```
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))

plt.xlim([-10, 10])
plt.ylim([-1.2, 1.2])
plt.axis('off')
plt.savefig("sinusoid.png")
plt.clf()

# 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))

    plt.xlim([-10, 10])
    plt.ylim([-1.2, 1.2])
    plt.axis('off')
    plt.savefig("sinusoid_taylor.png")
    plt.clf()
```