Example 1 :

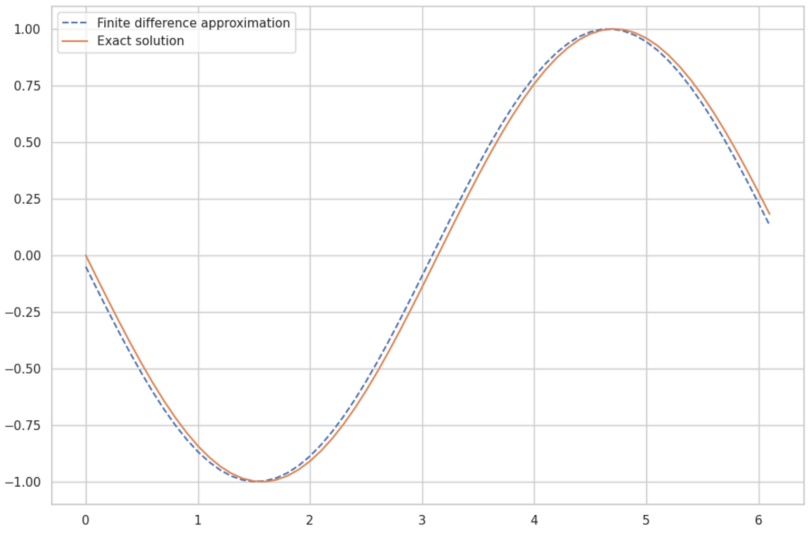
```
[9] import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    sns.set_theme(style="whitegrid")
```

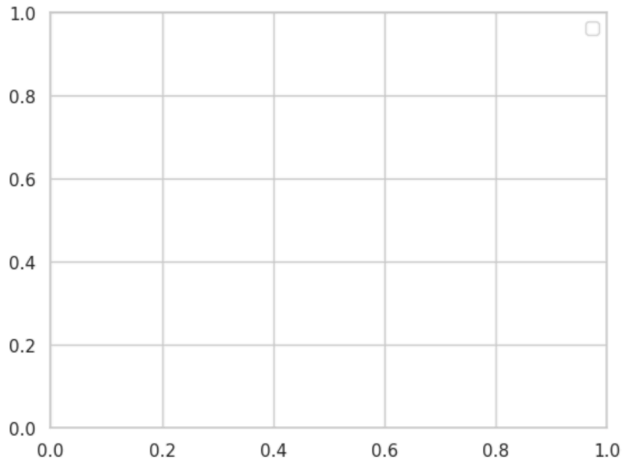
```
h = 0.1
x = np.arange(0, 2*np.pi, h)
y = np.cos(x)
forward_diff = np.diff(y)/h
x_diff = x[:-1:]
exact_solution = -np.sin(x_diff)
h = 0.1
x = np.arange(0, 2*np.pi, h)
y = np.cos(x)
forward_diff = np.diff(y)/h
x_diff = x[:-1]
exact_solution = -np.sin(x_diff)

plt.figure(figsize = (12, 8))
plt.plot(x_diff , forward_diff , '--', \
         label = 'Finite difference approximation')
plt.plot(x_diff , exact_solution , \
         label = 'Exact solution')
plt.legend()
plt.show()

max_error = max(abs(exact_solution - forward_diff))
print(max_error)
plt.legend()
plt.show()

max_error = max(abs(exact_solution - forward_diff))
print(max_error)
```





Example 3 :

```python
import numpy as np
x0 = 0.7
h = 2.**-np.arange(1, 30)
df = (np.cos(x0 + h) - np.cos(x0)) / h

true_value = -np.sin(x0)

print("k | Approximation        | Ratio of Errors      | Relative Difference")
print("---|----------------------|----------------------|-------------------------")

previous_approximation = None
previous_error = None

for k in range(1, len(h) + 1):
    approximation = df[k - 1]
    error = np.abs(approximation - true_value)
    ratio = np.abs(previous_error / error) if previous_error is not None else "N/A"
    relative_difference = np.abs((approximation - previous_approximation) / previous_approximation) if previous_approximation is not None else "N/A"

    formatted_approximation = f"{approximation:.15f}"
import numpy as np
x0 = 0.7
h = 2.**-np.arange(1, 30)
df = (np.cos(x0 + h) - np.cos(x0)) / h

true_value = -np.sin(x0)

print("k | Approximation        | Ratio of Errors      | Relative Difference")
print("---|----------------------|----------------------|-------------------------")

previous_approximation = None
previous_error = None

for k in range(1, len(h) + 1):
    approximation = df[k - 1]
    error = np.abs(approximation - true_value)
    ratio = np.abs(previous_error / error) if previous_error is not None else "N/A"
    relative_difference = np.abs((approximation - previous_approximation) / previous_approximation) if previous_approximation is not None else "N/A"

    formatted_approximation = f"{approximation:.15f}"
    formatted_ratio = f"{ratio:.6f}" if isinstance(ratio, float) else ratio
    formatted_relative_difference = f"{relative_difference:.10f}" if isinstance(relative_difference, float) else relative_difference
    print(f"{k:<3}| {formatted_approximation:<21}| {formatted_ratio:<17}| {formatted_relative_difference:<20}")
    previous_error = error
    previous_approximation = approximation

print(f"\nTrue value of the derivative: {true_value:.17f}")
```

```
k | Approximation        | Ratio of Errors      | Relative Difference
---|----------------------|----------------------|-------------------------
k | Approximation        | Ratio of Errors      | Relative Difference
---|----------------------|----------------------|-------------------------
1   | -0.804968865615630   | N/A                  | N/A
2   | -0.732636391282420   | 1.818068             | 0.0898574807
3   | -0.690281773285188   | 1.919472             | 0.0578112397
4   | -0.667691895578365   | 1.962328             | 0.0327255891
5   | -0.656062525879001   | 1.981809             | 0.0174172695
6   | -0.650166682293637   | 1.991065             | 0.0089867099
7   | -0.647198783534563   | 1.995573             | 0.0045648275
8   | -0.645709879406184   | 1.997797             | 0.0023005360
9   | -0.644964193616318   | 1.998901             | 0.0011548310
10  | -0.644591042911657   | 1.999451             | 0.0005785603
11  | -0.644404390684713   | 1.999726             | 0.0002895669
12  | -0.644311045362429   | 1.999863             | 0.0001448552
13  | -0.644264367900178   | 1.999931             | 0.0000724455
14  | -0.644241027968746   | 1.999966             | 0.0000362273
15  | -0.644229357705626   | 1.999982             | 0.0000181147
16  | -0.644223522496759   | 1.999991             | 0.0000090577
17  | -0.644220604881411   | 1.999990             | 0.0000045289
18  | -0.644219146081014   | 1.999971             | 0.0000022644
19  | -0.644218416651711   | 2.000021             | 0.0000011323
20  | -0.644218052038923   | 1.999483             | 0.0000005660
21  | -0.644217869732529   | 1.998967             | 0.0000002830
22  | -0.644217778462917   | 2.000487             | 0.0000001417
23  | -0.644217733293772   | 1.980742             | 0.0000000701
24  | -0.644217710942030   | 1.942939             | 0.0000000347
25  | -0.644217699766159   | 1.892038             | 0.0000000173
26  | -0.644217692315578   | 2.467260             | 0.0000000116
27  | -0.644217699766159   | 0.405308             | 0.0000000116
28  | -0.644217699766159   | 1.000000             | 0.0000000000
29  | -0.644217729568481   | 0.295966             | 0.0000000463

True value of the derivative: -0.64421768723769102
```