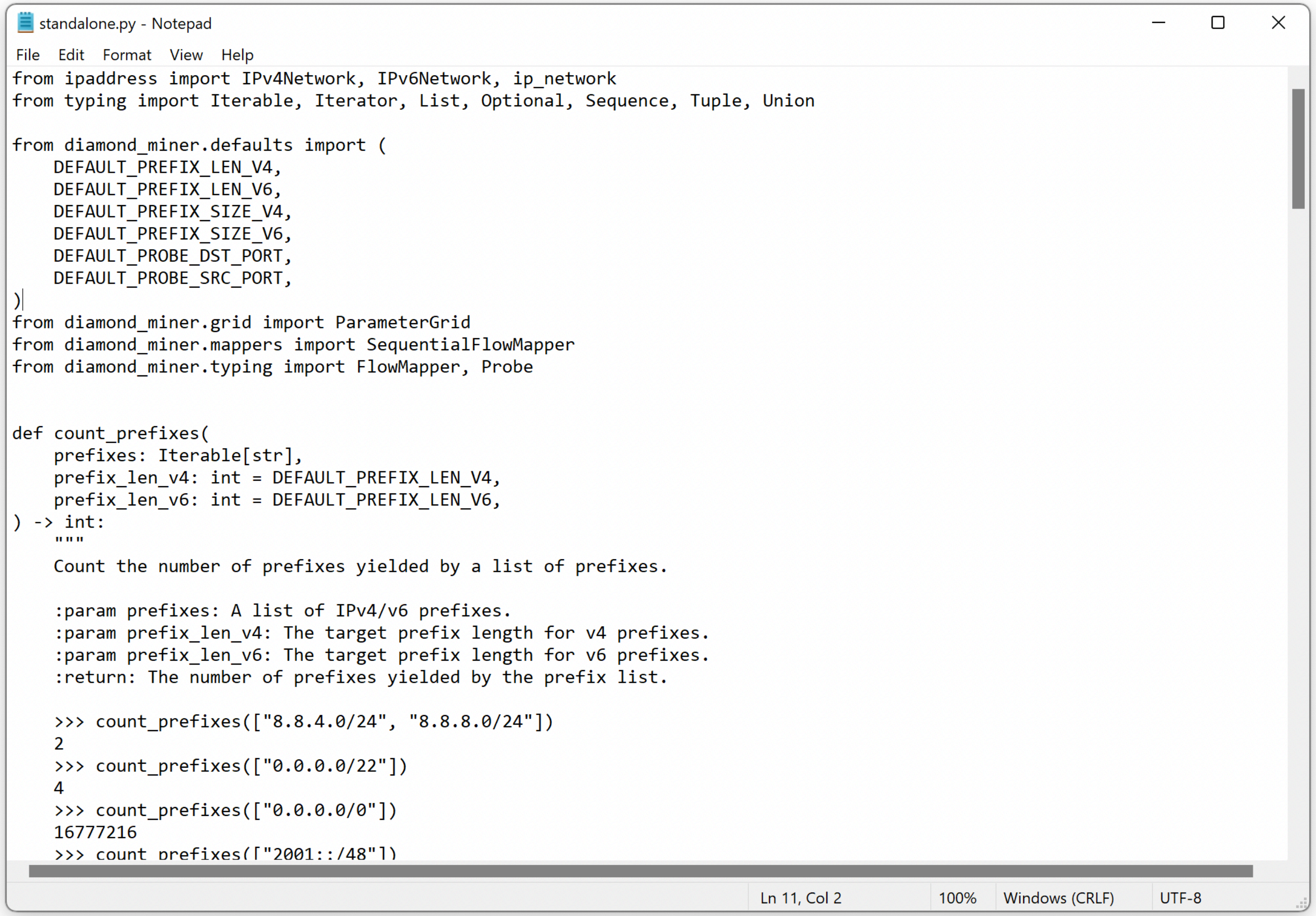
PyCharm

LINCS Python Workshop

From text editors to IDEs

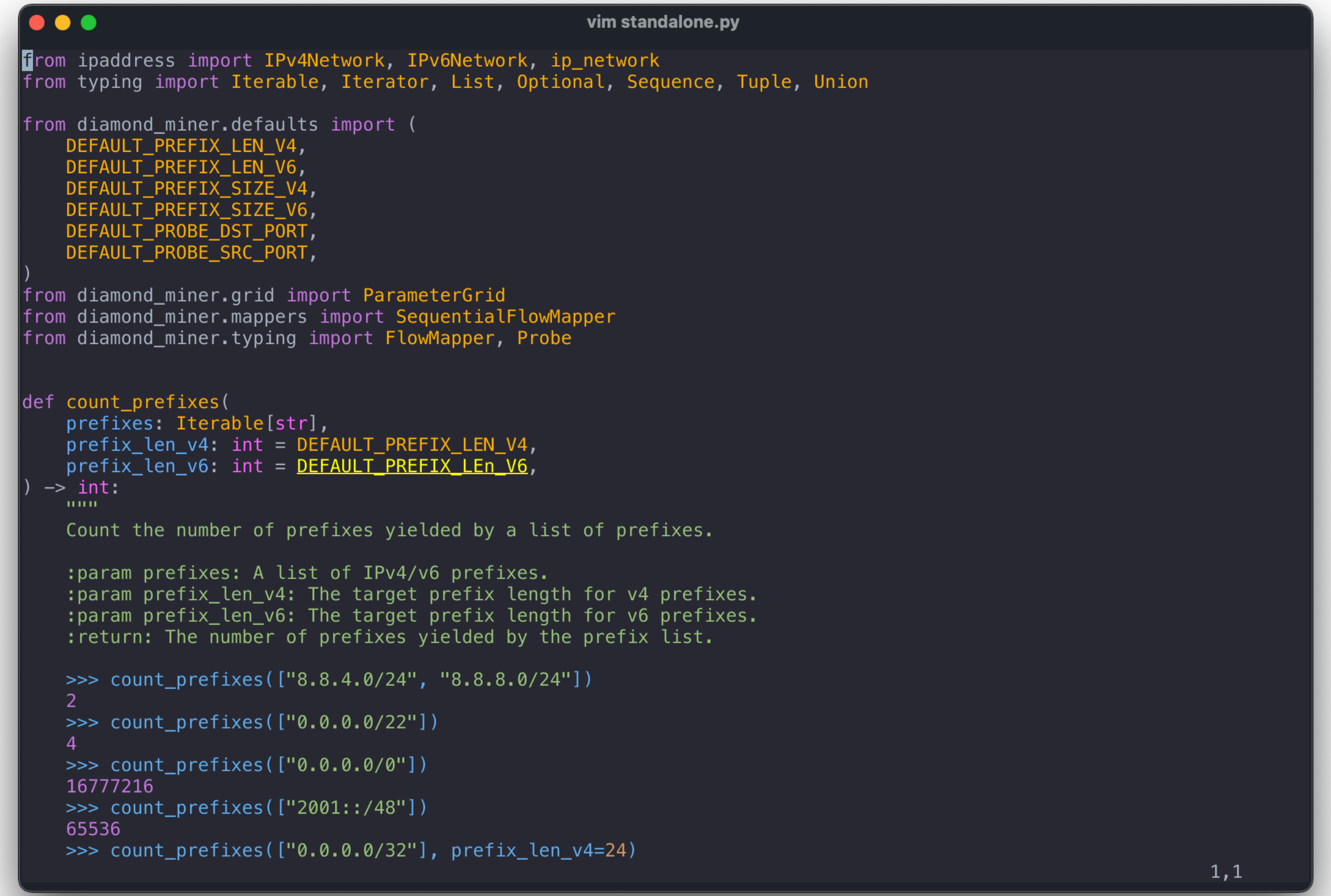


```
from ipaddress import IPv4Network, IPv6Network, ip_network
from typing import Iterable, Iterator, List, Optional, Sequence, Tuple, Union

from diamond_miner.defaults import (
    DEFAULT_PREFIX_LEN_V4,
    DEFAULT_PREFIX_LEN_V6,
    DEFAULT_PREFIX_SIZE_V4,
    DEFAULT_PREFIX_SIZE_V6,
    DEFAULT_PROBE_DST_PORT,
    DEFAULT_PROBE_SRC_PORT,
)
from diamond_miner.grid import ParameterGrid
from diamond_miner.mappers import SequentialFlowMapper
from diamond_miner.typing import FlowMapper, Probe


def count_prefixes(
    prefixes: Iterable[str],
    prefix_len_v4: int = DEFAULT_PREFIX_LEN_V4,
    prefix_len_v6: int = DEFAULT_PREFIX_LEN_V6,
) -> int:
    """
    Count the number of prefixes yielded by a list of prefixes.

    :param prefixes: A list of IPv4/v6 prefixes.
    :param prefix_len_v4: The target prefix length for v4 prefixes.
    :param prefix_len_v6: The target prefix length for v6 prefixes.
    :return: The number of prefixes yielded by the prefix list.

    >>> count_prefixes(["8.8.4.0/24", "8.8.8.0/24"])
    2
    >>> count_prefixes(["0.0.0.0/22"])
    4
    >>> count_prefixes(["0.0.0.0/0"])
    16777216
    >>> count_prefixes(["2001::/48"])
```

Text Editors

- Find / Replace

From text editors to IDEs

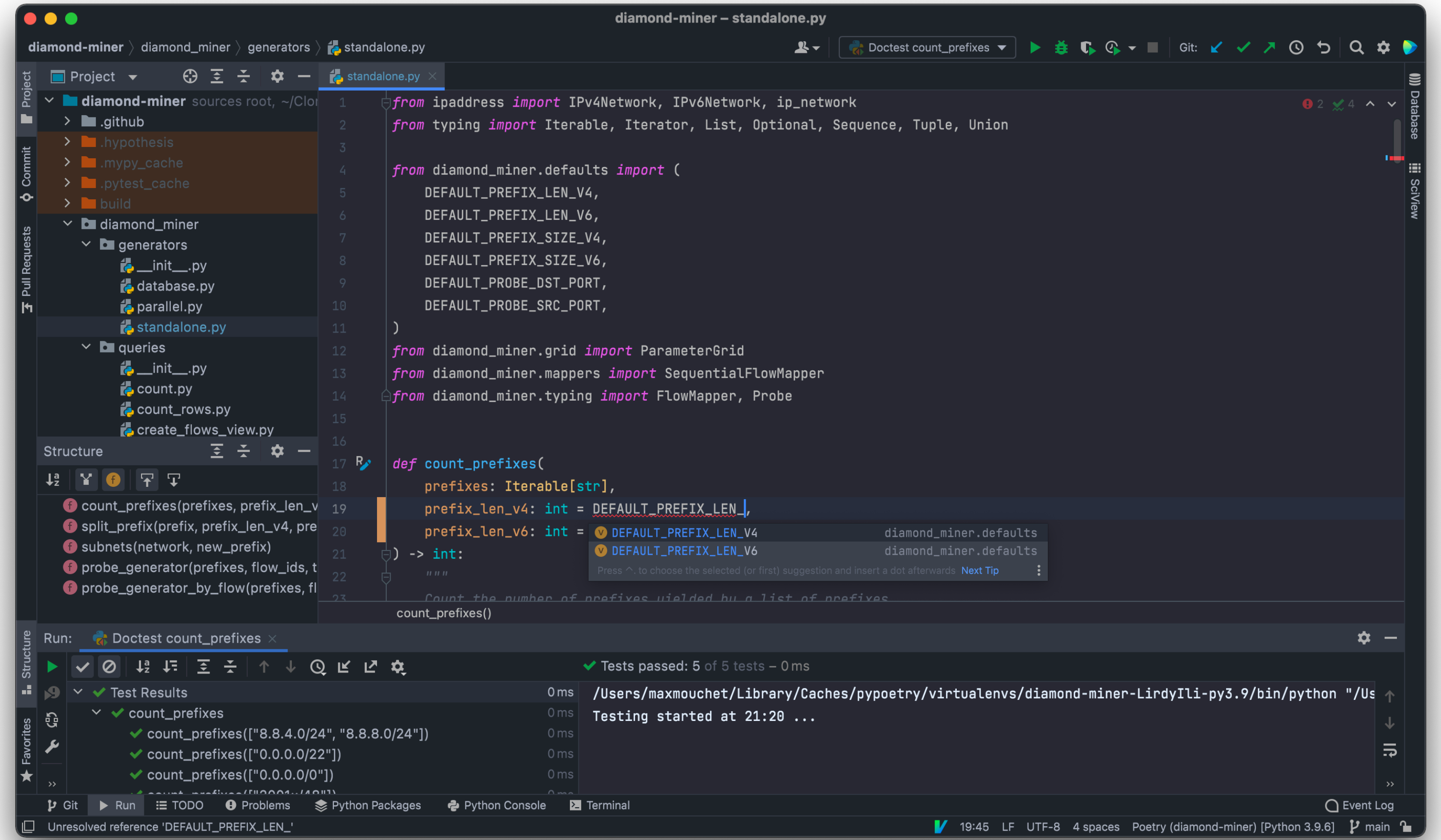


```
from ipaddress import IPv4Network, IPv6Network, ip_network
from typing import Iterable, Iterator, List, Optional, Sequence, Tuple, Union

from diamond_miner.defaults import (
    DEFAULT_PREFIX_LEN_V4,
    DEFAULT_PREFIX_LEN_V6,
    DEFAULT_PREFIX_SIZE_V4,
    DEFAULT_PREFIX_SIZE_V6,
    DEFAULT_PROBE_DST_PORT,
    DEFAULT_PROBE_SRC_PORT,
)
from diamond_miner.grid import ParameterGrid
from diamond_miner.mappers import SequentialFlowMapper
from diamond_miner.typing import FlowMapper, Probe


def count_prefixes(
    prefixes: Iterable[str],
    prefix_len_v4: int = DEFAULT_PREFIX_LEN_V4,
    prefix_len_v6: int = DEFAULT_PREFIX_LEn_V6,
) -> int:
    """
    Count the number of prefixes yielded by a list of prefixes.

    :param prefixes: A list of IPv4/v6 prefixes.
    :param prefix_len_v4: The target prefix length for v4 prefixes.
    :param prefix_len_v6: The target prefix length for v6 prefixes.
    :return: The number of prefixes yielded by the prefix list.

    >>> count_prefixes(["8.8.4.0/24", "8.8.8.0/24"])
    2
    >>> count_prefixes(["0.0.0.0/22"])
    4
    >>> count_prefixes(["0.0.0.0/0"])
    16777216
    >>> count_prefixes(["2001::/48"])
    65536
    >>> count_prefixes(["0.0.0.0/32"], prefix_len_v4=24)
                                                                    1,1
```

Code Editors

- Syntax highlighting
- Code completion
- Keyboard shortcuts
- Minimalist but extensible

From text editors to IDEs



```python
from ipaddress import IPv4Network, IPv6Network, ip_network
from typing import Iterable, Iterator, List, Optional, Sequence, Tuple, Union

from diamond_miner.defaults import (
    DEFAULT_PREFIX_LEN_V4,
    DEFAULT_PREFIX_LEN_V6,
    DEFAULT_PREFIX_SIZE_V4,
    DEFAULT_PREFIX_SIZE_V6,
    DEFAULT_PROBE_DST_PORT,
    DEFAULT_PROBE_SRC_PORT,
)
from diamond_miner.grid import ParameterGrid
from diamond_miner.mappers import SequentialFlowMapper
from diamond_miner.typing import FlowMapper, Probe


def count_prefixes(
    prefixes: Iterable[str],
    prefix_len_v4: int = DEFAULT_PREFIX_LEN_,
    prefix_len_v6: int = 
) -> int:
    """
    Count the number of prefixes yielded by a list of prefixes
```

```
DEFAULT_PREFIX_LEN_V4        diamond_miner.defaults
DEFAULT_PREFIX_LEN_V6        diamond_miner.defaults
Press ^. to choose the selected (or first) suggestion and insert a dot afterwards  Next Tip
```

```
✓ Tests passed: 5 of 5 tests – 0 ms

✓ Test Results                                   0 ms
  ✓ count_prefixes                               0 ms
    ✓ count_prefixes(["8.8.4.0/24", "8.8.8.0/24"]) 0 ms
    ✓ count_prefixes(["0.0.0.0/22"])             0 ms
    ✓ count_prefixes(["0.0.0.0/0"])              0 ms

/Users/maxmouchet/Library/Caches/pypoetry/virtualenvs/diamond-miner-LirdyIli-py3.9/bin/python "/Us
Testing started at 21:20 ...
```

```
Unresolved reference 'DEFAULT_PREFIX_LEN_'    19:45  LF  UTF-8  4 spaces  Poetry (diamond-miner) [Python 3.9.6]  main
```

IDEs

- All-in-one environment
- Run / Debug / Test
- Refactoring features

PyCharm favorites

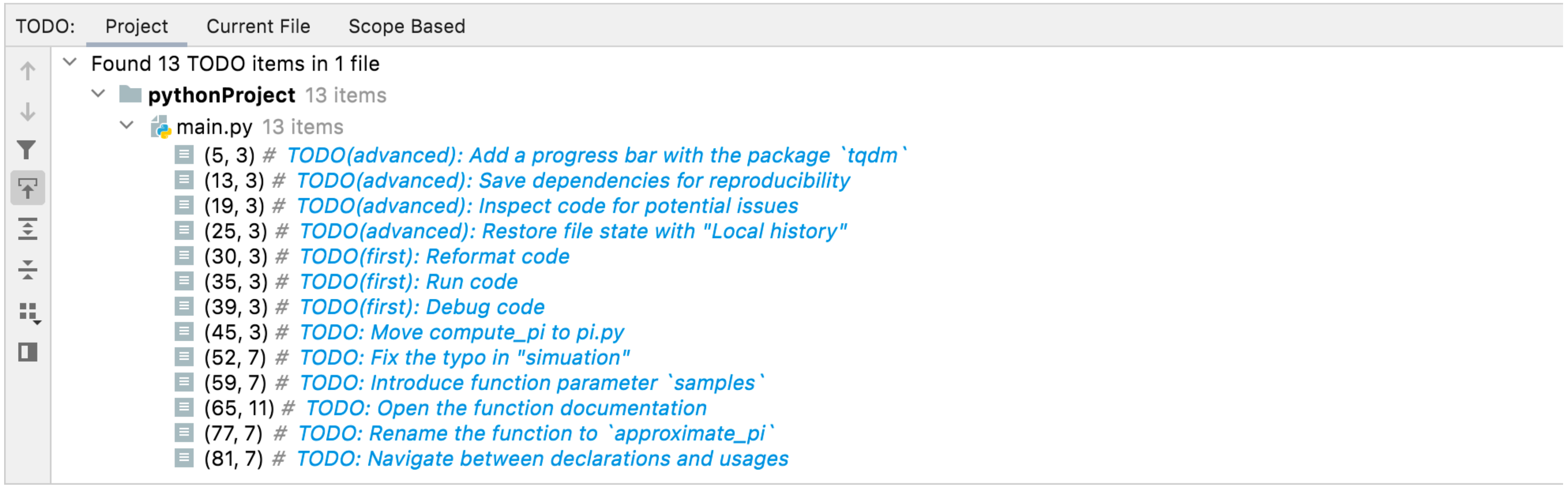
- Good code completion, navigation and refactoring features
    - Not easy for dynamically-typed languages such as Python!
- Completion and highlighting of other languages inside strings
- Database explorer and SQL completion based on the actual schema
- Run configurations and one-click debug/profile/coverage
- Python tests integration
- Good Vim-like plugin

PyCharm *less* favorites

- *Heavy*
  - 2GB vs 10MB of memory for Vim
- Not very convenient to edit files outside of a project
- Inferior support for remote development over SSH than Visual Studio Code
- IDEs tends to hide *low-level* details, such as Python virtual environments
  - It's important to also do things *manually* to understand how things works

Today’s plan

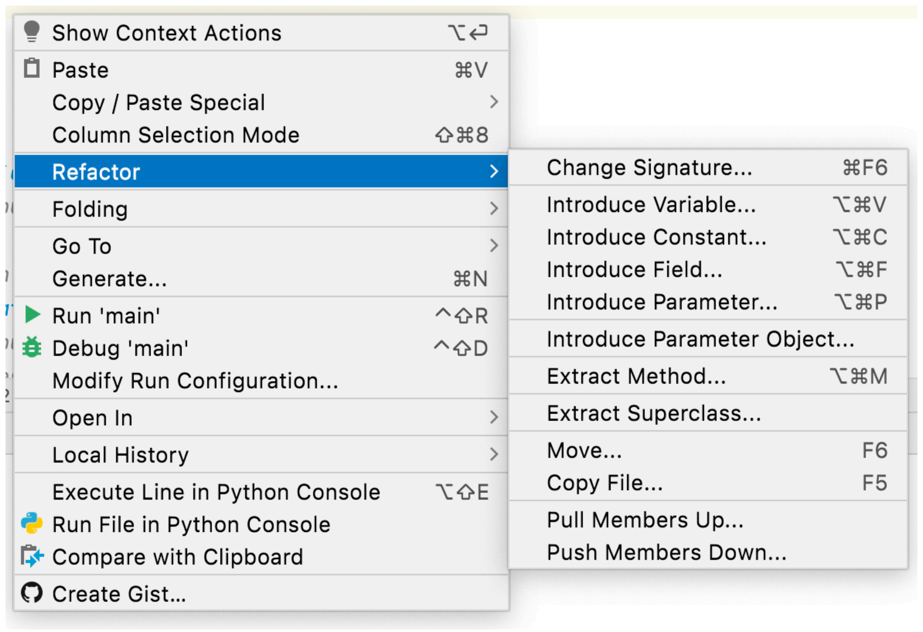
- PyCharm’s basics
- For more you can follow PyCharm’s integrated tutorial
  - Help > Learn IDE Features
- Progress on Google Sheets: shorturl.at/mEFTV

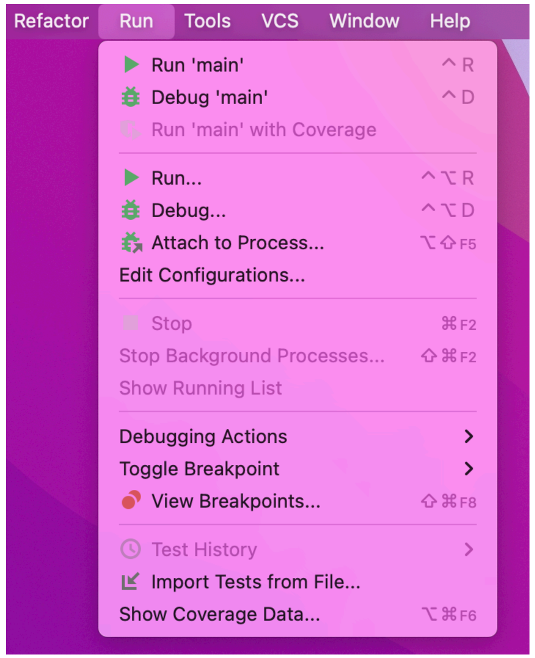


```
Found 13 TODO items in 1 file
  pythonProject 13 items
    main.py 13 items
      (5, 3) # TODO(advanced): Add a progress bar with the package `tqdm`
      (13, 3) # TODO(advanced): Save dependencies for reproducibility
      (19, 3) # TODO(advanced): Inspect code for potential issues
      (25, 3) # TODO(advanced): Restore file state with "Local history"
      (30, 3) # TODO(first): Reformat code
      (35, 3) # TODO(first): Run code
      (39, 3) # TODO(first): Debug code
      (45, 3) # TODO: Move compute_pi to pi.py
      (52, 7) # TODO: Fix the typo in "simuation"
      (59, 7) # TODO: Introduce function parameter `samples`
      (65, 11) # TODO: Open the function documentation
      (77, 7) # TODO: Rename the function to `approximate_pi`
      (81, 7) # TODO: Navigate between declarations and usages
```

A note on keyboard shortcuts

- Windows/Linux shortcuts are given first
- macOS shortcuts are given between parentheses if they differ
- https://plugins.jetbrains.com/plugin/9792-key-promoter-x





Let’s get started

1. Install PyCharm Community Edition
   - https://www.jetbrains.com/pycharm/download
2. Open the progress sheet
   - https://shorturl.at/mEFTV