````
# FunkyYak (Functional Kayak)

Taking a few lessons from developing and working with Kayak, here is a
stateless reverse-mode autodiff implementation that also offers
higher-order derivatives.

Example use:

```python
import numpy as np
import matplotlib.pyplot as plt
from funkyyak import grad

# Define a function capable of taking `Node` objects
def fun(x):
    return np.sin(x)

d_fun = grad(fun)    # First derivative
dd_fun = grad(d_fun) # Second derivative

x = np.linspace(-10, 10, 100)
plt.plot(x, map(fun, x), x, map(d_fun, x), x, map(dd_fun, x))
```
<img src="https://github.com/HIPS/FunkyYak/blob/master/examples/sinusoid.png"
width="600">
The function can even have control flow, which raises the prospect
of differentiating through an iterative routine like an
optimization. Here's a simple example.

```python
# Taylor approximation to sin function
def fun(x):
    currterm = x
    ans = currterm
````
````

```
        for i in xrange(1000):
            currterm = - currterm * x ** 2 / ((2 * i + 3) * (2 * i + 2))
            ans = ans + currterm
            if np.abs(currterm) < 0.2: break # (Very generous tolerance!)

        return ans

    d_fun = grad(fun)
    dd_fun = grad(d_fun)

    x = np.linspace(-10, 10, 100)
    plt.plot(x, map(fun, x), x, map(d_fun, x), x, map(dd_fun, x))
    ```

    <img
    src="https://github.com/HIPS/FunkyYak/blob/master/examples/sinusoid_taylor.png"
    width="600">
    We can take the derivative of the derivative automatically as well, as many
    times as we like:
    ```python
    # Define the tanh function
    def tanh(x):
        return (1.0 - np.exp(-x))  / ( 1.0 + np.exp(-x))

    d_fun = grad(tanh)           # First derivative
    dd_fun = grad(d_fun)         # Second derivative
    ddd_fun = grad(dd_fun)       # Third derivative
    dddd_fun = grad(ddd_fun)     # Fourth derivative
    ddddd_fun = grad(dddd_fun)   # Fifth derivative
    dddddd_fun = grad(ddddd_fun) # Sixth derivative

    x = np.linspace(-7, 7, 200)
    plt.plot(x, map(tanh, x),
             x, map(d_fun, x),
             x, map(dd_fun, x),
             x, map(ddd_fun, x),
             x, map(dddd_fun, x),
             x, map(ddddd_fun, x),
             x, map(dddddd_fun, x))
    ```

    <img src="https://github.com/HIPS/FunkyYak/blob/master/examples/tanh.png"
```
```