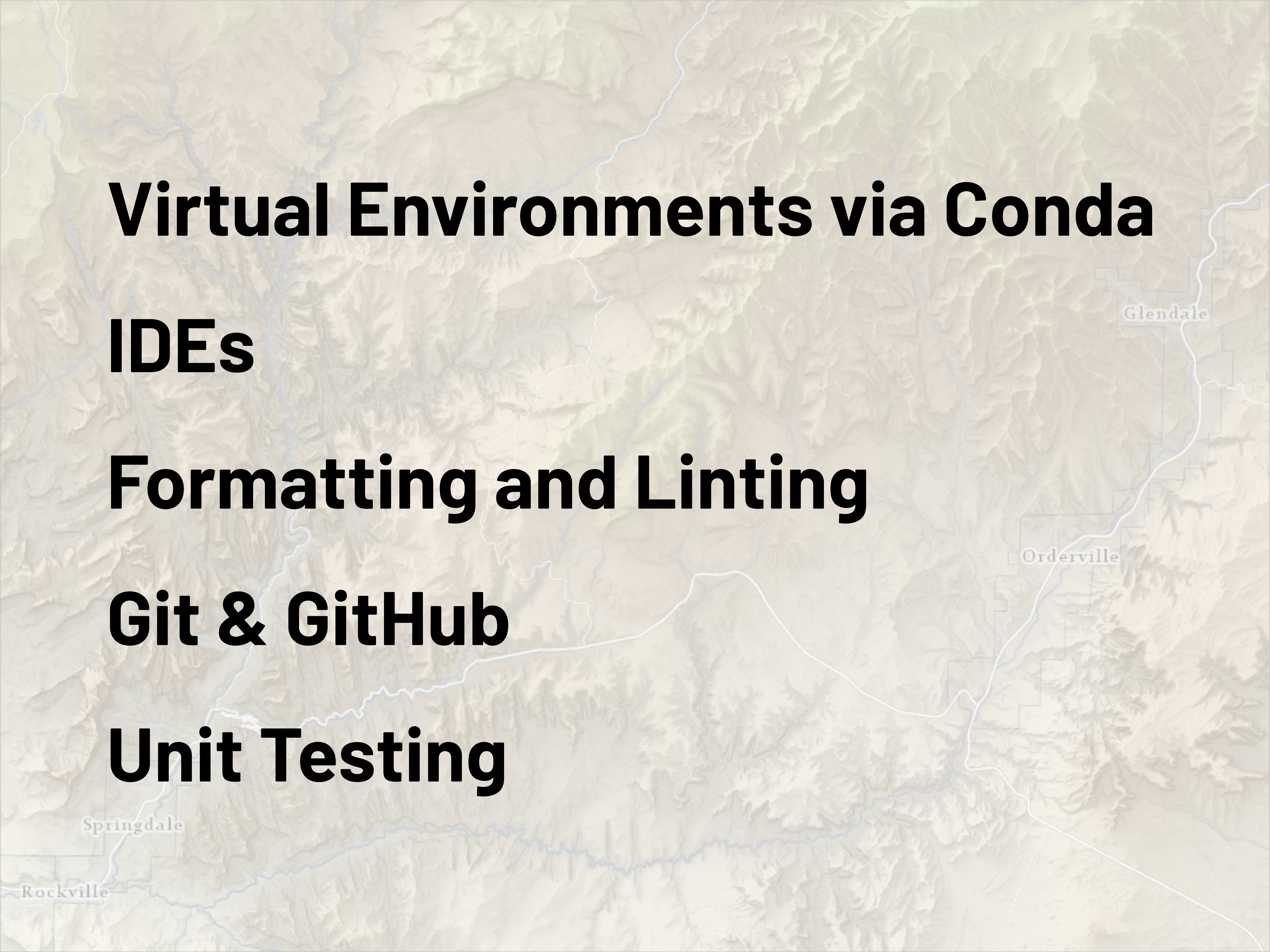
Tools and Tricks to Make Your Life Easier

Jacob Adams, UGRC

UGIC 2022

RoofBag.com

Rockville



Virtual Environments via Conda

IDEs

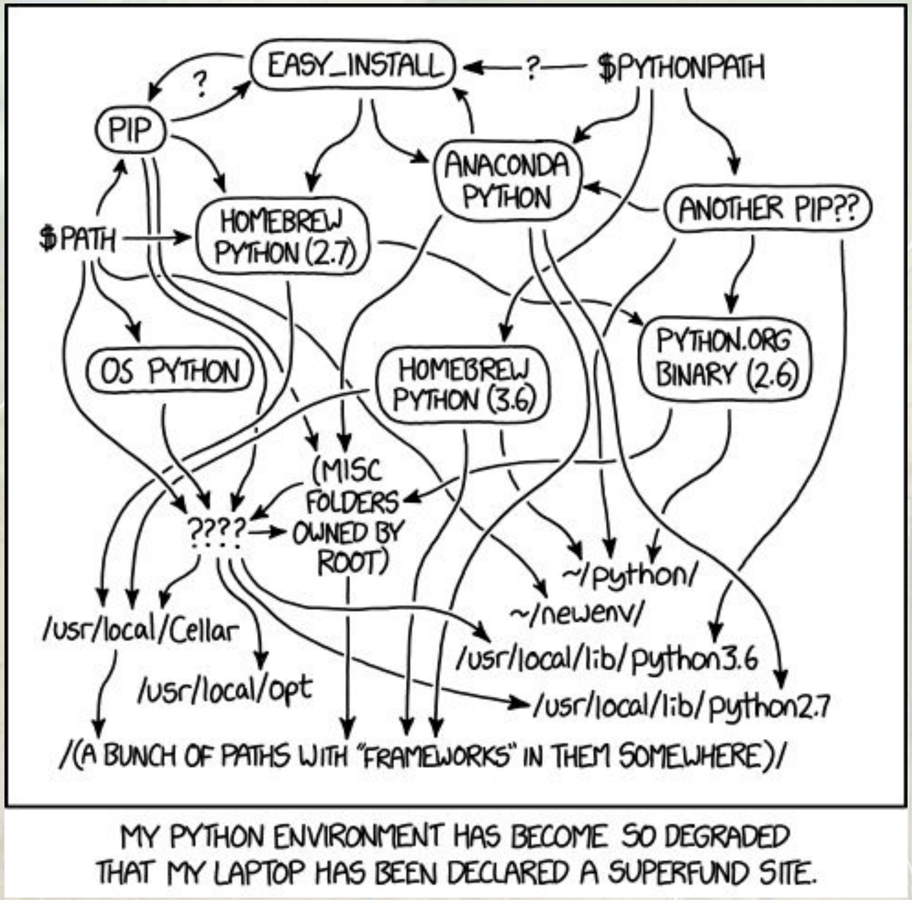
Formatting and Linting

Git & GitHub

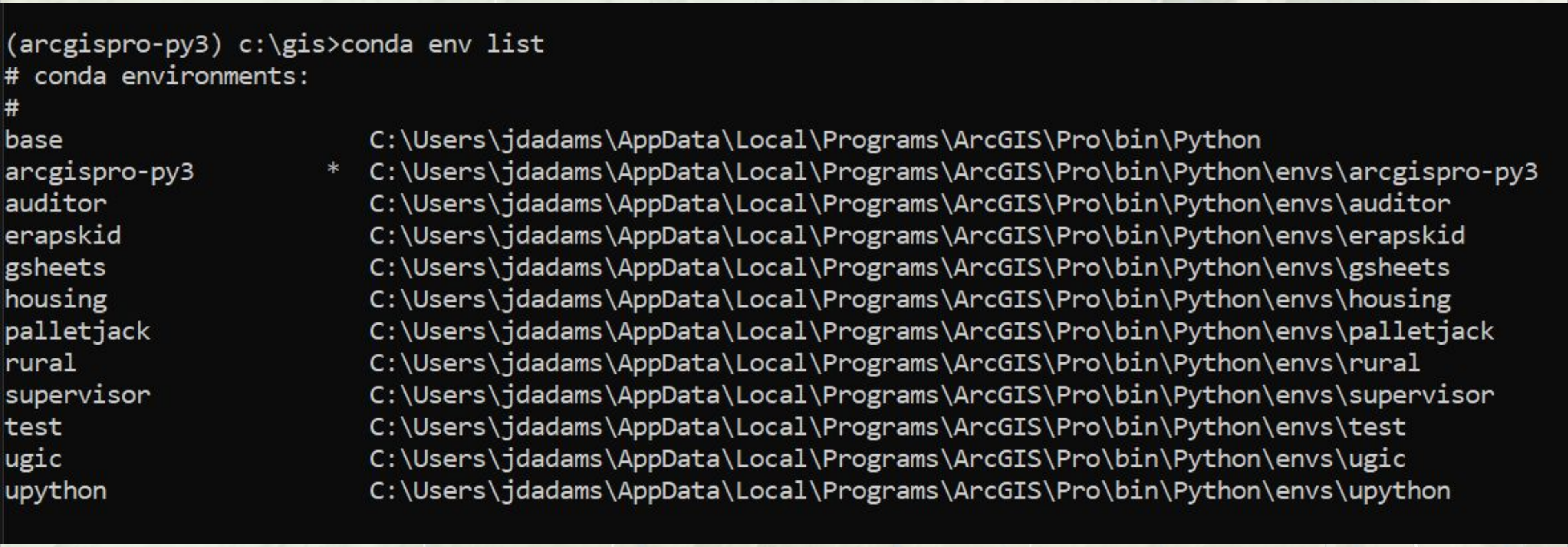
Unit Testing

DON'T
PANIC

Managing Your Environments With CONDA



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

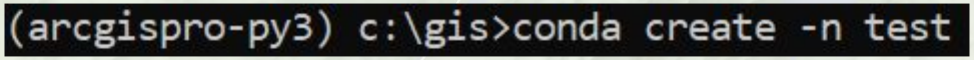


```
(arcgispro-py3) c:\gis>conda env list
# conda environments:
#
base                     C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python
arcgispro-py3         *  C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\arcgispro-py3
auditor                  C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\auditor
erapskid                 C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\erapskid
gsheets                  C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\gsheets
housing                  C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\housing
palletjack               C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\palletjack
rural                    C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\rural
supervisor               C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\supervisor
test                     C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\test
ugic                     C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\ugic
upython                  C:\Users\jdadams\AppData\Local\Programs\ArcGIS\Pro\bin\Python\envs\upython
```

Conda virtual environments: separate sandboxes for your projects

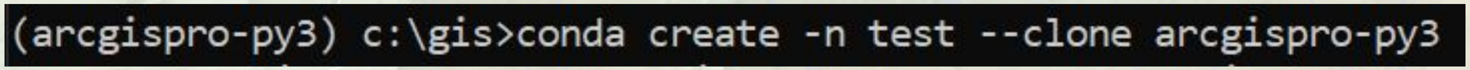
Lowes.com

Create a new, blank environment:



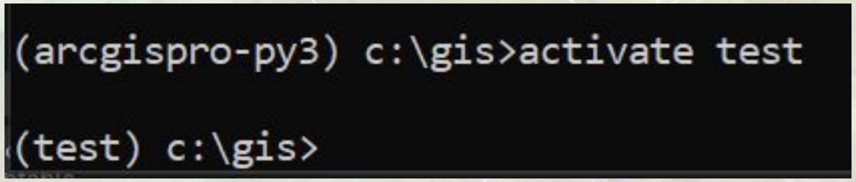
```
(arcgispro-py3) c:\gis>conda create -n test
```

Clone an existing environment:



```
(arcgispro-py3) c:\gis>conda create -n test --clone arcgispro-py3
```

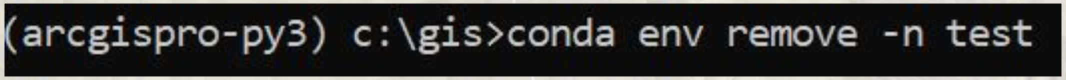
Activate an environment:



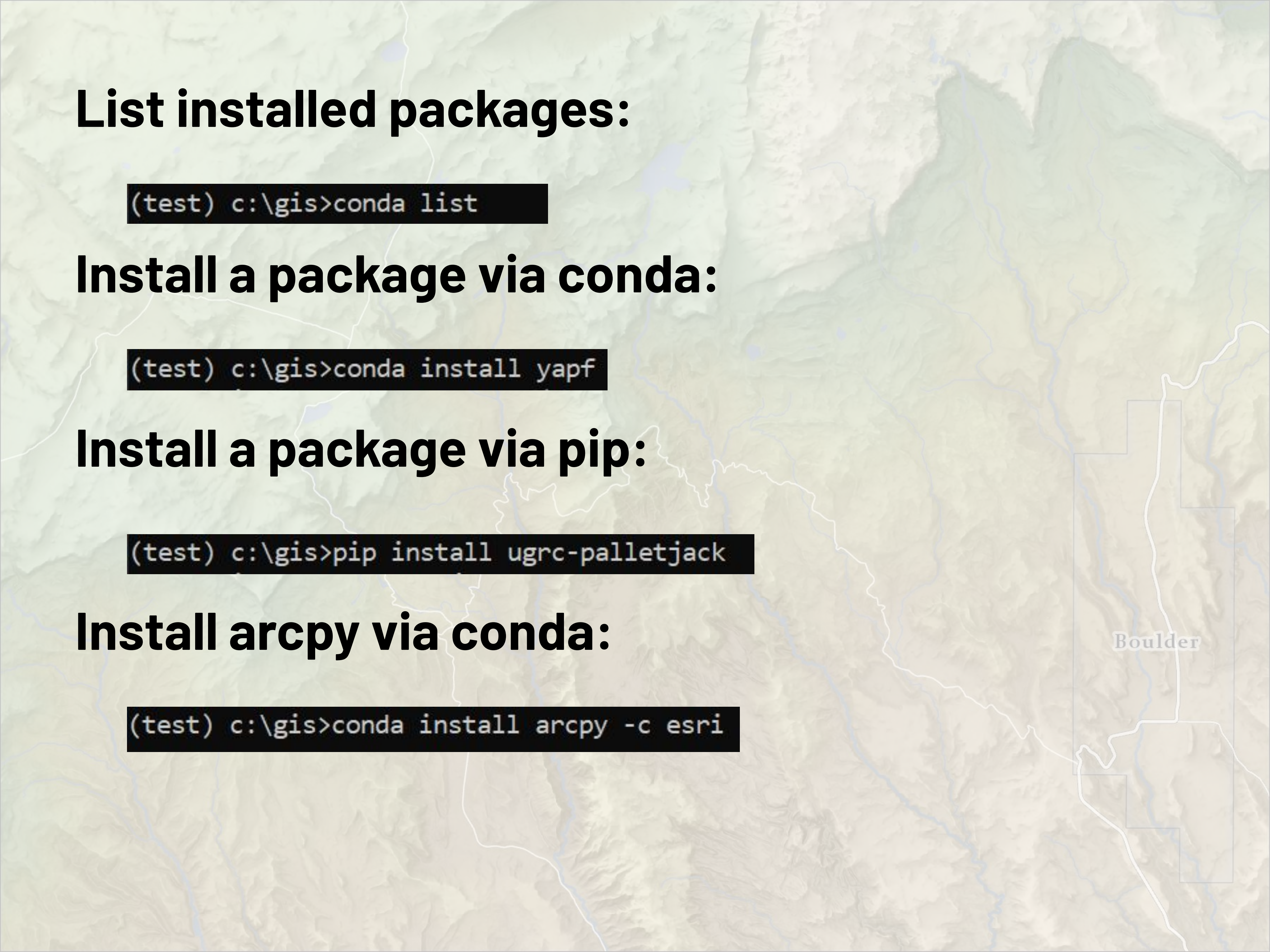
```
(arcgispro-py3) c:\gis>activate test

(test) c:\gis>
```

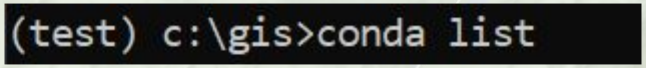
Delete an environment:



```
(arcgispro-py3) c:\gis>conda env remove -n test
```

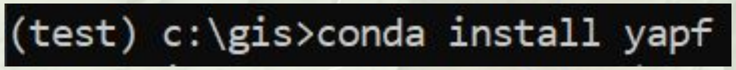


List installed packages:



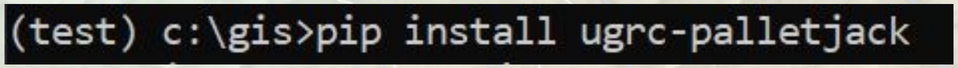
```
(test) c:\gis>conda list
```

Install a package via conda:



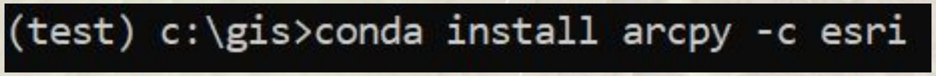
```
(test) c:\gis>conda install yapf
```

Install a package via pip:



```
(test) c:\gis>pip install ugrc-palletjack
```

Install arcpy via conda:



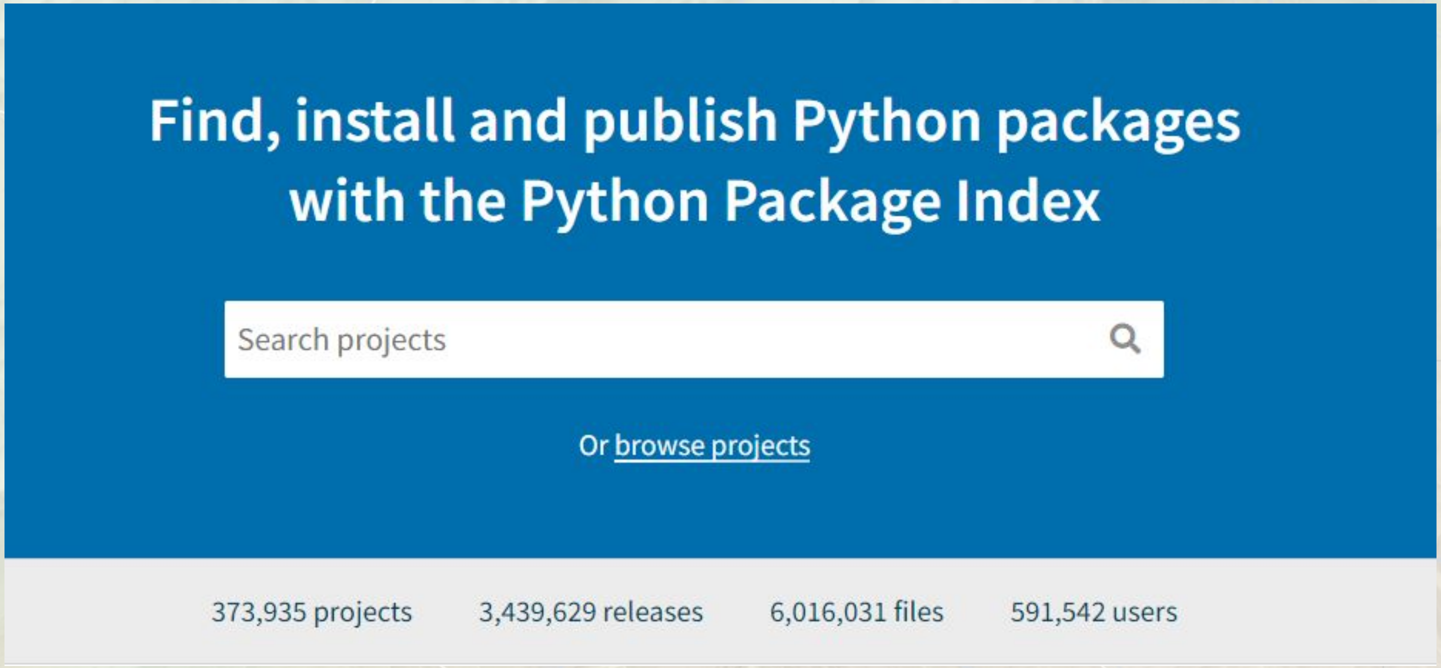
```
(test) c:\gis>conda install arcpy -c esri
```

CONDA vs python™ Package Index

- Installs Python
- Many channels
- Manages environments
- Installed with ArcGIS Pro

- Requires Python
- More packages
- Easy to automate dependencies
- Easily install local packages

Packaging Your Code


Find, install and publish Python packages with the Python Package Index

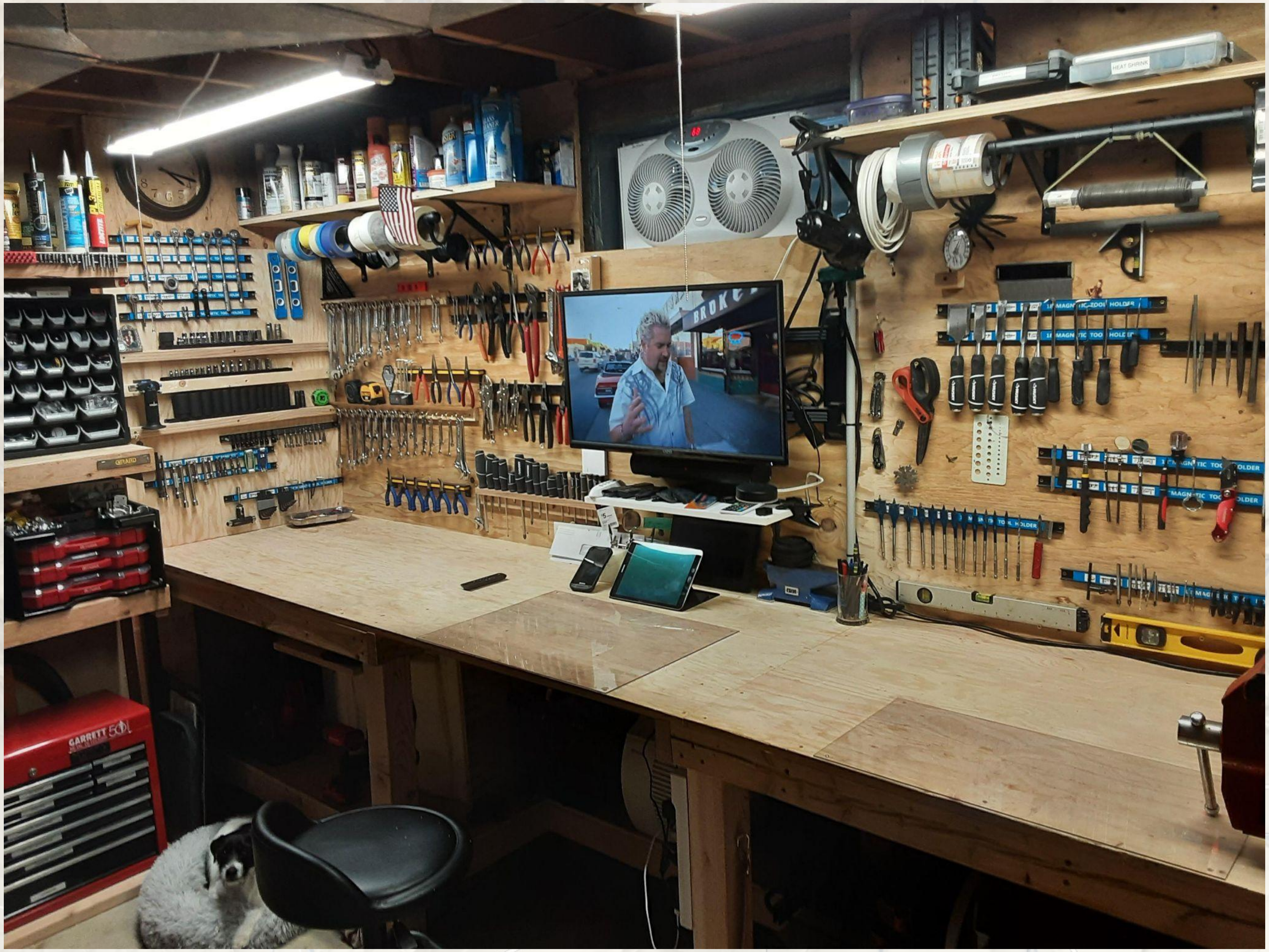
373,935 projects 3,439,629 releases 6,016,031 files 591,542 users

Configuration: `setup.py`

Local Installation: `pip install -e .[tests]`

Distribution: Upload to PyPi

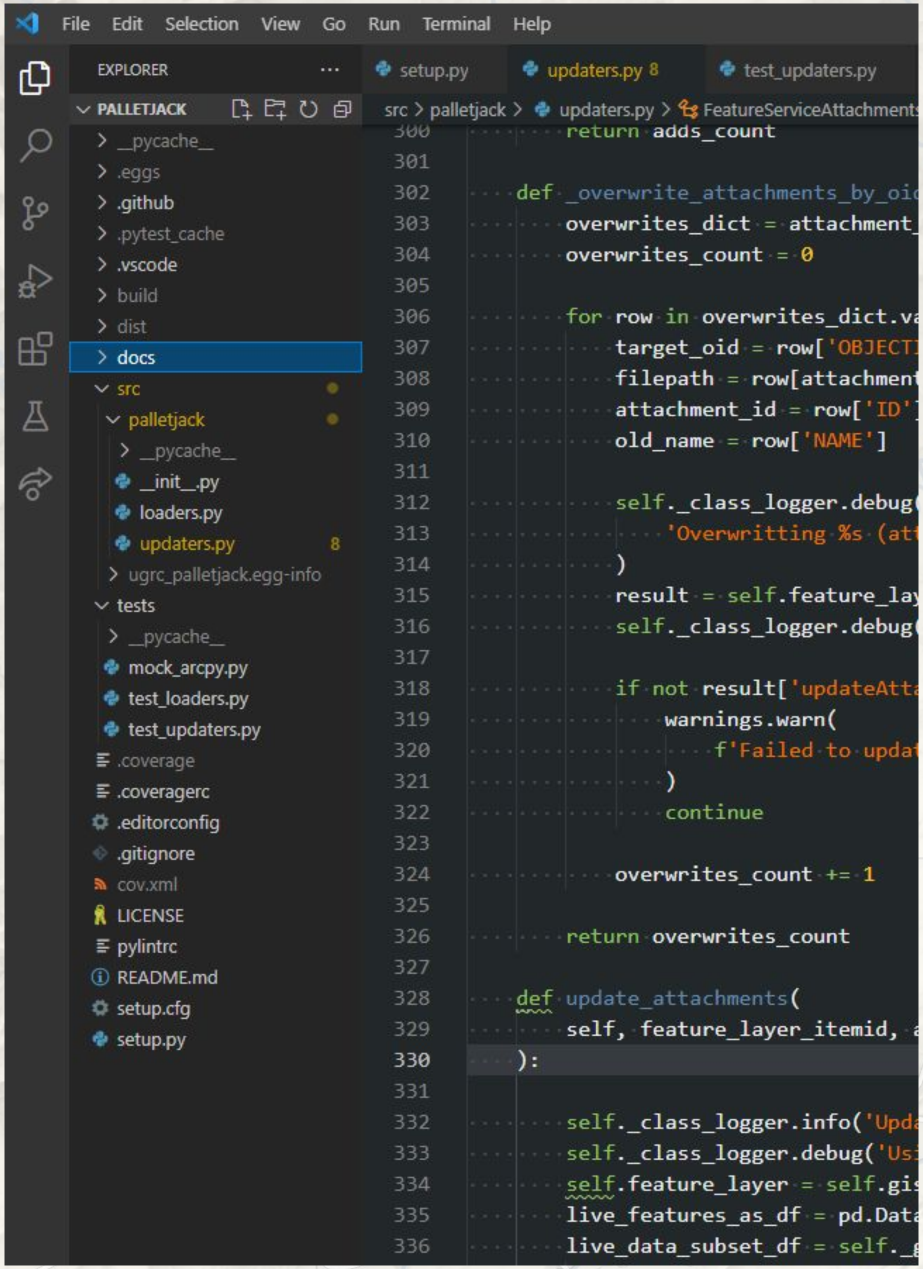
Writing with an IDE



https://www.reddit.com/r/Workbenches/comments/lsl4v0/small_but_gets_it_done/

Visual Studio Code:

- Code hinting
- Highlighting
- Multi-select
- Go to definition
- Extensions
- Renaming
- Debugging



```python
        return adds_count

    def _overwrite_attachments_by_oid
        overwrites_dict = attachment_
        overwrites_count = 0

        for row in overwrites_dict.va
            target_oid = row['OBJECTI
            filepath = row[attachment
            attachment_id = row['ID']
            old_name = row['NAME']

            self._class_logger.debug(
                'Overwritting %s (at
            )
            result = self.feature_lay
            self._class_logger.debug(

            if not result['updateAtta
                warnings.warn(
                    f'Failed to updat
                )
                continue

            overwrites_count += 1

        return overwrites_count

    def update_attachments(
        self, feature_layer_itemid, a
    ):

        self._class_logger.info('Upda
        self._class_logger.debug('Usi
        self.feature_layer = self.gis
        live_features_as_df = pd.Data
        live_data_subset_df = self._g
```

Formatting & Linting

Why Code Style Matters

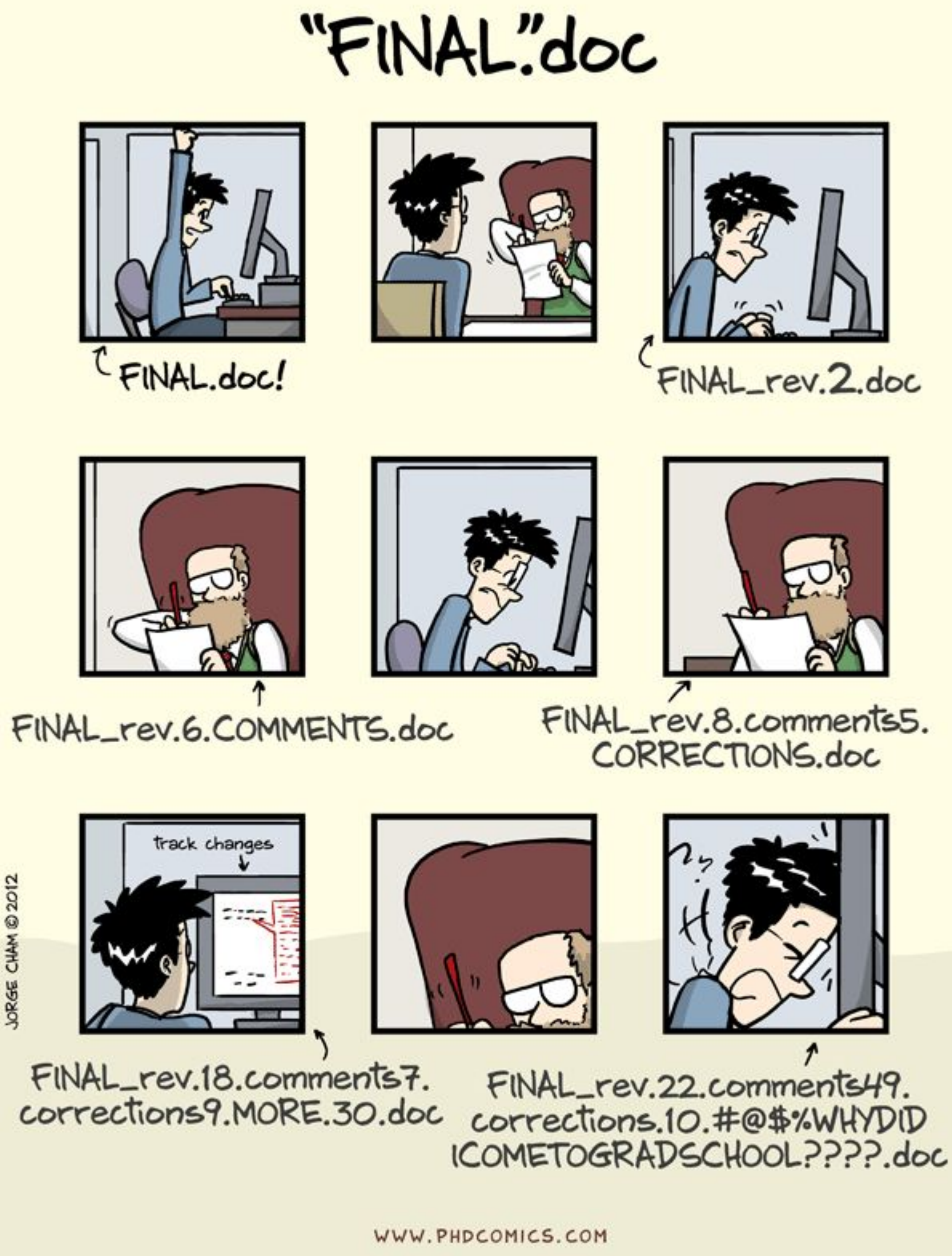
Green River

Formatting vs Linting

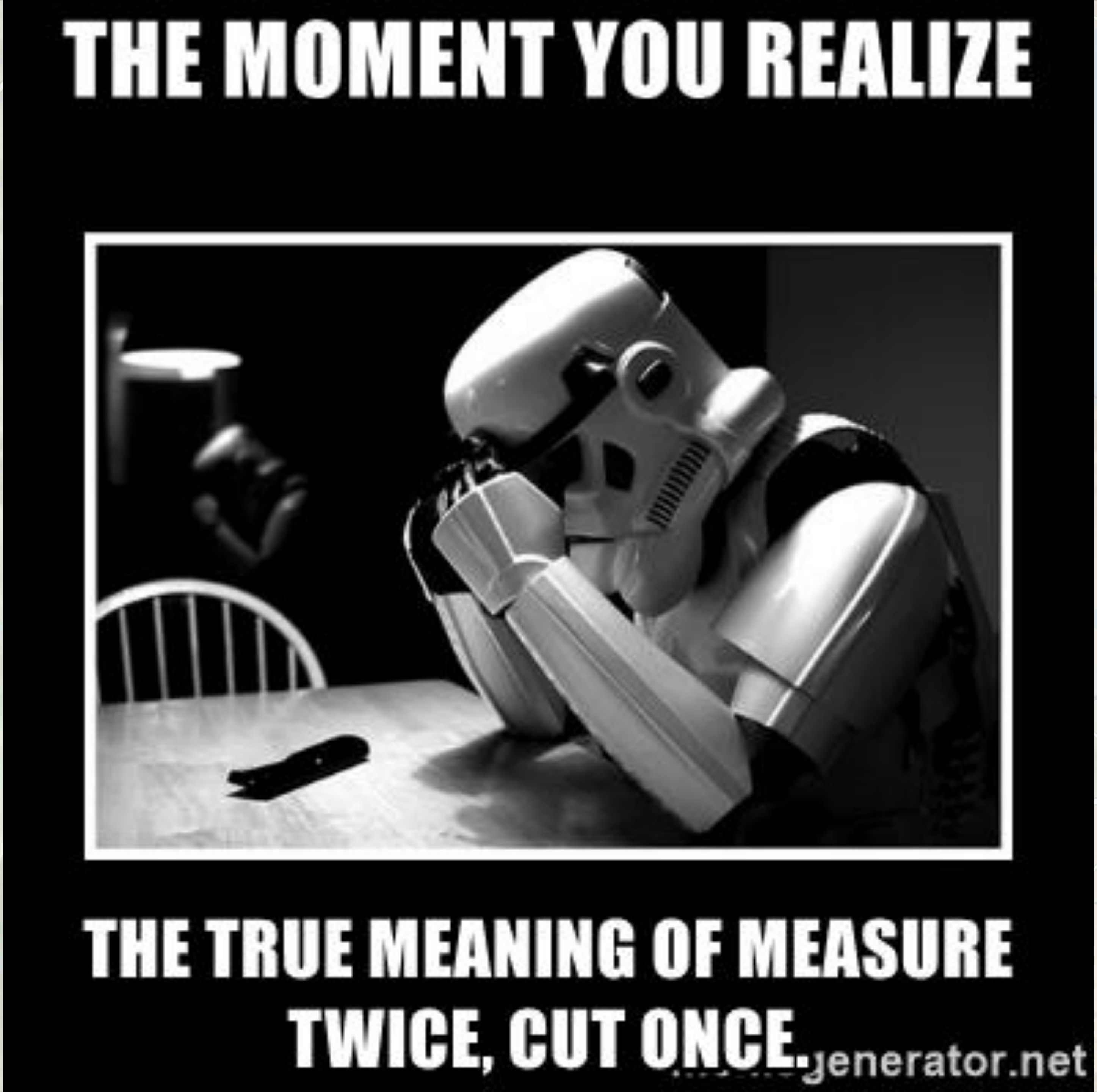
- PEP 8
- Consistent style
- Doc strings and comments
- Performed by `yapf`
- Controlled by config file (`setup.cfg`, `pyproject.toml`, ...)

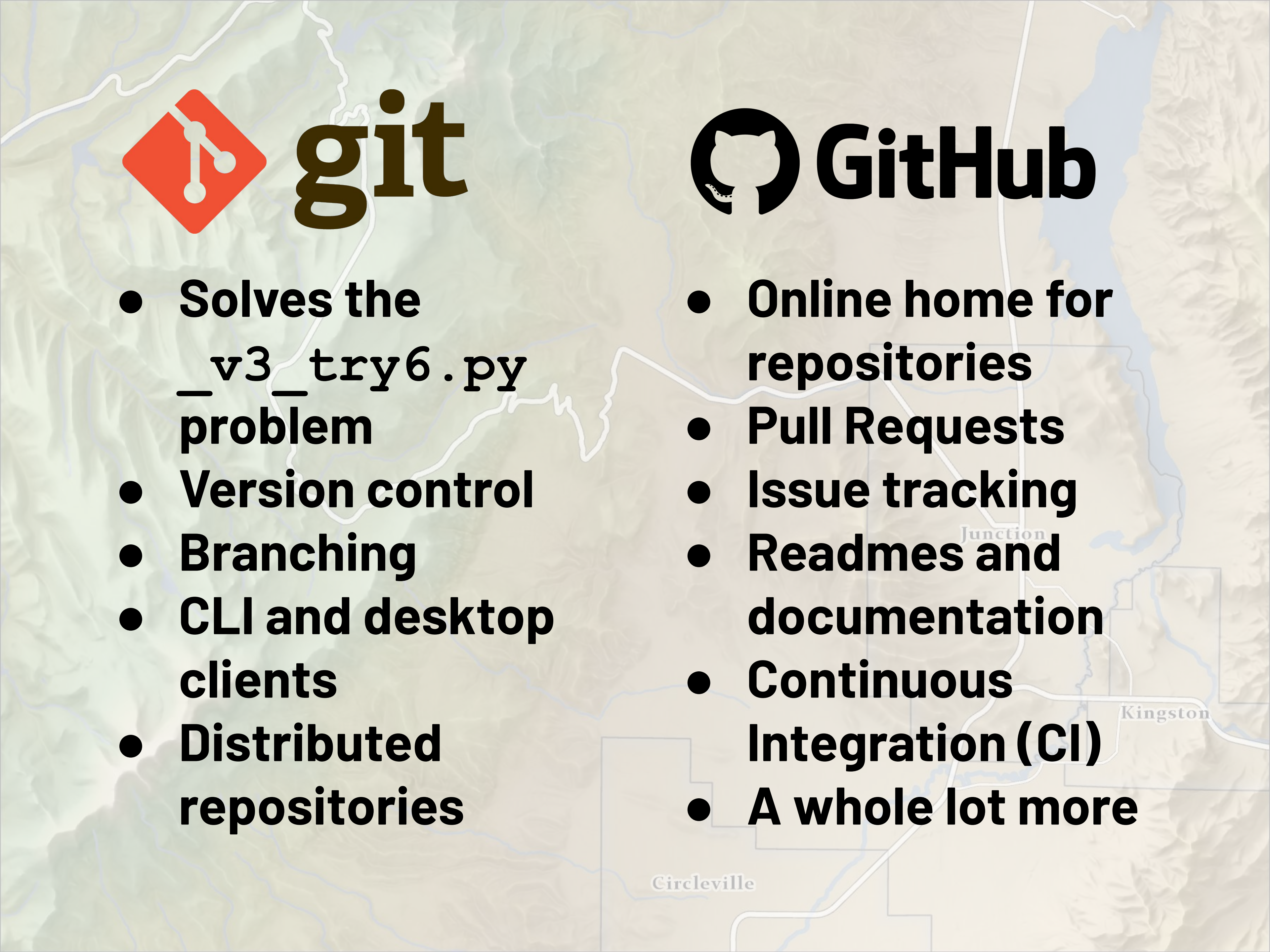
- Detects errors and "code smell"
- Enforces good programming practices
- Performed by `pylint`
- Controlled by `.pylintrc`

Getting a Handle on File Names: Version Control with Git & GitHub



"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5. CORRECTIONS.doc
track changes
FINAL_rev.18.comments7. corrections9.MORE.30.doc
FINAL_rev.22.comments49. corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM


THE MOMENT YOU REALIZE
THE TRUE MEANING OF MEASURE TWICE, CUT ONCE.
generator.net

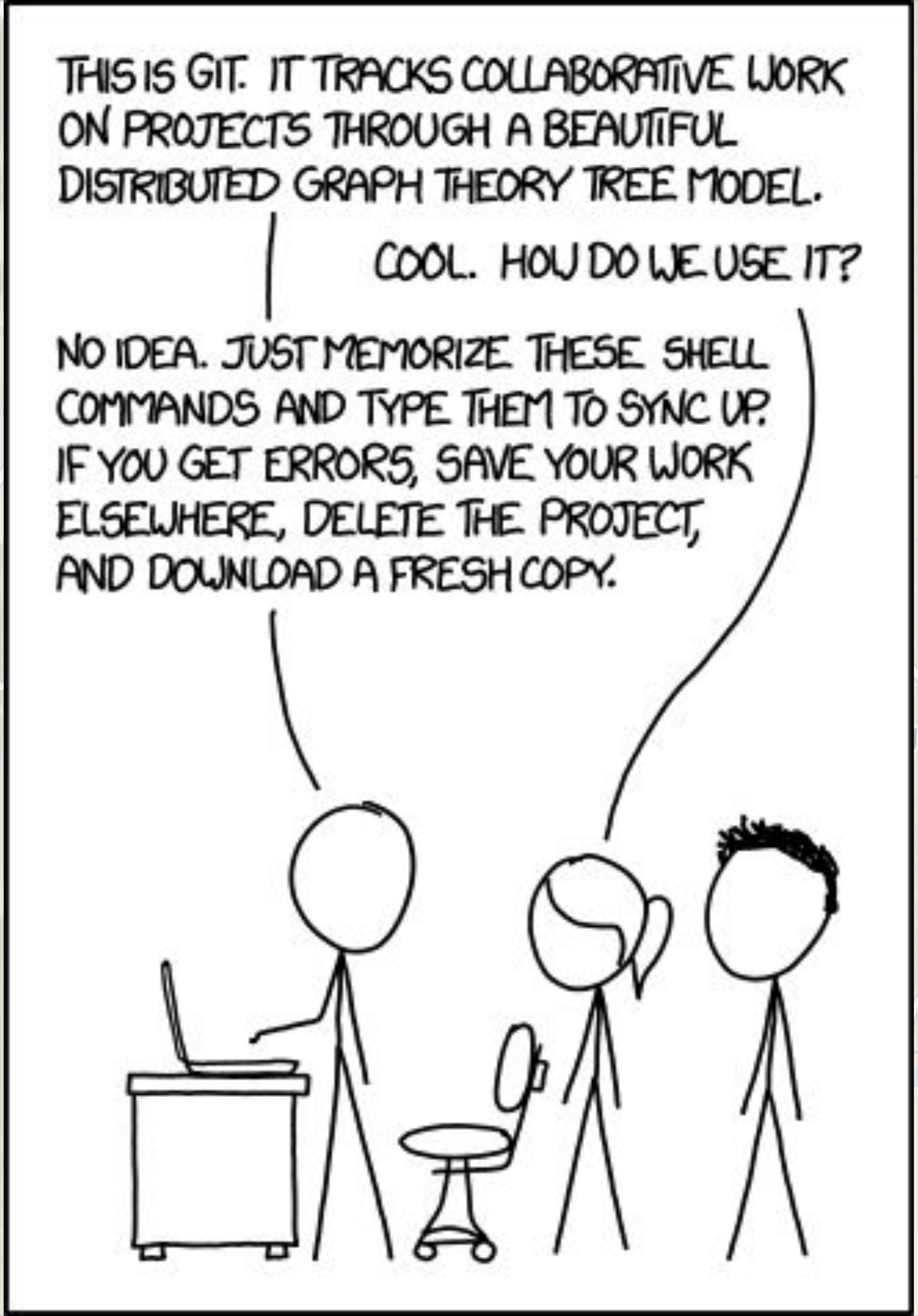


git

- Solves the `_v3_try6.py` problem
- Version control
- Branching
- CLI and desktop clients
- Distributed repositories

GitHub

- Online home for repositories
- Pull Requests
- Issue tracking
- Readmes and documentation
- Continuous Integration (CI)
- A whole lot more


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

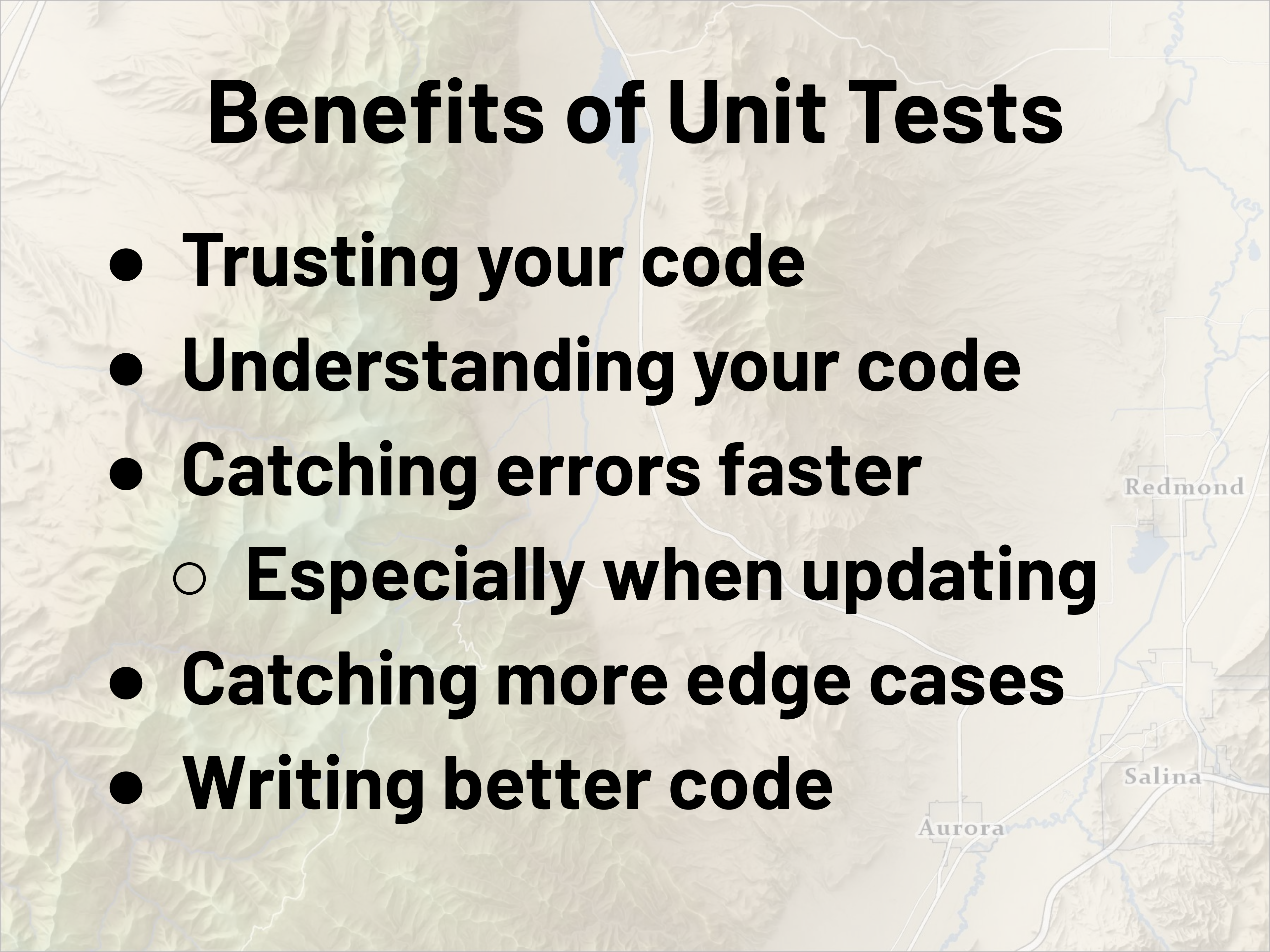
Unit Testing: Knowing When You Break Things

I do one thing at a time. I do it very well, and then I move on.
Charles Emerson Winchester III

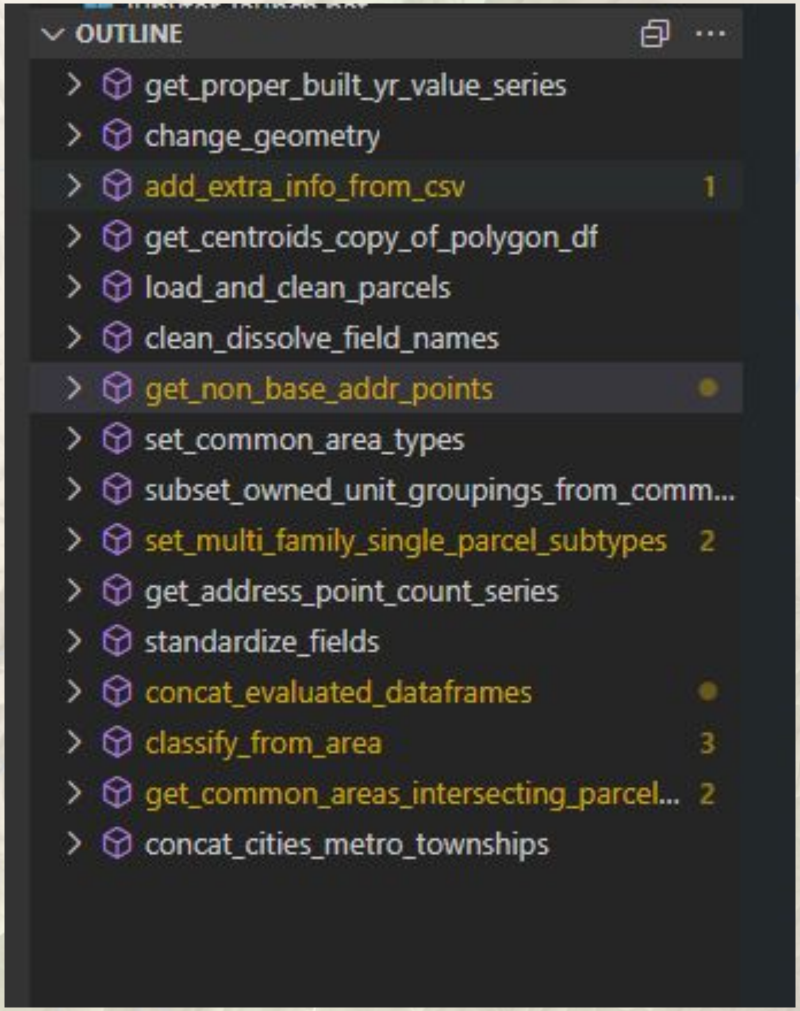
https://reddit.com/u/Patchen35

Benefits of Unit Tests

- Trusting your code
- Understanding your code
- Catching errors faster
  - Especially when updating
- Catching more edge cases
- Writing better code

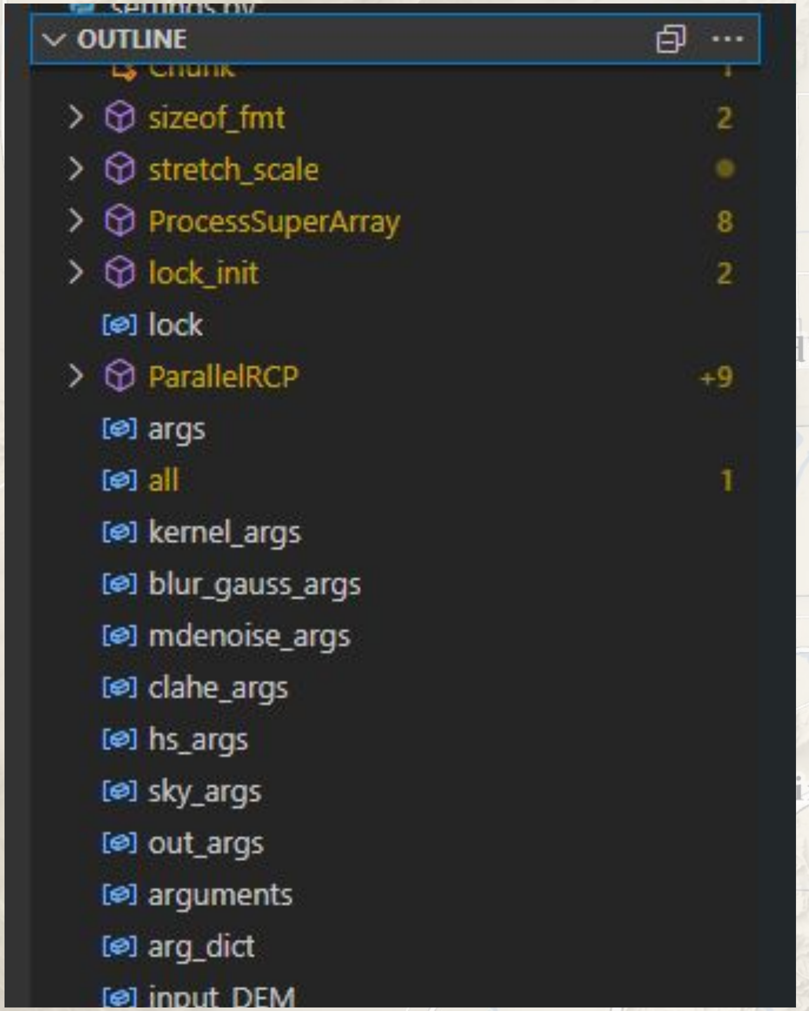


Good Program Design



```
OUTLINE
> get_proper_built_yr_value_series
> change_geometry
> add_extra_info_from_csv                          1
> get_centroids_copy_of_polygon_df
> load_and_clean_parcels
> clean_dissolve_field_names
> get_non_base_addr_points                         ●
> set_common_area_types
> subset_owned_unit_groupings_from_comm...
> set_multi_family_single_parcel_subtypes          2
> get_address_point_count_series
> standardize_fields
> concat_evaluated_dataframes                      ●
> classify_from_area                               3
> get_common_areas_intersecting_parcel...          2
> concat_cities_metro_townships
```

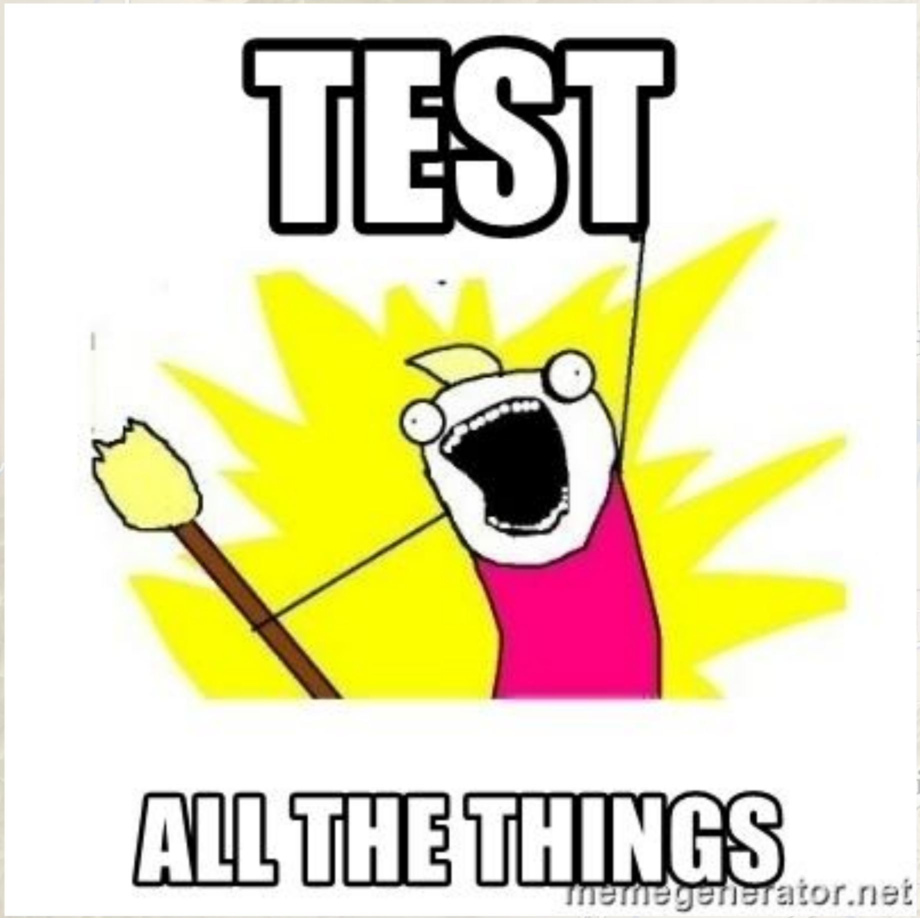
Bad Program Design

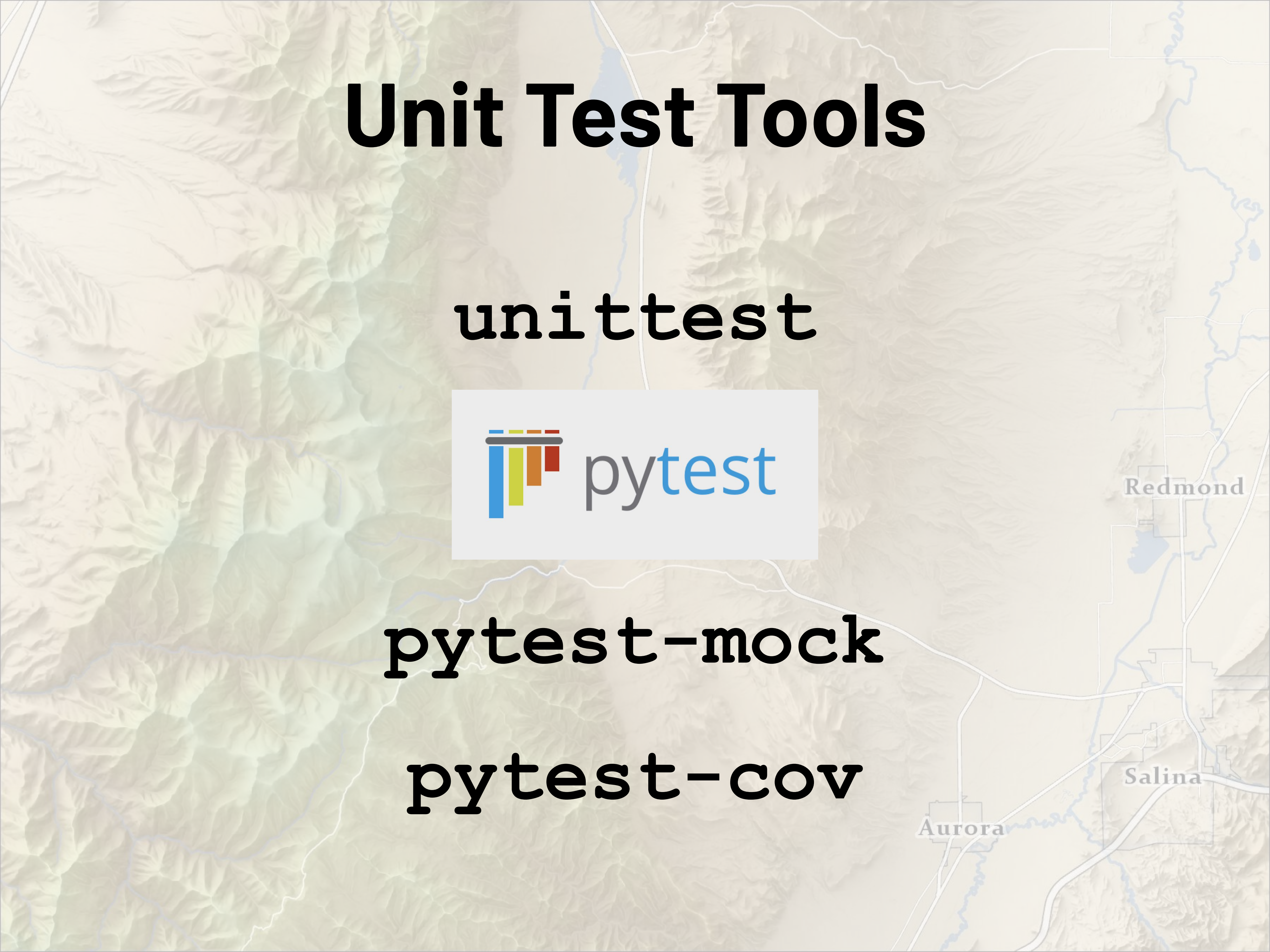


```
OUTLINE
> sizeof_fmt                 2
> stretch_scale              ●
> ProcessSuperArray          8
> lock_init                  2
  lock
> ParallelRCP               +9
  args
  all                        1
  kernel_args
  blur_gauss_args
  mdenoise_args
  clahe_args
  hs_args
  sky_args
  out_args
  arguments
  arg_dict
  input_DEM
```

Basic Steps of a Unit Test

- Arrange
- Act
- Assert





Unit Test Tools

`unittest`

pytest

`pytest-mock`

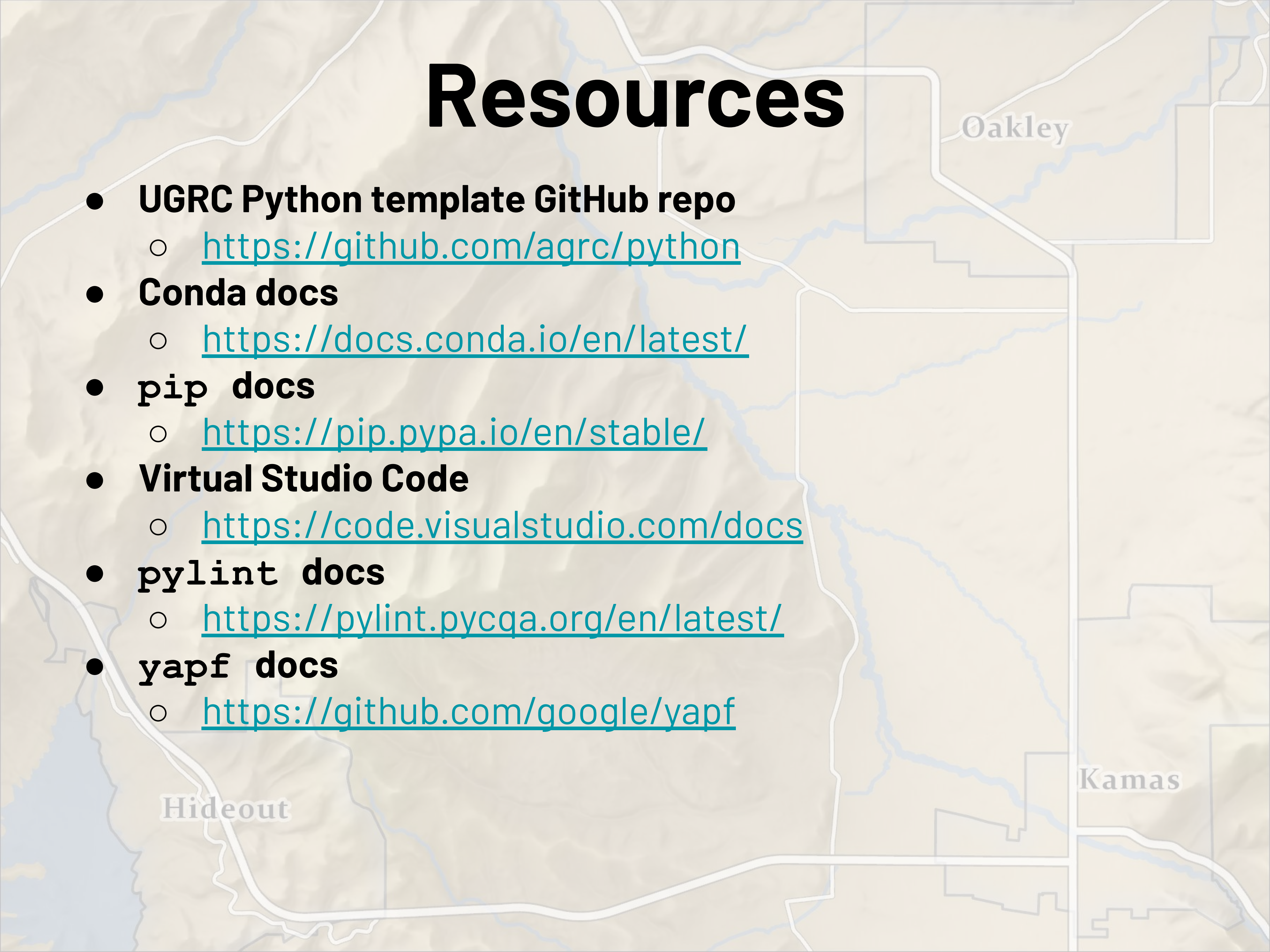
`pytest-cov`

UGRC

Utah Geospatial Resource Center

jdadams@utah.gov

gis.utah.gov/presentations

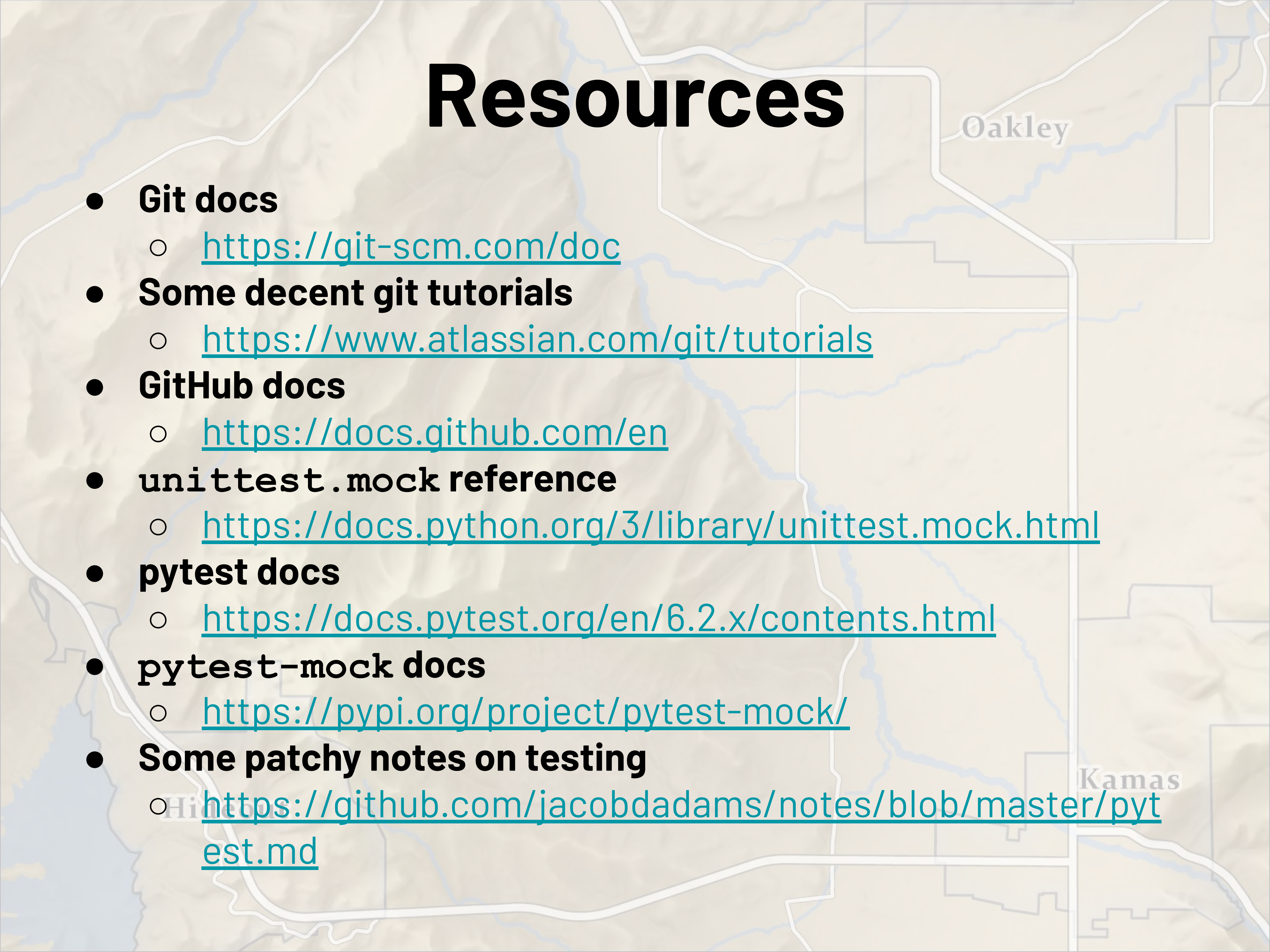


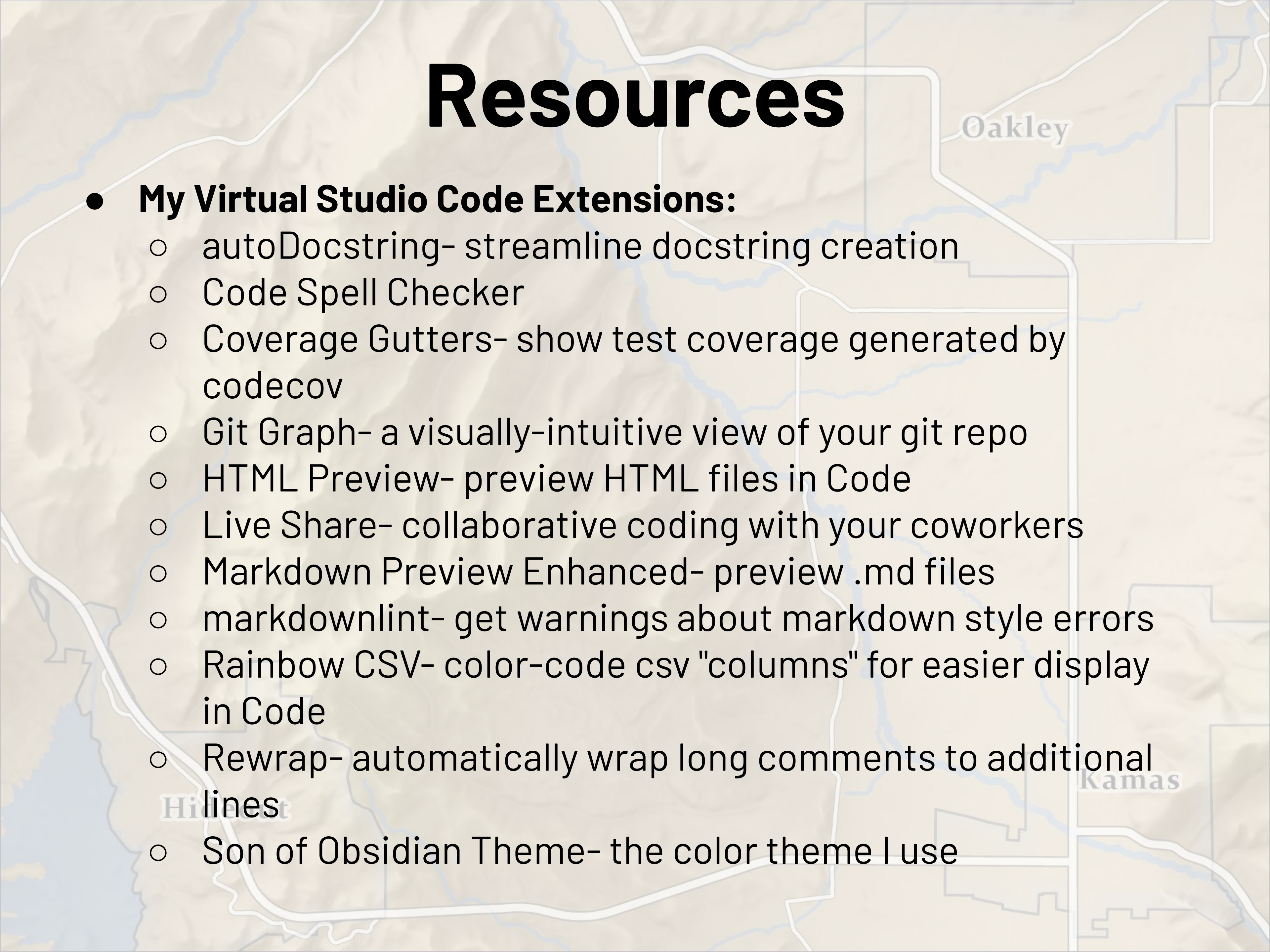
Resources

- **UGRC Python template GitHub repo**
  - https://github.com/agrc/python
- **Conda docs**
  - https://docs.conda.io/en/latest/
- **`pip` docs**
  - https://pip.pypa.io/en/stable/
- **Virtual Studio Code**
  - https://code.visualstudio.com/docs
- **`pylint` docs**
  - https://pylint.pycqa.org/en/latest/
- **`yapf` docs**
  - https://github.com/google/yapf

Resources

- **Git docs**
  - https://git-scm.com/doc
- **Some decent git tutorials**
  - https://www.atlassian.com/git/tutorials
- **GitHub docs**
  - https://docs.github.com/en
- `unittest.mock` **reference**
  - https://docs.python.org/3/library/unittest.mock.html
- **pytest docs**
  - https://docs.pytest.org/en/6.2.x/contents.html
- `pytest-mock` **docs**
  - https://pypi.org/project/pytest-mock/
- **Some patchy notes on testing**
  - https://github.com/jacobdadams/notes/blob/master/pytest.md





Resources

- **My Virtual Studio Code Extensions:**
  - autoDocstring- streamline docstring creation
  - Code Spell Checker
  - Coverage Gutters- show test coverage generated by codecov
  - Git Graph- a visually-intuitive view of your git repo
  - HTML Preview- preview HTML files in Code
  - Live Share- collaborative coding with your coworkers
  - Markdown Preview Enhanced- preview .md files
  - markdownlint- get warnings about markdown style errors
  - Rainbow CSV- color-code csv "columns" for easier display in Code
  - Rewrap- automatically wrap long comments to additional lines
  - Son of Obsidian Theme- the color theme I use