



# Hello Bug, My Old Friend

**Patrick Deutschmann  
30 May 2025**

# About me

## Patrick Deutschmann



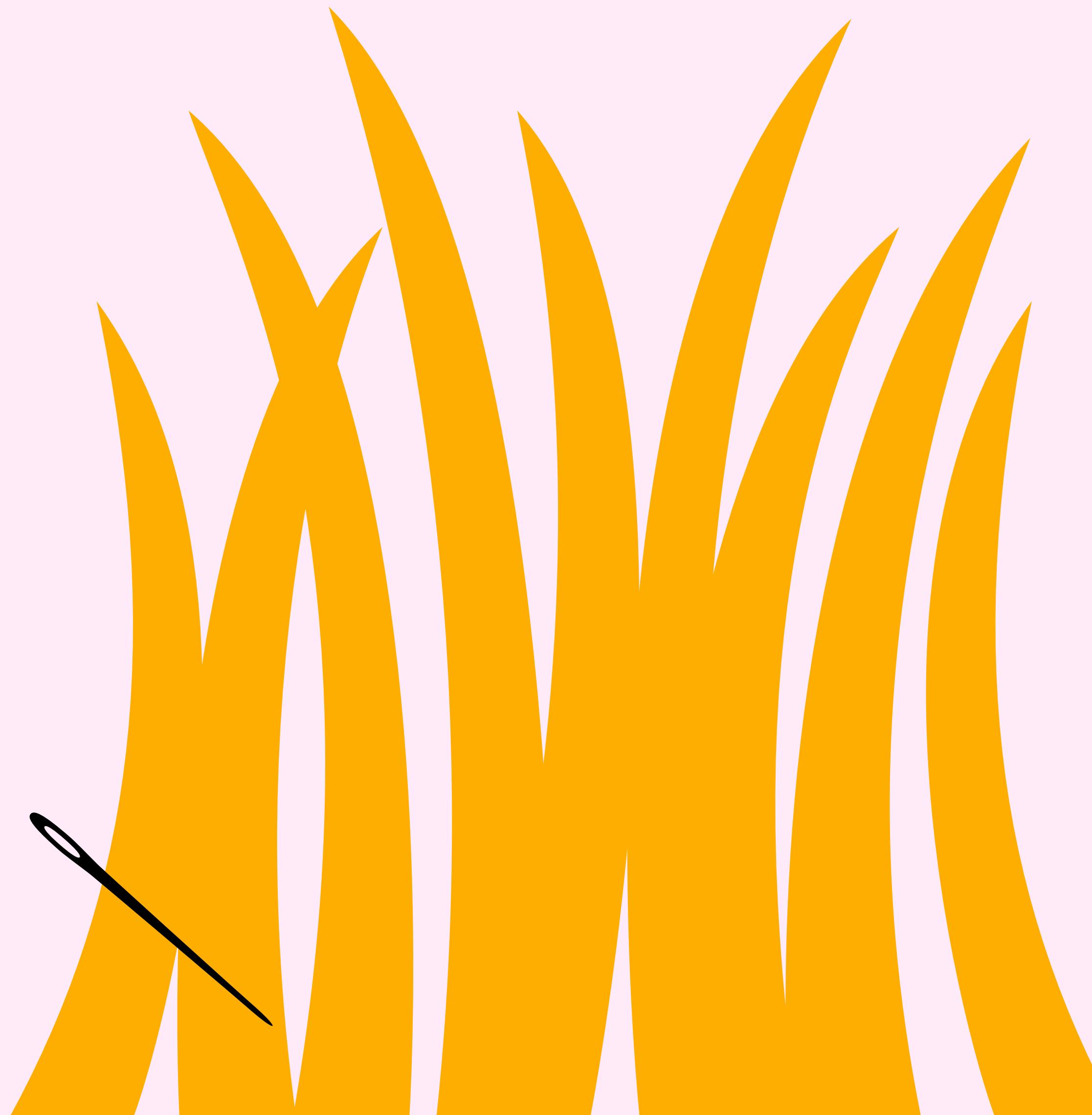
- Based in Graz, Austria
- Python / ML / AI
- ML Engineer @ Dedalus
- AI for medical software in Python



# I love debugging\*

\*most of the time

# Debugging



Find 

=

Find the needle  
in the haystack

# About this talk

The Best Way to Debug™

1

Technique

2

Tools

3

Mindsets

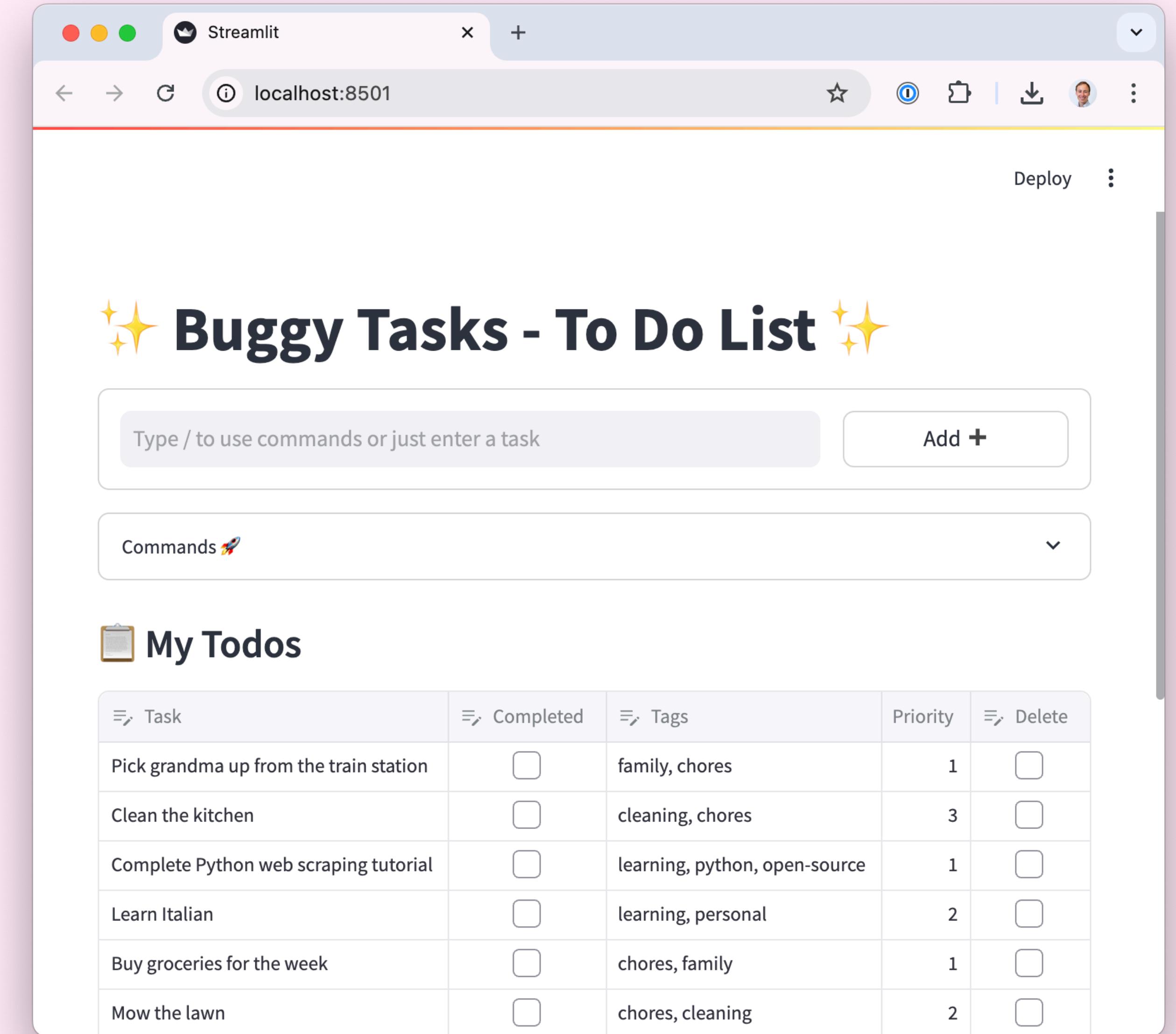
+ 2 Italian food references

# Demo project

vibe  
coding

# Demo project

 [deutschmn/buggy-tasks](https://github.com/deutschmn/buggy-tasks)



The screenshot shows a Streamlit application running at `localhost:8501`. The title is **Buggy Tasks - To Do List**. The interface includes a search bar with placeholder text "Type / to use commands or just enter a task" and an "Add +" button. A dropdown menu labeled "Commands" is open. Below this is a section titled "My Todos" with a clipboard icon. A table lists six tasks:

Task	Completed	Tags	Priority	Delete
Pick grandma up from the train station	<input type="checkbox"/>	family, chores	1	<input type="checkbox"/>
Clean the kitchen	<input type="checkbox"/>	cleaning, chores	3	<input type="checkbox"/>
Complete Python web scraping tutorial	<input type="checkbox"/>	learning, python, open-source	1	<input type="checkbox"/>
Learn Italian	<input type="checkbox"/>	learning, personal	2	<input type="checkbox"/>
Buy groceries for the week	<input type="checkbox"/>	chores, family	1	<input type="checkbox"/>
Mow the lawn	<input type="checkbox"/>	chores, cleaning	2	<input type="checkbox"/>

**Let's get down to business!**



# Delete

## My Todos

Task	Completed	Tags	Priority	Delete
Blog about Python 3.14	<input type="checkbox"/>	python, writing	3	<input type="checkbox"/>
Pick grandma up from the train station	<input type="checkbox"/>	family, chores	1	<input type="checkbox"/>
Clean the kitchen	<input type="checkbox"/>	cleaning, chores	3	<input type="checkbox"/>
Complete Python web scraping tutorial	<input type="checkbox"/>	learning, python, open-source	1	<input type="checkbox"/>
Learn Italian	<input type="checkbox"/>	learning, personal	2	<input type="checkbox"/>



# Delete fix

The screenshot shows a code editor interface with a search results panel on the left and the code file `app.py` on the right.

**Search Results:**

- SEARCH
- delete
- Aa ab \*
- ...
- 10 results in 1 file - [Open in editor](#)

**Code Editor:**

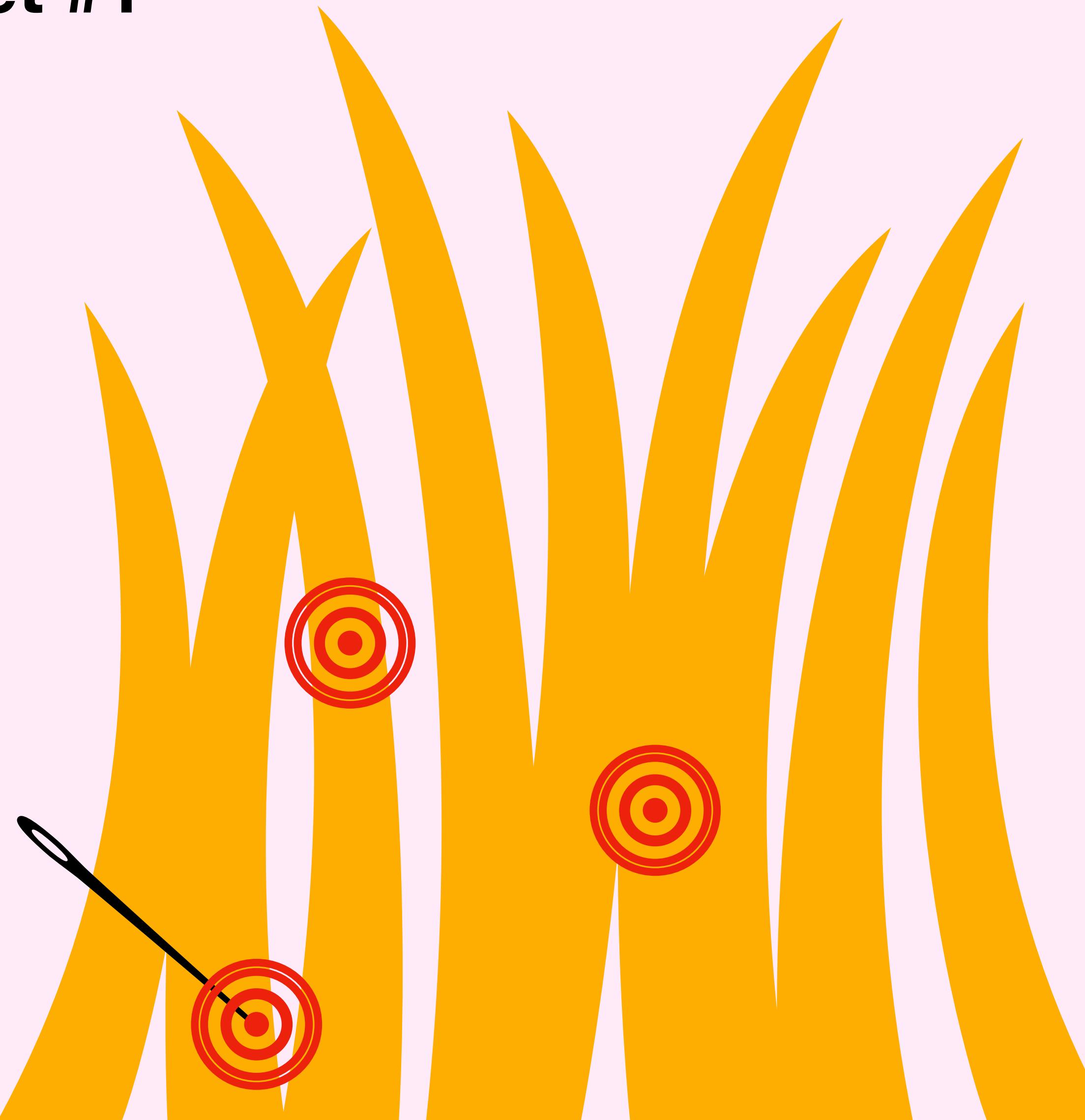
File: `app.py` (buggy\_tasks)

```
buggy_tasks > app.py > display.todos_with_data_editor > indices_to_delete
84 def display.todos_with_data_editor() -> None:
120
121     # Process any rows marked for deletion
122     rows_marked_for_deletion = edited_df[edited_df["Delete"]].index
123     # Check if any rows need to be deleted
124     if not rows_marked_for_deletion.empty:
125         # Filter out the todos that should be deleted using list comprehension
126         indices_to_delete: list[Any] = [i - 1 for i in rows_marked_for_deletion]
127         st.session_state.todos = [
128             todo for i, todo in enumerate(st.session_state.todos) if i not in indices_to_delete
129         ]
130         # Persist changes to storage
131         save.todos(st.session_state.todos)
132         # Refresh the UI to reflect changes
133         st.rerun()
134
135     # Apply any edits made in the data editor to our todos list
136     updated.todos_list: list = []
137     # Process each row in the edited dataframe
```

The code editor highlights several identifiers in orange, such as `buggy_tasks`, `app.py`, `display.todos_with_data_editor`, `indices_to_delete`, `edited_df`, `rows_marked_for_deletion`, `st`, `session_state`, `todos`, `todo`, `save.todos`, and `updated.todos_list`. A red box highlights the expression `[i - 1 for i in rows_marked_for_deletion]` in line 126.

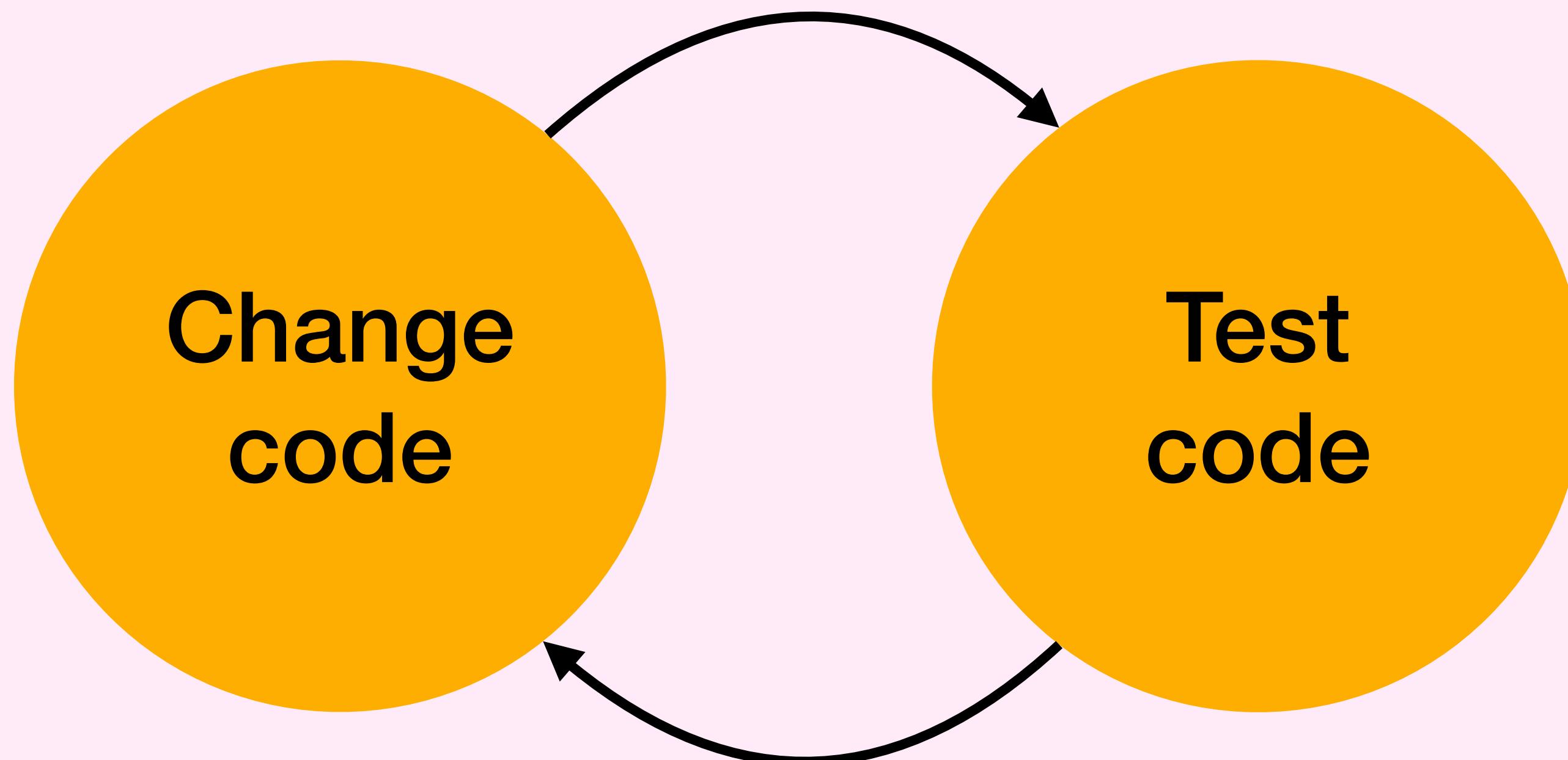
# Hawk debugging

## Mindset #1



Find the needle  
in the haystack  
fast and intuitively

# Debugging cycle

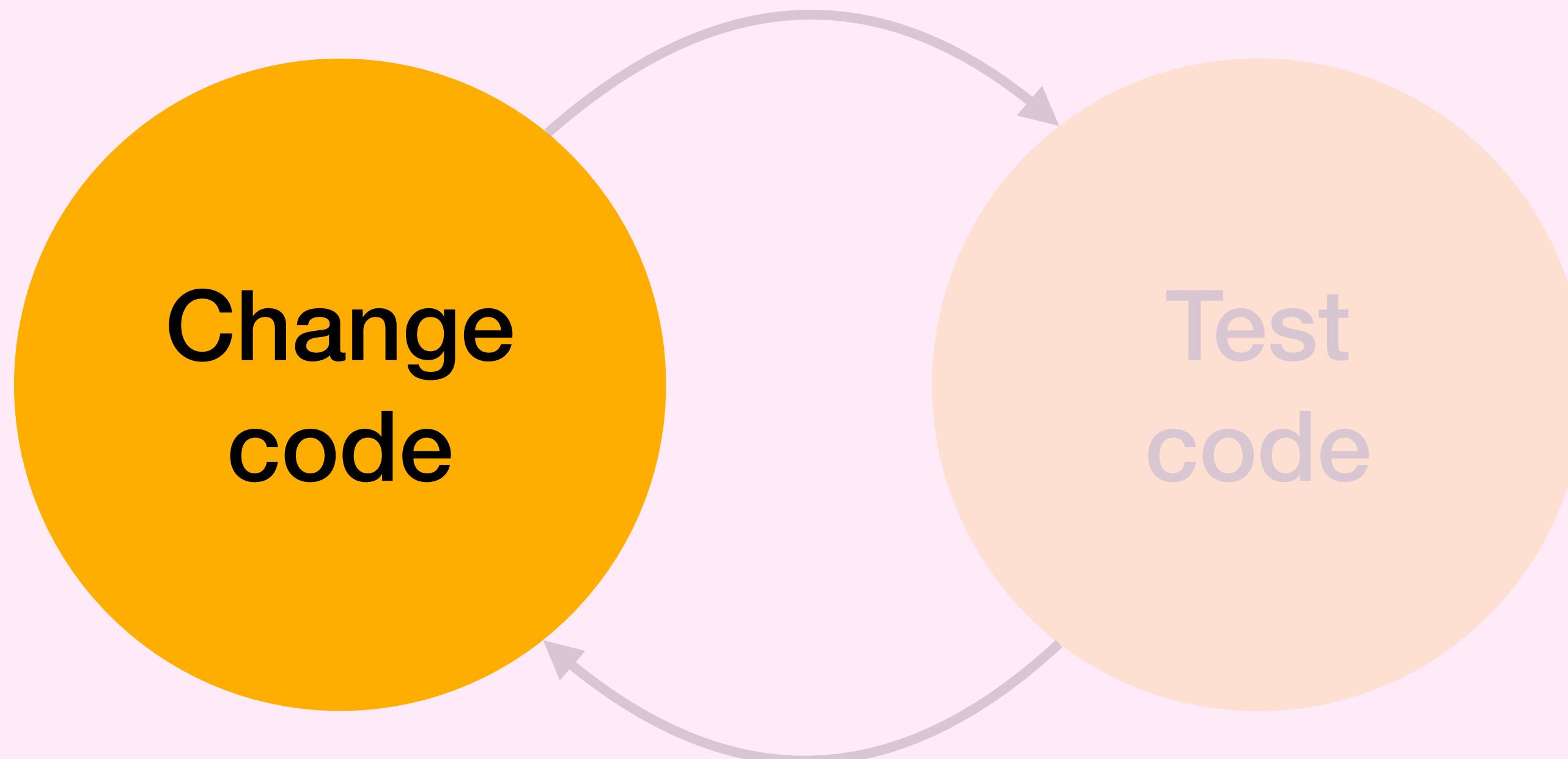


**Context switches ⚡**

→ slow cycles are bad

# Faster debugging cycle

## Technique #1



### Editor / IDE

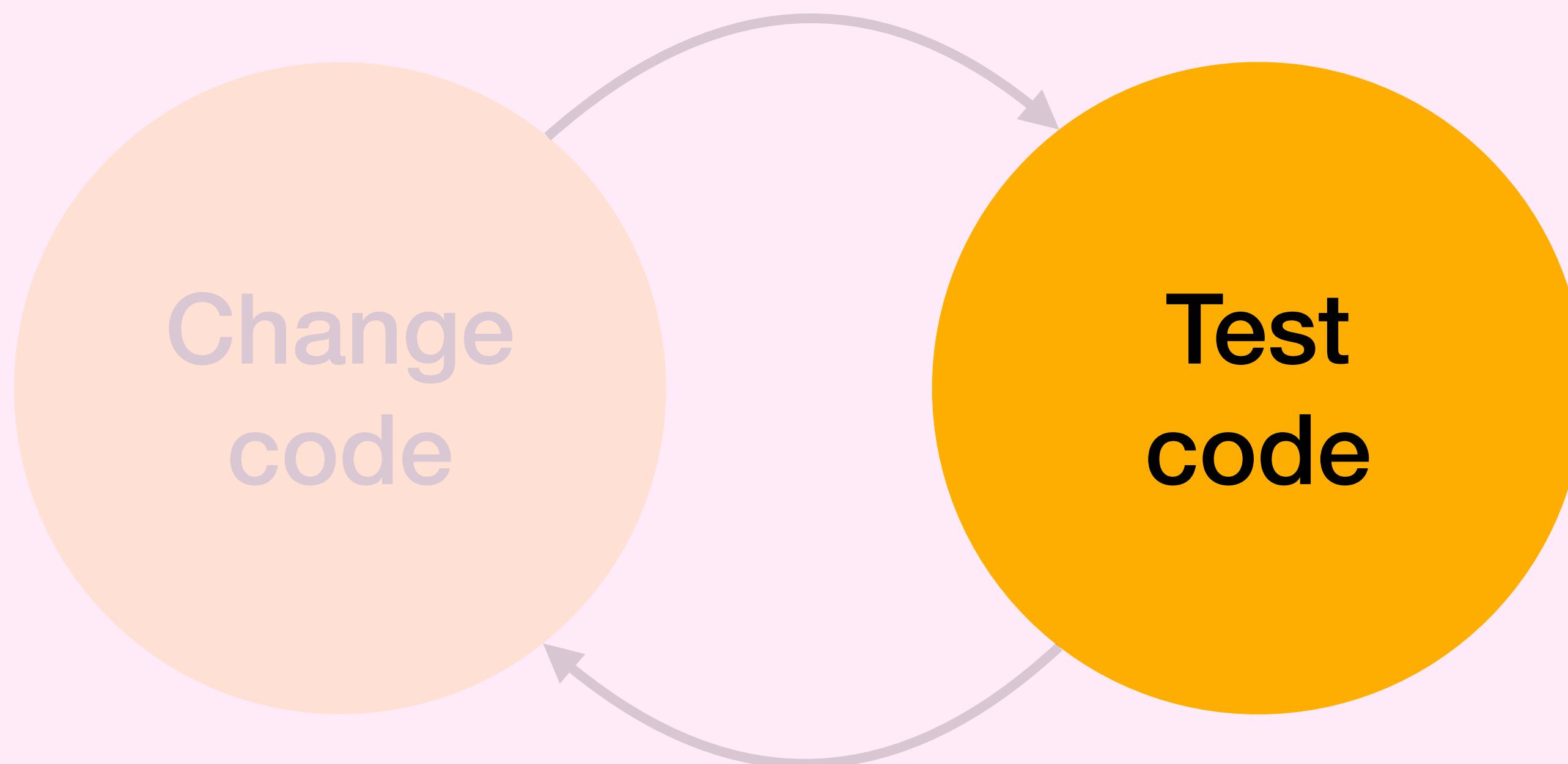
→ install your favourite

### Docker builds

→ mount or dev in container

# Faster debugging cycle

## Technique #1



### Startup time

→ only run parts of the code

### Manual test steps

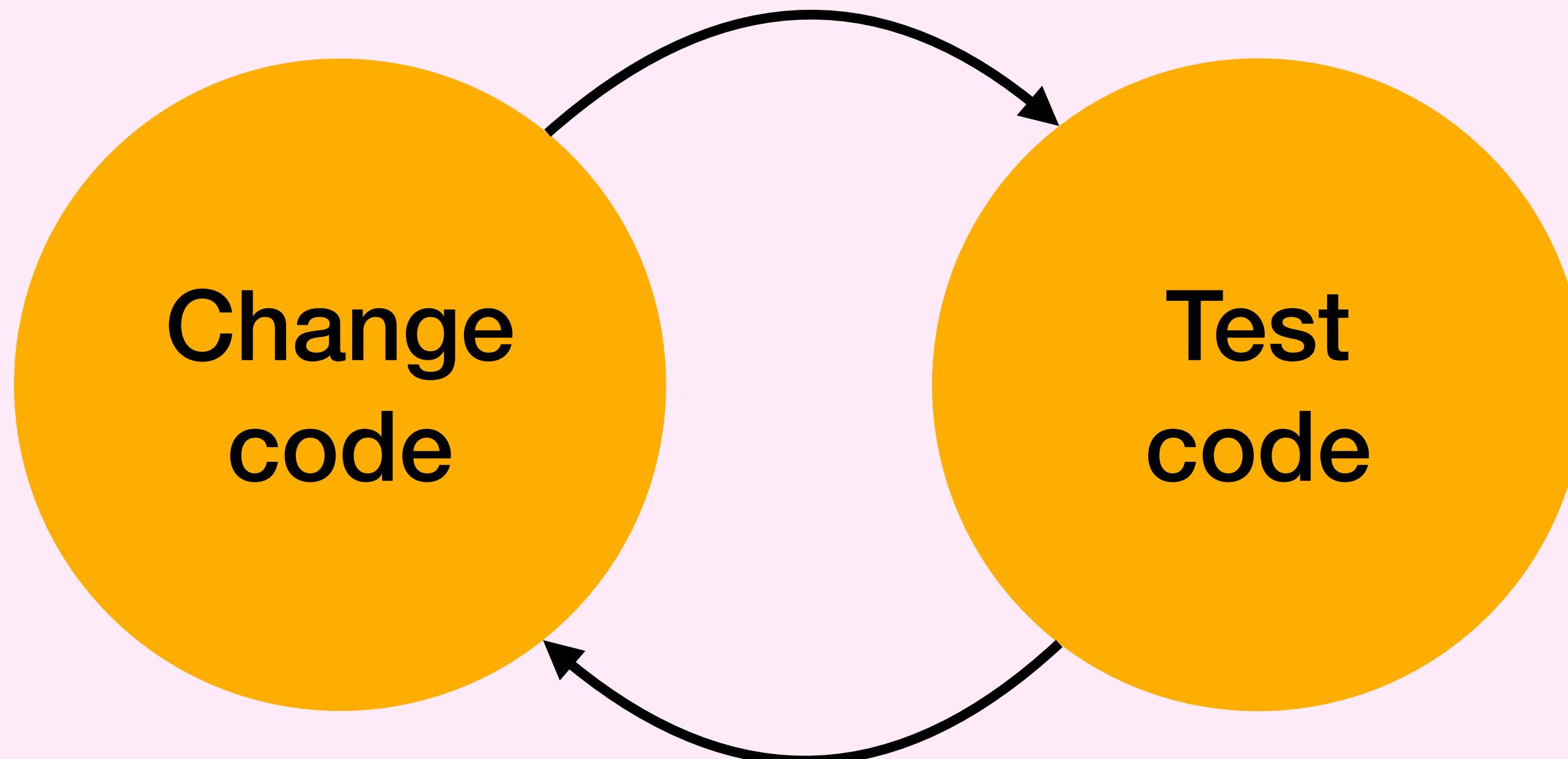
→ automated (unit) tests

### Big dataset triggers bug

→ minimise

# Faster debugging cycle

Technique #1



**Find what's slow.  
Make it faster!**

# IDE debugger

## Tool #1



### 1. Navigate call stack

### 2. Inspect variables

### 3. Try out stuff

```
#!/usr/bin/env /Users/pd/.pyenv/versions/buggy-tasks/bin/python /Users/pd/.vscode/extensions/ms-python.debugpy-2025.9.2025051201-darwin-arm64/bundled/libs/debugpy/adapter/../../../debugpy/launcher 62247 -- -m streamlit run /Users/pd/Work/public/pycon-it-2025/buggy-tasks/buggy_tasks/app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.20.10.2:8501
```

# Recap



Hawk debugging



Faster debugging cycle



IDE debugger



# Bad priority prediction

A screenshot of a "My Todos" application interface. The title "My Todos" is at the top left, accompanied by a clipboard icon. To the right is a search bar with icons for eye, download, search, and refresh. Below the title is a table with columns: Task, Completed (checkbox), Tags, Priority, and Delete. A single task is listed: "Prepare python presentation". The "Completed" column has an empty checkbox. The "Tags" column contains "work, python". The "Priority" column contains the number "3", which is highlighted with a red rectangular border. The "Delete" column has an empty checkbox.

Task	Completed	Tags	Priority	Delete
Prepare python presentation	<input type="checkbox"/>	work, python	3	<input type="checkbox"/>

**Expected: 1**

**Actual: 3**



# Bad priority prediction

fix

Let's hawk-debug A bald eagle with its wings spread, facing left.

- Tag extraction?  
→ no
- Model training process?  
→ no
- Inference code?  
→ no



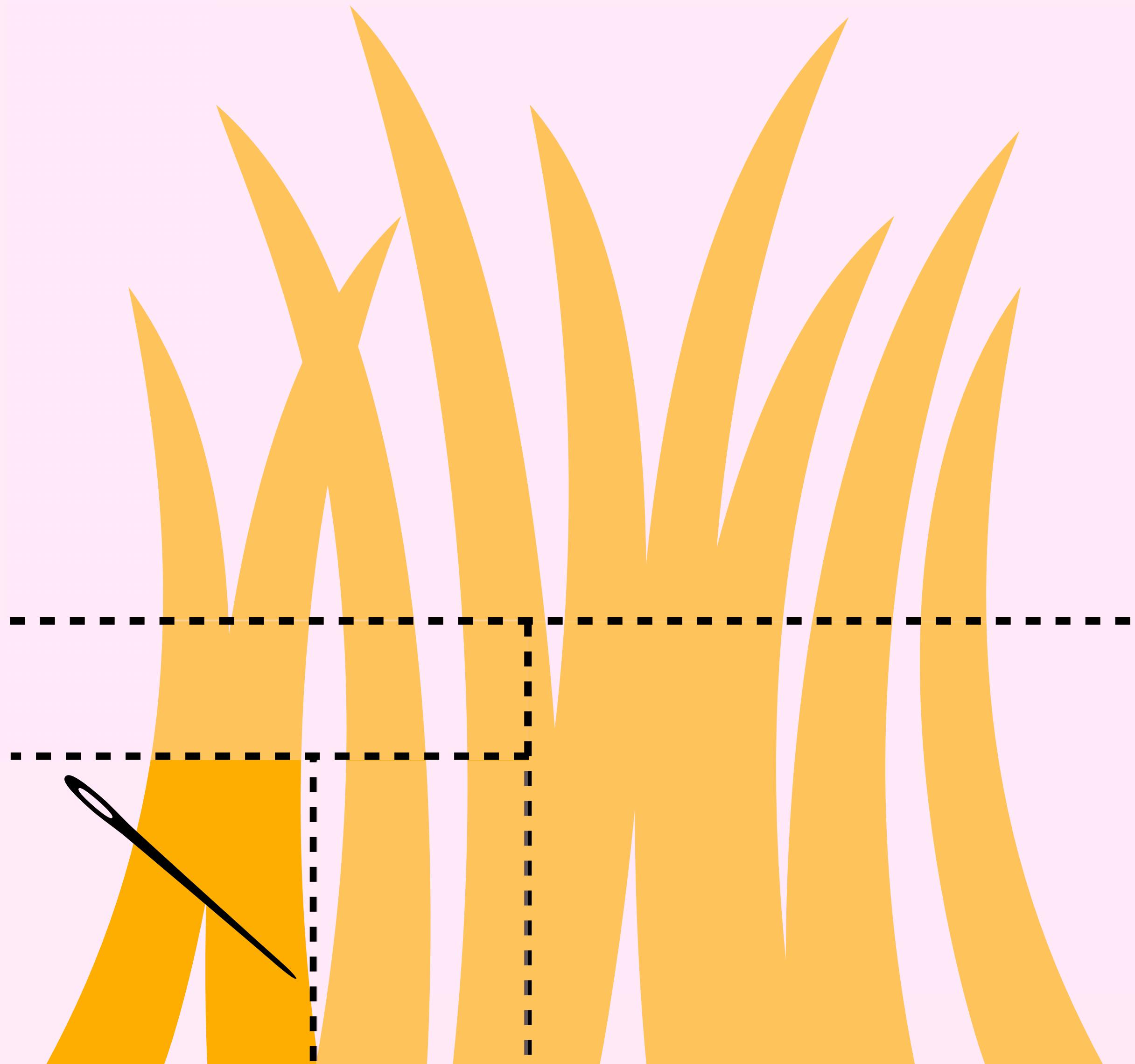
# Tortoise debugging

## Mindset #2



Find the needle  
in the haystack  
slowly and  
systematically

- 1) code parts 📁
- 2) change history 🕒

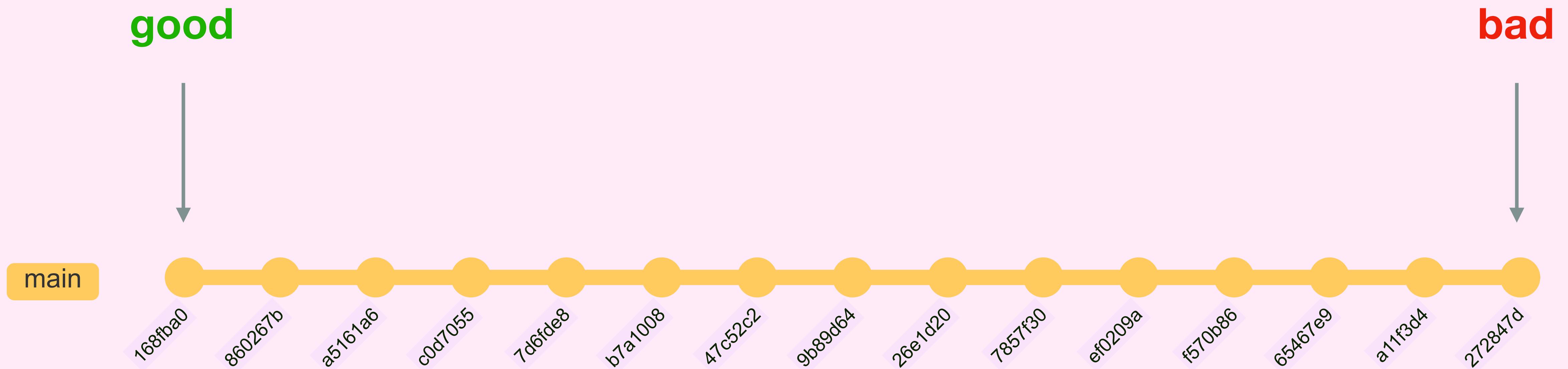


# git-bisect

## Tool #2



Which is the first  
**bad** commit?



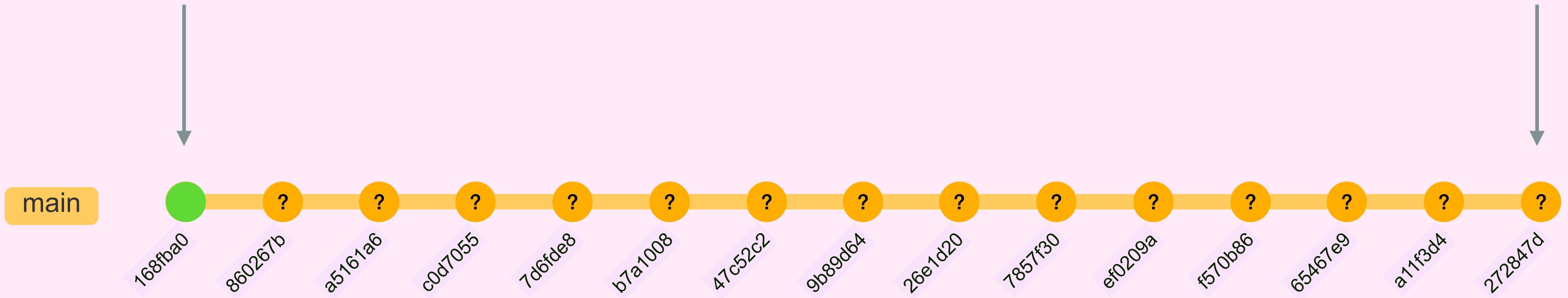
# git-bisect

## Tool #2



good

bad



# git-bisect

## Tool #2



# git-bisect

## Tool #2



# git-bisect

## Tool #2



# git-bisect

## Tool #2



# git-bisect

## Tool #2



```
# test_prio.py
from buggy_tasks.priority import (
    train_priority_model,
    compute_priority
)
def test_prio():
    train_priority_model()
    assert compute_priority(["python", "work"]) == 1
```

```
pytest ../test_prio.py
```

Test	Exit code	Commit
PASS	0	Good
FAIL	1	Bad

# git-bisect

## Tool #2

```
x - o  ✘  ✘  pd@patricksmac-2:~/Work/public/pycon-it-2025/buggy-tasks

> git bisect start
status: waiting for both good and bad commits
> git bisect bad 272847d
status: waiting for good commit(s), bad commit known
> git bisect good 168fba0
Bisecting: 6 revisions left to test after this (roughly 3 steps)
[9b89d645262f0faa602036db81cc93946644c823] Refactor registry
> git bisect run pytest ./test_prio.py
running 'pytest' './test_prio.py'
```

bisect found first bad commit



```
● ○ ● ✘  ✘  pd@patricksmac-2:~/Work/public/pycon-it-2025/buggy-tasks

> git bisect start
status: waiting for both good and bad commits
> git bisect bad 272847d
status: waiting for good commit(s), bad commit known
> git bisect good 168fba0
Bisecting: 6 revisions left to test after this (roughly 3 steps)
[9b89d645262f0faa602036db81cc93946644c823] Refactor registry
> git bisect run pytest ./test_prio.py
running 'pytest' './test_prio.py'

=====
platform darwin -- Python 3.9.7, pytest-6.2.3, py-1.10.0, pluggy-1.0.0
rootdir: /Users/pd/Work/public/pycon-it-2025
plugins: anyio-3.6.2
collected 1 item

./test_prio.py . [100%]

=====
1 passed in 43.94s =====
Bisecting: 3 revisions left to test after this (roughly 2 steps)
[ef0209af5d5la39633660588c378318c569f86e] Format 10
running 'pytest' './test_prio.py'

=====
platform darwin -- Python 3.9.7, pytest-6.2.3, py-1.10.0, pluggy-1.0.0
rootdir: /Users/pd/Work/public/pycon-it-2025
plugins: anyio-3.6.2
collected 1 item

./test_prio.py F [100%]

=====
FAILURES =====
test_prio

def test_prio():
    """Test the priority model training and prediction with a concrete example."""
    train_priority_model()

    result = compute_priority(["python", "work"])
    > assert result == 1, f"Expected priority 1, got {result}"
E AssertionError: Expected priority 1, got 3
E assert 3 == 1

./test_prio.py:9: AssertionError
=====
short test summary info =====
FAILED ./test_prio.py::test_prio - AssertionError: Expected priority 1, got 3
=====
1 failed in 1.12s =====
Bisecting: 0 revisions left to test after this (roughly 1 step)
[7857f30c81209c75731914ea9a84d6d3be323d] Rewrite translate
running 'pytest' './test_prio.py'

=====
platform darwin -- Python 3.9.7, pytest-6.2.3, py-1.10.0, pluggy-1.0.0
rootdir: /Users/pd/Work/public/pycon-it-2025
plugins: anyio-3.6.2
collected 1 item

./test_prio.py F [100%]

=====
FAILURES =====
test_prio

def test_prio():
    """Test the priority model training and prediction with a concrete example."""
    train_priority_model()

    result = compute_priority(["python", "work"])
    > assert result == 1, f"Expected priority 1, got {result}"
E AssertionError: Expected priority 1, got 3
E assert 3 == 1

./test_prio.py:9: AssertionError
=====
short test summary info =====
FAILED ./test_prio.py::test_prio - AssertionError: Expected priority 1, got 3
=====
1 failed in 0.87s =====
Bisecting: 0 revisions left to test after this (roughly 0 steps)
[26e1d20fe6f8eb1636917b85335e0de5993a7] Rename train data file
running 'pytest' './test_prio.py'

=====
platform darwin -- Python 3.9.7, pytest-6.2.3, py-1.10.0, pluggy-1.0.0
rootdir: /Users/pd/Work/public/pycon-it-2025
plugins: anyio-3.6.2
collected 1 item

./test_prio.py F [100%]

=====
FAILURES =====
test_prio

def test_prio():
    """Test the priority model training and prediction with a concrete example."""
    train_priority_model()

    result = compute_priority(["python", "work"])
    > assert result == 1, f"Expected priority 1, got {result}"
E AssertionError: Expected priority 1, got 3
E assert 3 == 1

./test_prio.py:9: AssertionError
=====
short test summary info =====
FAILED ./test_prio.py::test_prio - AssertionError: Expected priority 1, got 3
=====
1 failed in 0.85s =====
26e1d20fe6f8eb1636917b85335e0de5993a7 is the first bad commit
commit 26e1d20fe6f8eb1636917b85335e0de5993a7
Author: Patrick Deutsmann <patrick.deutsmann@dedalus.com>
Date: Tue May 13 10:33:27 2025 +0200

Rename train data file

buggy_tasks/priority.py |  8 --
data/reference.json   |   9 -
data/train-data.json  | 281 ++++++-
pyproject.toml        |   4 +
4 files changed, 286 insertions(+), 13 deletions(-)
create mode 100644 data/train-data.json
bisection found first bad commit
~/W/p/p/buggy-tasks @26e1d20fe6f8eb1636917b85335e0de5993a7 bisect >
```

57s ◆ buggy-tasks 3.10.14



# Bad priority prediction fix

Commit [26e1d20](#)

 deutschmn committed 5 days ago

Rename train data file

main (#9)

1 parent [9b89d64](#) commit [26e1d20](#) ↗

4 files changed +286 -13 lines changed

Search within code

> buggy_tasks/priority.py	↗	...	+4 -4
> data/reference.json	↗	...	-9
> data/train-data.json	↗	...	+281
> pyproject.toml	↗	...	+1



## Hawk debugging

Fast

Sometimes works

Emotionally draining



## Tortoise debugging

Slow

Always works

Steady progress

# Awareness

## Mindset #3

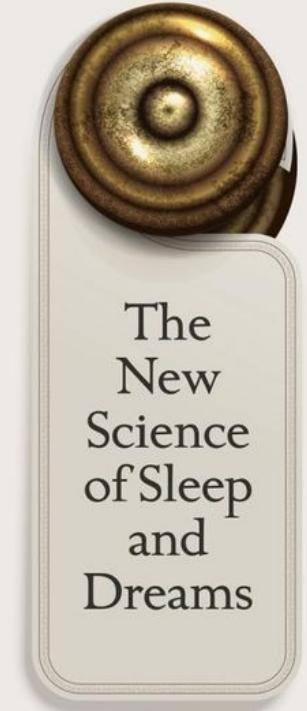
### Know your body

- Eat
- Drink
- Sleep



The International Bestseller  
'Startling, vital, a life raft' GUARDIAN

MATTHEW  
WALKER  
Why We  
Sleep



# Awareness

## Mindset #3



### Know yourself

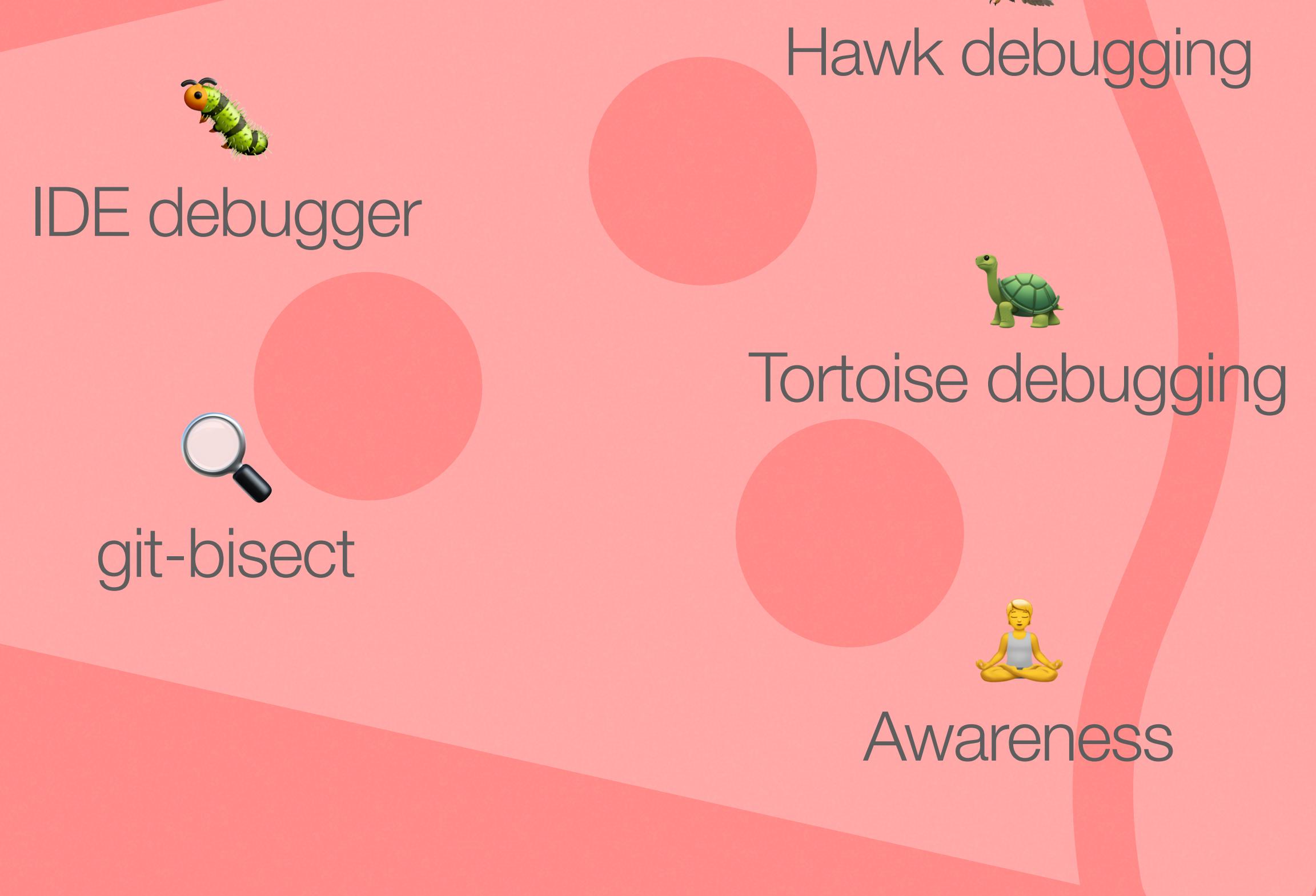
- When are you most productive during the day?
- Breaks
  - Frequent short
  - Long breaks
- Talk to people (or rubber or )

# Technique

Thank you!



Faster debugging cycle



# Tools

# Mindsets

# Questions?



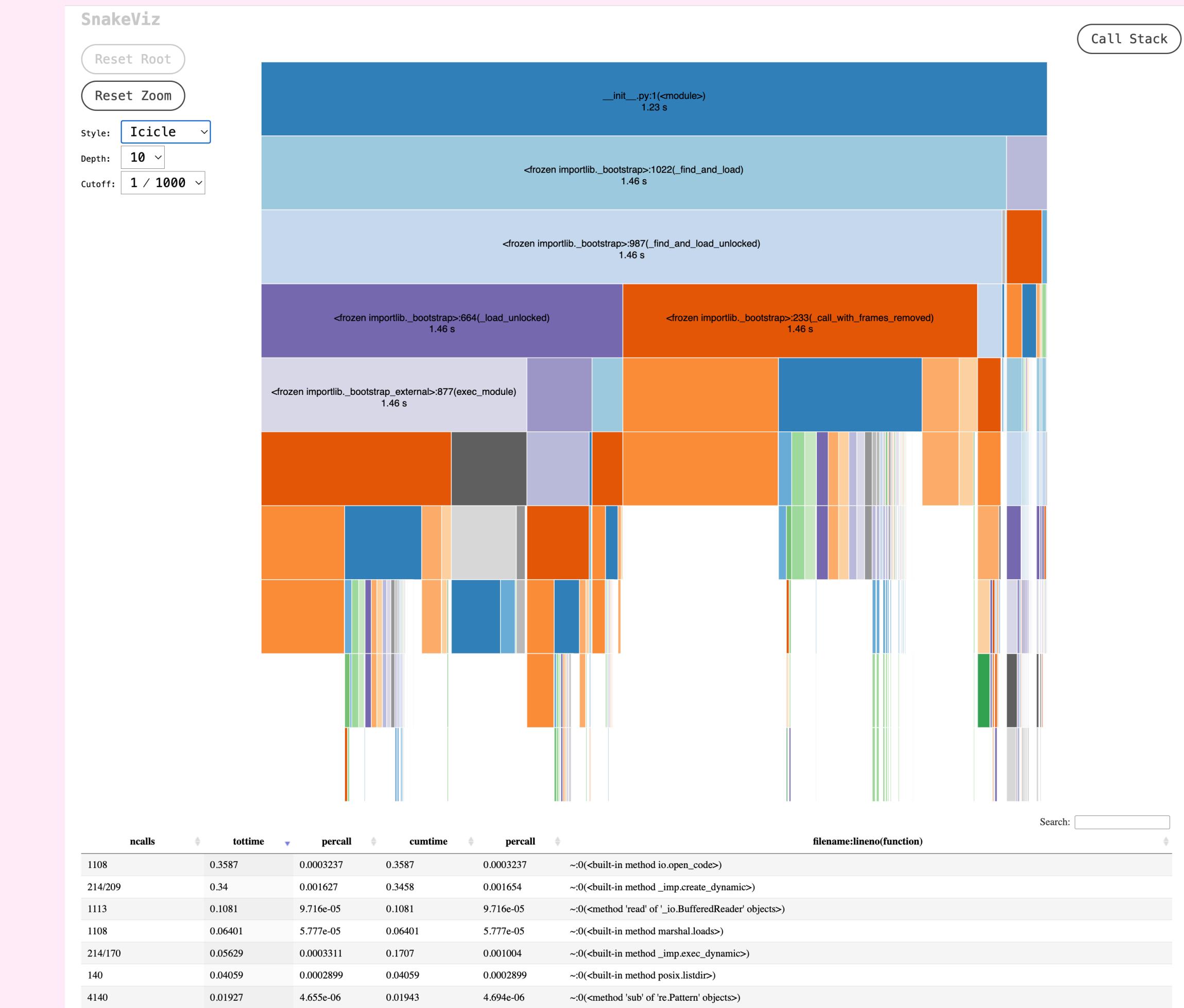
# Bonus: Non-functional bugs

Slow? 🐌

→ profiler (e.g. cProfile + snakeviz)

```
python -m cProfile -o out.prof train_model.py  
python -m snakeviz out.prof
```

 [jiffyclub/snakeviz](https://github.com/jiffyclub/snakeviz)





# Bonus: Non-functional bugs

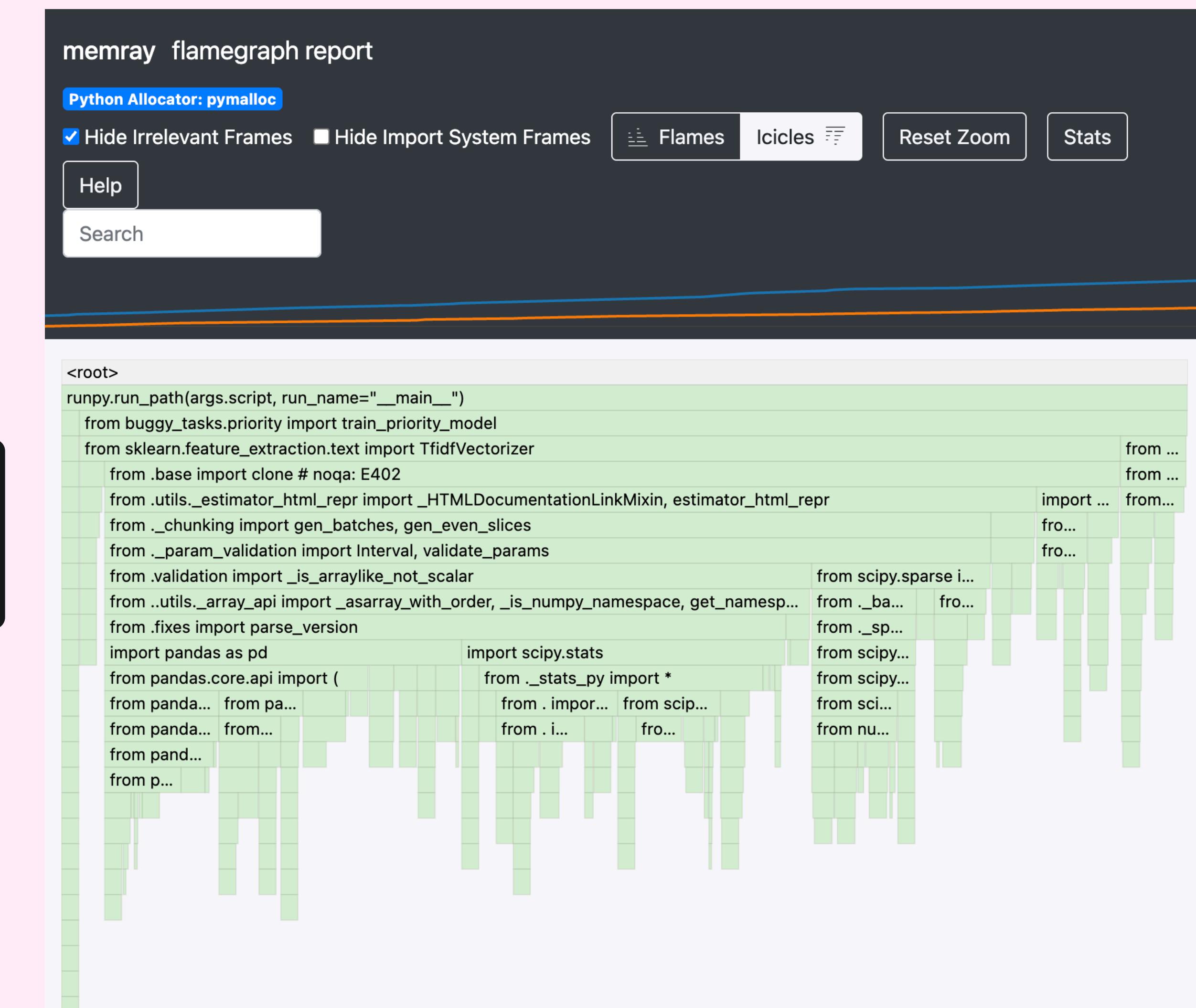
# High memory usage?

→ memory profiler (e.g. memray)

```
python -m memray run -o out.bin train_model.py  
python -m memray flamegraph out.bin
```



# bloomberg/memray



# Questions?