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
Today’s workshop

Background
Specifics
Demonstration
Collaborative project
Quiz
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
def Hello(name):
greeting = "Hello " + name
print greeting
```python
1 def Hello(name):
2     greeting = "Hello " + name
3     print greeting
4
def Bye(name):
5     valediction = "Bye bye " + name
6     print valediction
```
def Raise(x, y):
    z = x ** y
    return z

def Square(x):
    y = Raise(x, 2)
    return y

def Cube(x):
    y = Raise(x, 3)
    return y

Revert a commit
def Raise(x, y):
    z = x ** y
    return z

def Square(x):
    y = Raise(x, 2)
    return y

def Cube(x):
    y = Raise(x, 3)
    return y

Revert a commit
def Raise(x, y):
    z = x ** y
    return z

def Square(x):
    y = Raise(x, 2)
    return y

def Cube(x):
    y = Raise(x, 3)
    return y

Roll back to a previous commit
def Raise(x, y):
    z = x ** y
    return z

def Square(x):
    y = Raise(x, 2)
    return y
Branch

Repo

Branch 1

Branch 2

Branch 3
Pull request
Two handy apps

GitHub

Atom
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.

```php
php/class.experiment.php

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

```python
#!/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='shapes'>" + self.shape_count
        points = [(str(vertex[0]) + "", str(vertex[1])) for vertex in shape]
        canvas += "\n        <polygon points=" + ("\n            \n        <".join(points)) + "" style='fill:%s; stroke:%s; fill-opacity:%s; stroke-opacity:%s; stroke-width:1' />
        \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='shapes'>" + self.shape_count
        canvas += "\n            <circle cx='" + str(position[0]) + 'cy=' + str(position[1]) + 'r=' + str(radius) + ' style="fill:%s; stroke:%s; fill-opacity:%s; stroke-opacity:%s;" />
        \n        self.canvas += canvas
        self.shape_count += 1

    def save(self, filename='drawing'):
        canvas = self.header()
        canvas += self.addCanvas()
        canvas += self.addFooter()
        f = open(filename + '.svg', 'w')
        f.write(canvas)
        f.close()
```
Experiment/data/**/*
Data/**/*
svg-polygons

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

Usage

First import the module:
```
```python
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.
```
```python
triangle1 = [(100, 70), (325, 210), (60, 300)]
triangle2 = [(455, 346), (39, 231), (86, 312)]
```

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

<table>
<thead>
<tr>
<th>Team 1</th>
<th>Team 2</th>
<th>Team 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadFile()</td>
<td>PairwiseDistances()</td>
<td>MonteCarlo()</td>
</tr>
</tbody>
</table>
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?
Homework
(if you’re keen)

Merge the functions created by the other two teams into your fork of the repository

Test out the final code for yourself!