

Using GitHub for scientific research

Jon W. Carr

Language Evolution and Computation Research Unit
School of Philosophy, Psychology and Language Sciences
University of Edinburgh



Three good reasons to use GitHub

Version control

Share and collaborate

Publish your work

github pages fork
master repository atom
.gitignore markdown public collaborator
sync command line tools SHA hash
commit blame
branch merge
pull request issues history
clone wiki gist
private git
release

Today's workshop

Background

Specifics

Demonstration

Collaborative project

Quiz



GitHub

GitHub Education

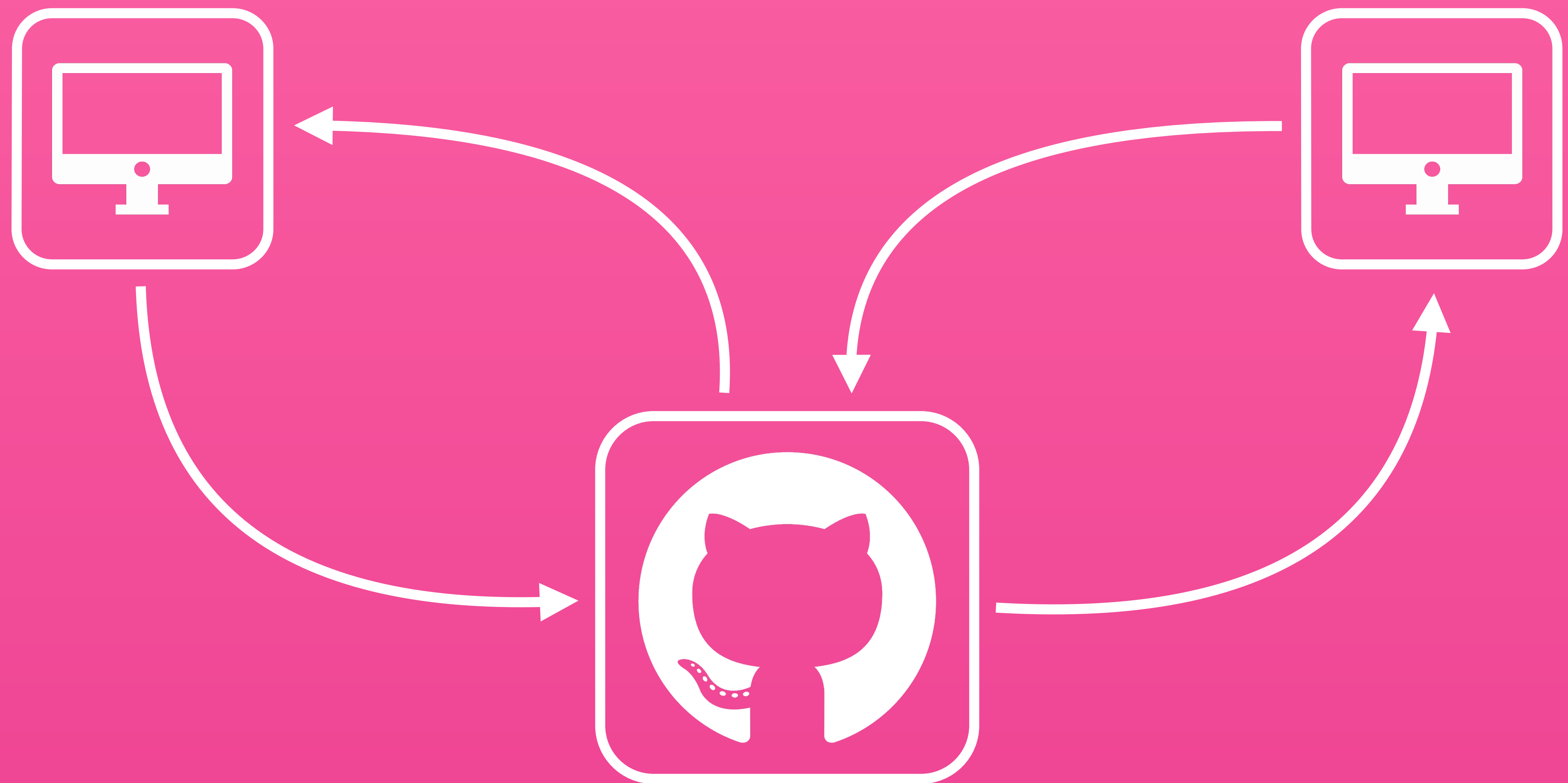


Free micro account for students

Host 5 private repositories

Usually \$7 per month

Big picture



Key terms



Repository



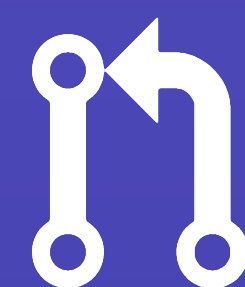
Commit



Branch

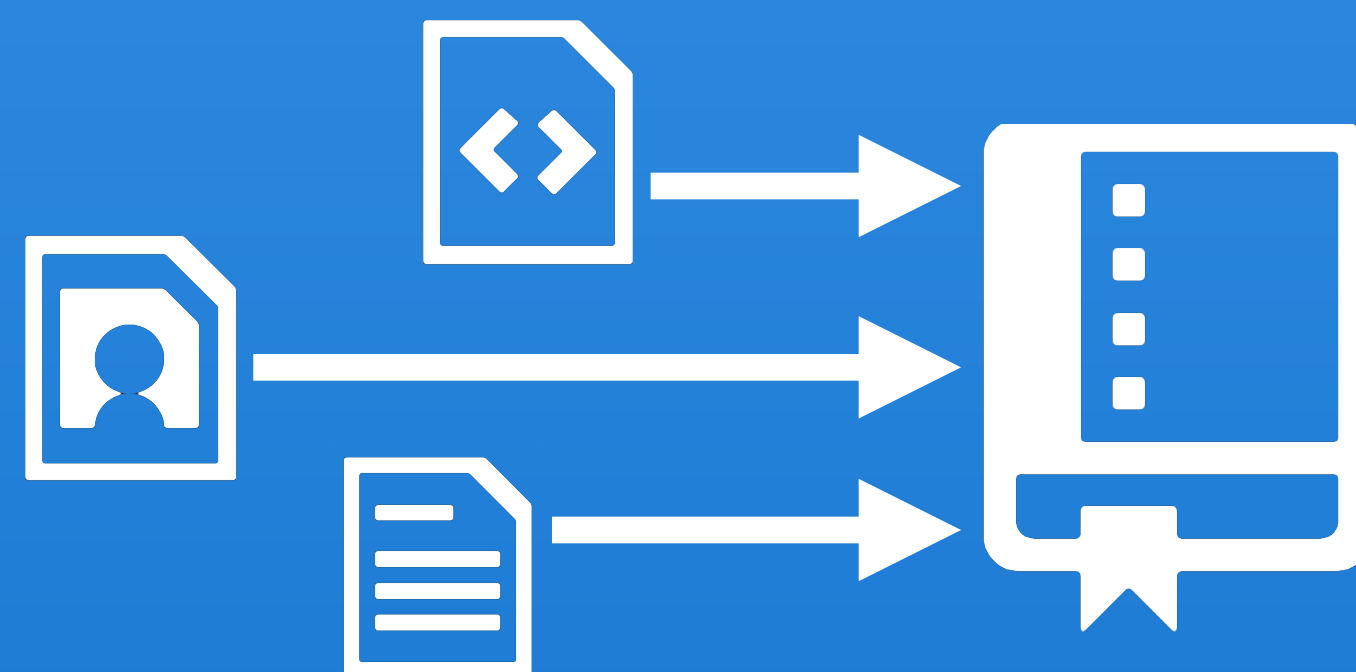


Fork

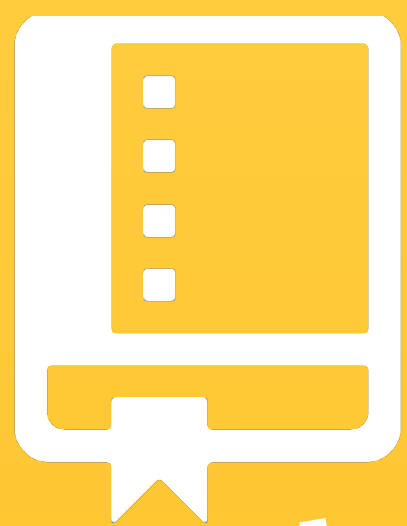


Pull request

Repository



+ Commit



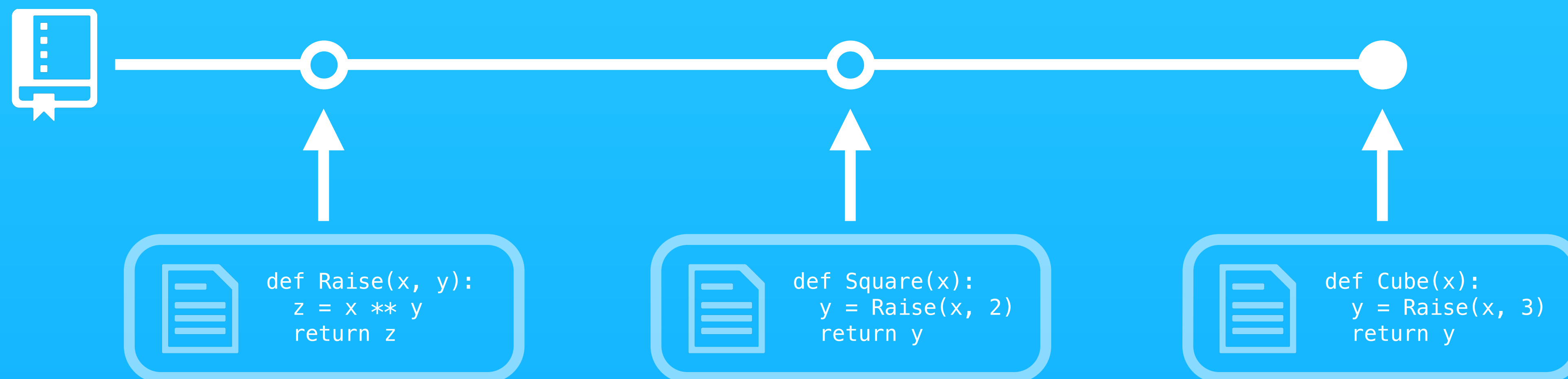
```
1 def Hello(name):  
2     greeting = "Hello " + name  
3     print greeting
```

+ Commit

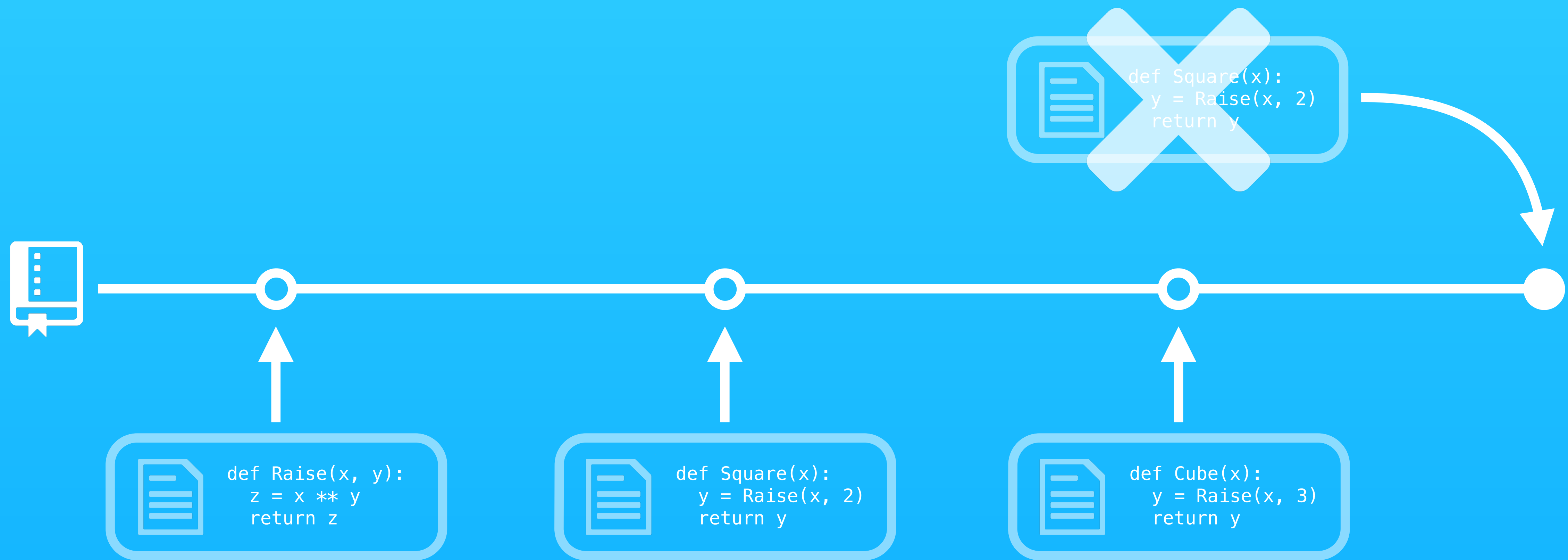


```
1 def Hello(name):  
2     greeting = "Hello " + name  
3     print greeting  
4  
5 def Bye(name):  
6     valediction = "Bye bye " + name  
7     print valediction
```

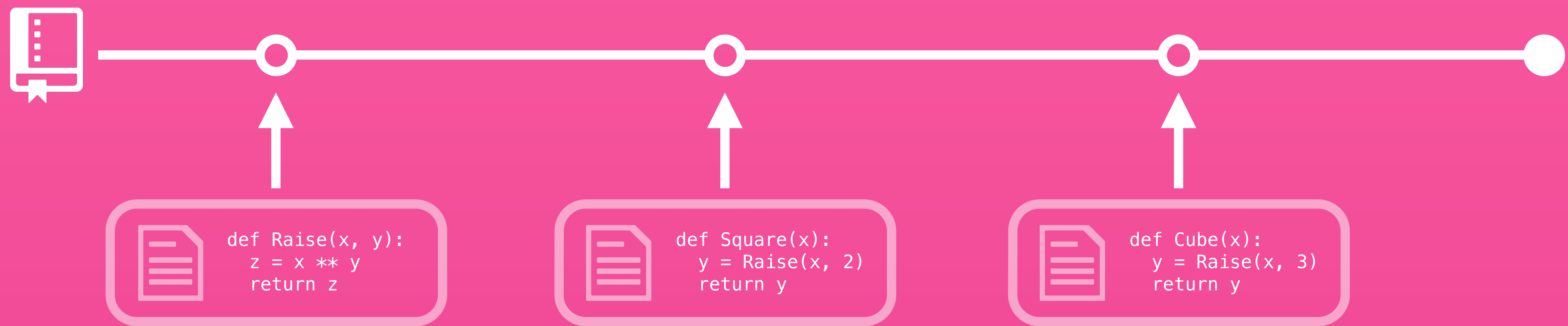
Revert a commit



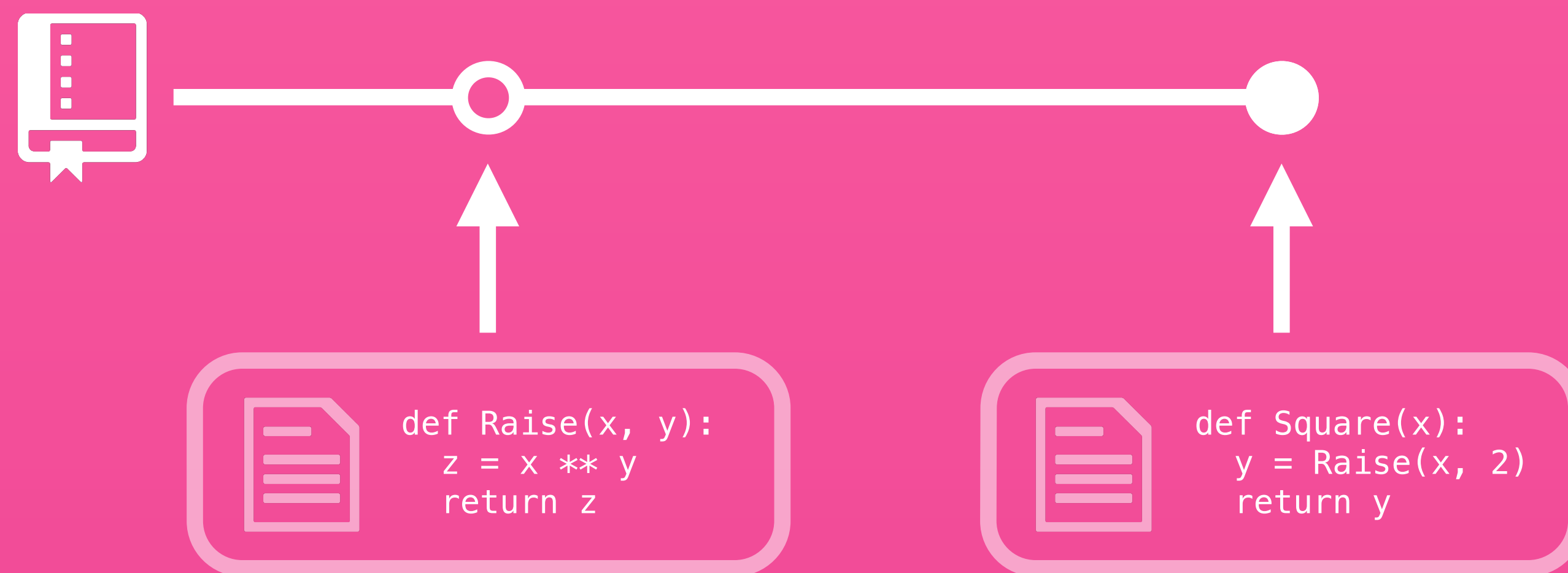
Revert a commit



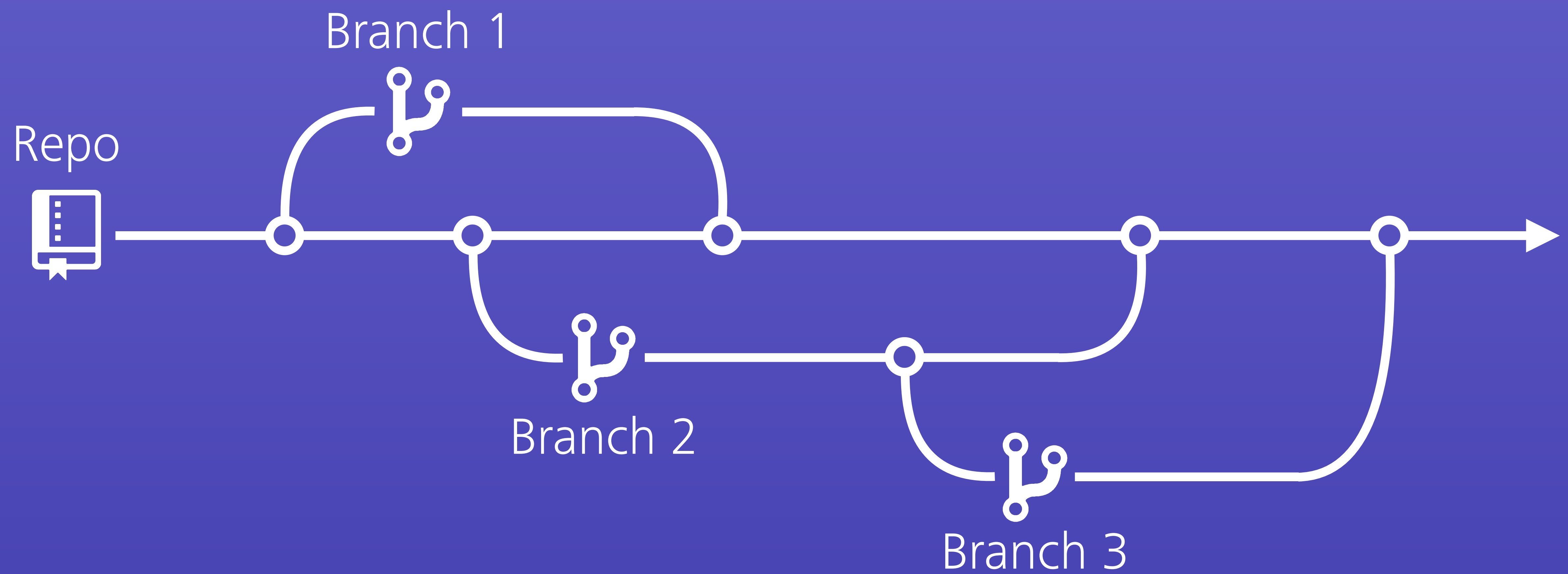
Roll back to a previous commit



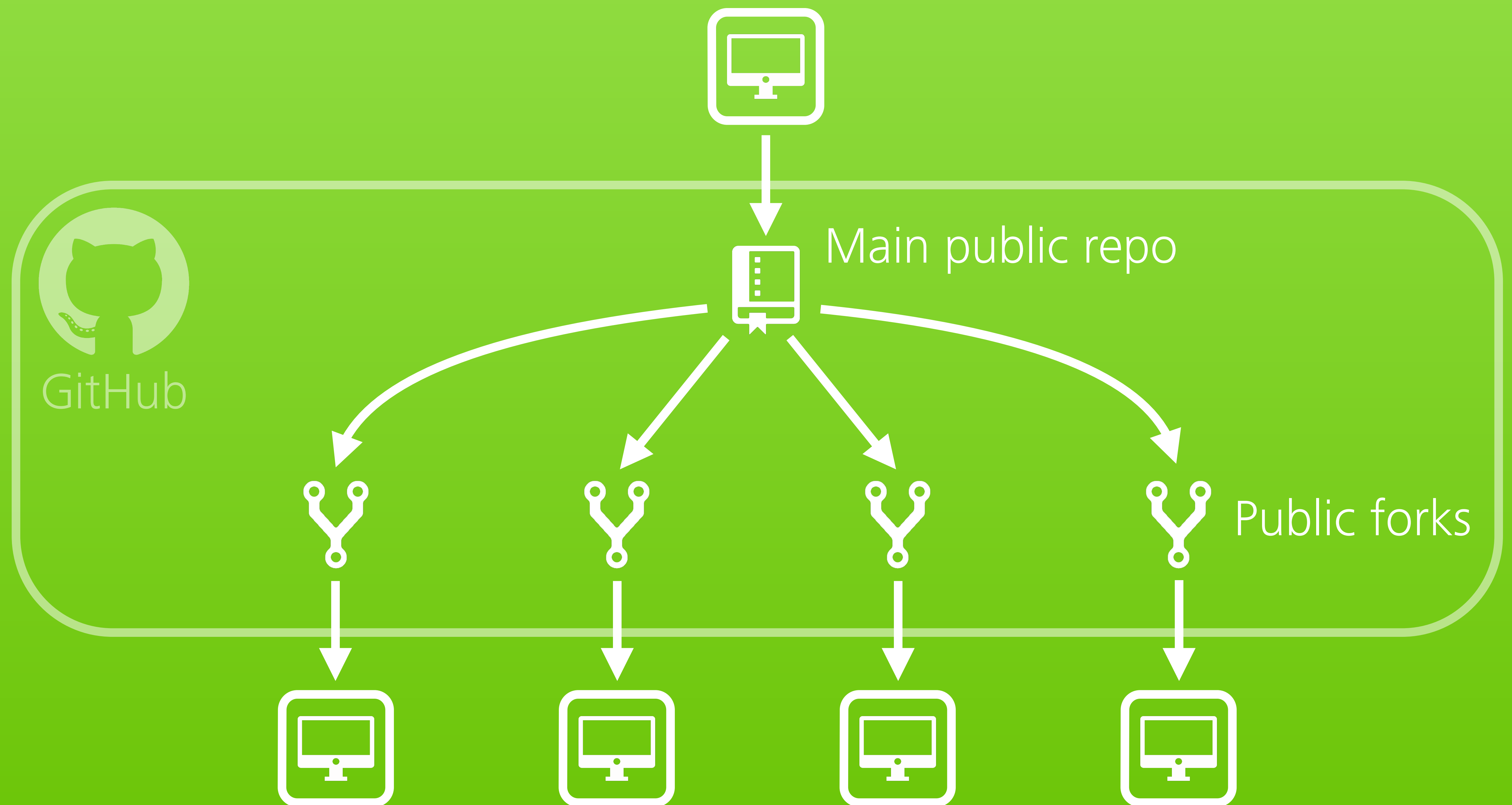
Roll back to a previous commit



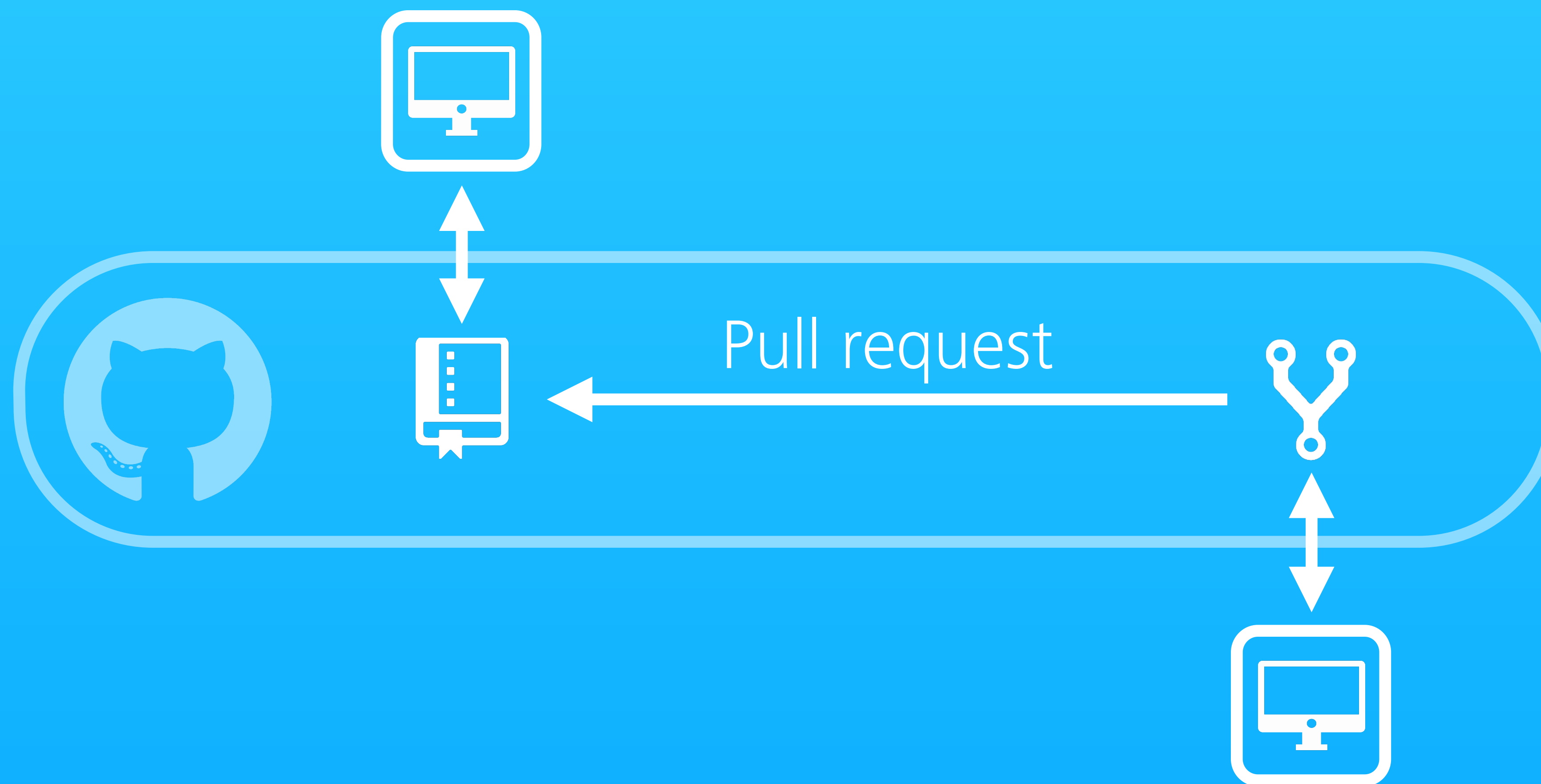
Branch



Fork



Pull request



Two handy apps



GitHub



Atom



GitHub

+

master ▾

ChangesHistoryBranchesSettings

Sync

Filter Repositories

GitHub

Alpha-only

Copycats

Infinity

MantelTest

MCMC-Cipher-Solver

MonteCarloVowels

PageTrendTest

RomanNumeralGenerator

SimpleSignUp

svg-polygons

Other

Procrustes

401 commits

Updated .gitignore

1 day ago by jwcarr

Fixed bug in variable name in getExclu...

28 days ago by jwcarr

Manually pass owner as username for...

29 days ago by jwcarr

Return empty array if no exclusions ha...

29 days ago by jwcarr

Unset experiment object before deletin...

29 days ago by jwcarr

Manually pass owner to initialize Experi...

29 days ago by jwcarr

Provide a re-signup link when del...

29 days ago by jwcarr

Manually pass owner to initialize...

29 days ago by jwcarr

Added change_details form valid...

1 month ago by jwcarr

Fixed bug in variable name in getExclusions()

Tried to get exclusions from a variable local to the function rather than the class, resulting in the excluded experiments being overwritten with an empty array.

jwcarr 5d17f5c 28 days ago ⚙ ▾

php/class.experiment.php

...	...	@@ -112,7 +112,7 @@ class Experiment {
112	112	public function getExclusions() {
113	113	if (isset(\$this->exclusions) == False) {
114	114	\$this->exclusions = explode('; ', \$this->extractElement('exclusions', \$this->file->data));
115	-	if (\$exclusions[0] == '') { \$this->exclusions = array(); }
	115	+ if (\$this->exclusions[0] == '') { \$this->exclusions = array(); }
116	116	}
117	117	return \$this->exclusions;
118	118	}



Atom

svg_polygons.py - /Users/jon/Sites/Infinity - Atom

Infinity

> Data

> Experiment

> images

> Visualizer

.gitignore

analysis.py

geometry.py

initial_set_generator.py

language_generator.py

MantelTest.py

page.py

pointedness.py

random_assign_to_chain.p

README.md

svg_polygons.py

vocalize.py

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

```
#!/usr/bin/env python

class Canvas:

    width = 0
    height = 0
    canvas = ''
    shape_count = 0

    def __init__(self, width=500, height=500):
        self.width = width
        self.height = height

    def polygon(self, shape, border_colour='black', fill_colour=None, opacity=1.0):
        canvas = "\n <g id='shape%s'>" % self.shape_count
        points = [(str(vertex[0]) + "," + str(vertex[1])) for vertex in shape]
        canvas += "\n    <polyon points='" + (" ".join(points)) + "' style='fill:%s; stroke:%s; fill-opacity:%s; stroke-opacity:%s; stroke-wid"
        canvas += "\n    </g>\n"
        self.canvas += canvas
        self.shape_count += 1

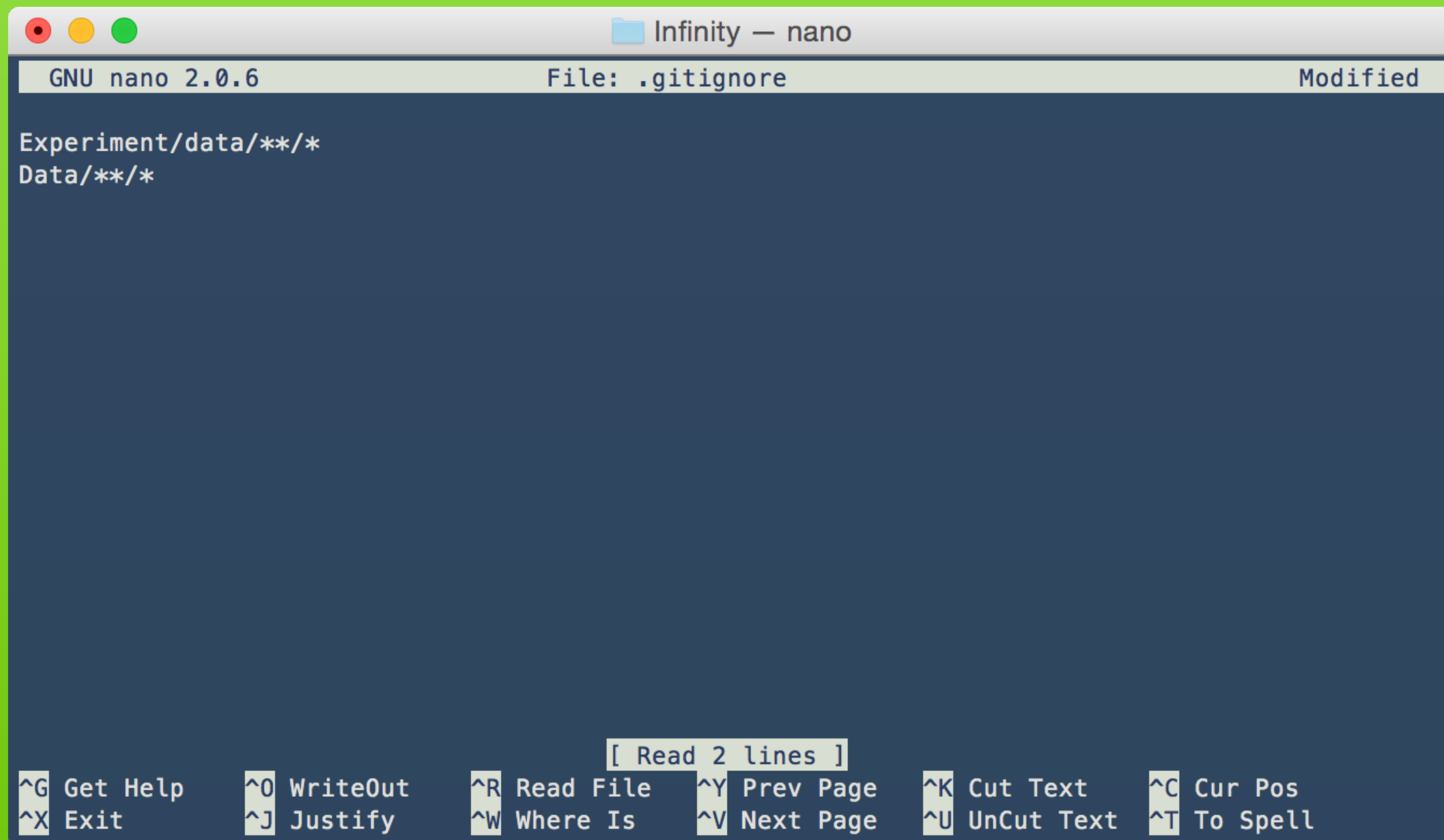
    def circle(self, position, radius=1, border_colour='black', fill_colour=None, opacity=1.0):
        canvas = "\n <g id='shape%s'>" % self.shape_count
        canvas += "\n    <circle cx='%s' cy='%s' r='%s' style='stroke:%s; fill:%s; fill-opacity:%s; stroke-opacity:%s;' />" % (position[0], pos
        canvas += "\n    </g>\n"
        self.canvas += canvas
        self.shape_count += 1

    def save(self, filename='drawing'):
        canvas = self.addHeader()
        canvas += self.addCanvas()
        canvas += self.addFooter()
        f = open(filename + '.svg', 'w')
        f.write(canvas)
        f.close()
```

svg_polygons.py 1,1

UTF-8 Python master +50

.gitignore



The screenshot shows a macOS-style window titled "Infinity — nano". The window contains a nano 2.0.6 text editor editing a file named ".gitignore". The editor's status bar at the top indicates "GNU nano 2.0.6", "File: .gitignore", and "Modified". The main editing area has a dark blue background and contains two lines of text: "Experiment/data/**/*" and "Data/**/*". At the bottom of the window, a status bar displays "[Read 2 lines]" and a series of keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, and ^T To Spell.

```
GNU nano 2.0.6 File: .gitignore Modified

Experiment/data/**/*
Data/**/*

[ Read 2 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

(GitHub Flavored) Markdown

The screenshot shows the Atom text editor interface. The left sidebar displays a file tree for a project named 'svg-polygons', containing 'example_drawing.svg', 'README.md', and 'svg_polygons.py'. The main editor area is split into two panes. The left pane shows the 'README.md' file with line numbers 1 through 31. The right pane shows a preview of the README.md file. The README content includes a title, a description, a usage section with code blocks for importing the module and creating a canvas, and a list of coordinates for two triangles.

README.md

```
1 |svg-polygons
2 |=====
3 |
4 | A Python class for drawing polygons and saving as an SVG file
5 |
6 |
7 | Usage
8 | -----
9 |
10 | First import the module:
11 |
12 | ```python
13 | import svg_polygons
14 | ```
15 |
16 | Let's say you want to draw two triangles. These should be
17 | represented as a list of three tuples. Each tuple gives the x
18 | and y coordinates for a vertex of the triangle.
19 |
20 | ```python
21 | triangle1 = [(100, 70), (325, 210), (60, 300)]
22 | triangle2 = [(455, 346), (39, 231), (80, 312)]
23 | ```
24 |
25 | Now create a Canvas object specifying its width and height (in
26 | this case the canvas is 500x500):
27 |
28 | ```python
29 | my_drawing = svg_polygons.Canvas(500, 500)
30 | ```
31 |
32 | Now you can draw your triangles to the canvas, optionally
33 | specifying a border colour, fill colour, and opacity level:
34 |
35 | ```python
```

README.md Preview

svg-polygons

A Python class for drawing polygons and saving as an SVG file

Usage

First import the module:

```
import svg_polygons
```

Let's say you want to draw two triangles. These should be represented as a list of three tuples. Each tuple gives the x and y coordinates for a vertex of the triangle.

```
triangle1 = [(100, 70), (325, 210), (60, 300)]
triangle2 = [(455, 346), (39, 231), (80, 312)]
```

Now create a Canvas object specifying its width and height (in this case the canvas is 500x500):

```
my_drawing = svg_polygons.Canvas(500, 500)
```

Now you can draw your triangles to the canvas, optionally specifying a border colour, fill colour, and opacity level:

```
```python
```

README.md 1,1 UTF-8 GitHub Markdown ? master

# Other GitHub features

**Issues:** Bug reporting and feature requests

**Wiki:** Document the project

**Pages:** Free webpage for your project

**Gist:** A mini repo for snippets of text

*Demo...*

# Collaborative project

Aim: Build a simple Mantel Test module in Python

Three teams will each implement part of the code

Then we'll test our code at the end

Team 1

`ReadFile()`

Team 2

`PairwiseDistances()`

Team 3

`MonteCarlo()`

# Quiz

1. What's the name of the GitHub mascot?
2. What's the difference between a fork and a branch?
3. What's a pull request?
4. Can you delete a previous commit?
5. What would you use a .gitignore file for?
6. What could you use GitHub Issues for?

