Day 23?

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
import seaborn as sns
sns.set_theme()
```

In [3]:

```
# Euler's Method

def f(x,t):
    return(-x**3 + np.sin(t))

# define boundary conditions

a = 0.0 # starting point
b = 3.0 # ending point
N = 100 # number of points between a and b
dt = (b-a)/N

x = 0.0 # initial condition


tpointsE = np.arange(a, b, dt)
xpointsE = []

for t in tpointsE:
    xpointsE.append(x)
    x = x + dt*f(x,t)
```

In [4]:

```
# Runge-Kutta 2nd order

def f(x,t):
    return(-x**3 + np.sin(t))

# define boundary conditions

a = 0.0 # starting point
b = 3.0 # ending point
N = 100 # number of points between a and b
dt = (b-a)/N

x = 0.0 # initial condition


tpointsRK2 = np.arange(a, b, dt)
xpointsRK2 = []

for t in tpointsRK2:
    xpointsRK2.append(x)
    k1 = dt*f(x,t)
    k2 = dt*f(x+0.5*k1,t+0.5*dt)
    x = x + k2
```

In [5]:

```
# Runge-Kutta 4nd order

def f(x,t):
    return(-x**3 + np.sin(t))

# define boundary conditions

a = 0.0 # starting point
b = 3.0 # ending point
N = 100 # number of points between a and b
dt = (b-a)/N

x = 0.0 # initial condition


tpointsRK4 = np.arange(a, b, dt)
xpointsRK4 = []

for t in tpointsRK4:
    xpointsRK4.append(x)
    k1 = dt*f(x,t)
    k2 = dt*f(x+0.5*k1,t+0.5*dt)
    k3 = dt*f(x+0.5*k2,t+0.5*dt)
    k4 = dt*f(x+k3, t+dt)
    x = x + (k1+2*k2+2*k3+k4)/6
```

In [6]:

```
fig0,ax0 = plt.subplots()

ax0.plot(tpointsE, xpointsE, 'o')
ax0.plot(tpointsRK2, xpointsRK2, 'o')
ax0.plot(tpointsRK4, xpointsRK4, 'o')
```

Out[6]: [<matplotlib.lines.Line2D at 0x7f6001b31240>]

In [ ]:

```
```