

```

1
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from funkyyak import grad
5
6 # Define a function capable of taking `Node` objects
7 def fun(x):
8     return np.sin(x)
9
10 d_fun = grad(fun)    # First derivative
11 dd_fun = grad(d_fun) # Second derivative
12
13 x = np.linspace(-10, 10, 100)
14 plt.plot(x, map(fun, x), x, map(d_fun, x), x, map(dd_fun, x))
15
16 plt.xlim([-10, 10])
17 plt.ylim([-1.2, 1.2])
18 plt.axis('off')
19 plt.savefig("sinusoid.png")
20 plt.clf()
21
22 # Taylor approximation to sin function
23 ✓ def fun(x):
24     currterm = x
25     ans = currterm
26     for i in xrange(1000):
27         currterm = - currterm * x ** 2 / ((2 * i + 3) * (2 * i + 2))
28         ans = ans + currterm
29         if np.abs(currterm) < 0.2: break # (Very generous tolerance!)
30
31     return ans
32
33 d_fun = grad(fun)

```

```
34     dd_fun = grad(d_fun)
35
36     x = np.linspace(-10, 10, 100)
37     plt.plot(x, map(fun, x), x, map(d_fun, x), x, map(dd_fun, x))
38
39     plt.xlim([-10, 10])
40     plt.ylim([-1.2, 1.2])
41     plt.axis('off')
42     plt.savefig("sinusoid_taylor.png")
43     plt.clf()
```
