

Grafer

Programmering 2

vt 24

Outline

Grafer

Repetition

Nyheter

Uppdaterande graf

Uppdatera värden

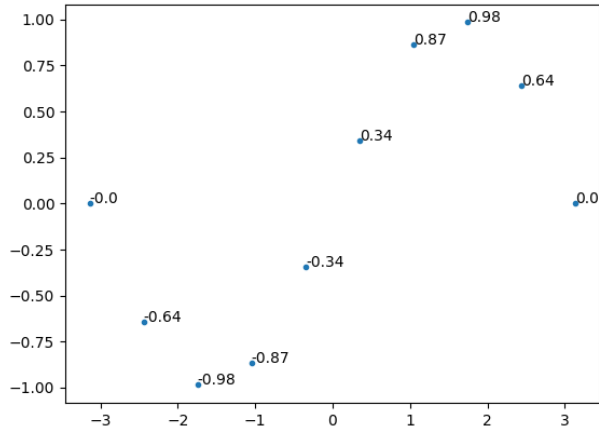
Lägg till nya värden

Övningar

Repetition

```
1 from matplotlib import pyplot as plt
2 import numpy as np
3
4 x_data = np.linspace(-np.pi, np.pi, 10) # Skapar tio tal mellan
      -pi och pi
5 y_data = [np.sin(x) for x in x_data] # En sinus-kurva
6
7 plt.plot(x_data, y_data, '.')
```

Annoteringar



Annoteringar

```
1 from matplotlib import pyplot as plt
2 import numpy as np
3
4 x_data = np.linspace(-np.pi, np.pi, 10) # Skapar tio tal mellan
      -pi och pi
5 y_data = [np.sin(x) for x in x_data] # En sinus-kurva
6
7 plt.plot(x_data, y_data, '.')
```

8

```
9 for i in range(len(y_data)):
10     # Labels
11     plt.annotate(round(y_data,2), (x_data[i], y_data[i]))
12
13 plt.show()
```

Outline

Grafer

Repetition

Nyheter

Uppdaterande graf

Uppdatera värden

Lägg till nya värden

Övningar

Uppdaterande graf

- ▶ Det finns ett par olika sätt att uppdatera en graf
 - ▶ Uppdatera redan existerande värden
 - ▶ Lägg till nya värden
- ▶ Båda kräver samma funktion `plt.ion()`
- ▶ Vi behöver också använda fler objekt

- För att ha en interaktiv graf behöver vi ha mer kontroll över grafen
- Det får vi genom att använda figurer

```
1 from matplotlib import pyplot as plt
2 import numpy as np
3
4 x_data = np.linspace(-np.pi, np.pi, 100)
5 y_data = [np.sin(x) for x in x_data]
6
7 fig = plt.figure() # Skapar en figur (typ fönstret)
8 ax = fig.add_subplot(111) # Lägger till ett ritområde
9 ax.plot(x_data, y_data) # Rita grafen
10 fig.show() # Visa fönstret med grafen
```


Uppdatera värden

```
1 fig = plt.figure() # Skapar en figur (typ fönstret)
2 ax = fig.add_subplot(111) # Lägger till ett ritområde
3 ax.plot(x_data, y_data) # Rita grafen
4 # Skapa ny data
5 ny_y = np.array([np.cos(x) for x in x_data])
6 line = ax.get_lines()[0] # Hämta första ritade linjen
7 line.set_ydata(ny_y)
8 fig.show()
```

Uppdaterande graf

Se förändringarna

```
1 plt.ion()
2 fig = plt.figure() # Skapar en figur (typ fönstret)
3 ax = fig.add_subplot(111) # Lägger till ett ritområde
4 ax.plot(x_data, y_data) # Rita grafen
5 line = ax.get_lines()[0]
6 fig.show() # Visa fönstret med grafen
7 t = 0
8 while True:
9     t += np.pi/10 # Förskjutningen
10    line.set_ydata(np.array([np.sin(x+t) for x in x_data]))
11    fig.canvas.draw() # Ritar ut
```

Graf med nya värden

Se förändringarna

```
1 plt.ion()
2 fig = plt.figure() # Skapar en figur (typ fönstret)
3 ax = fig.add_subplot(111) # Lägger till ett ritområde
4 ax.plot(x_data, y_data) # Rita grafen
5 line = ax.get_lines()[0]
6 fig.show() # Visa fönstret med grafen
7 t = np.pi
8 while i < 100:
9     t += np.pi/10
10    x_data = np.append(x_data, t) # Arrayer är besvärliga
11    line.set_xdata(x_data) # Uppdatera x_data, nästa rad y_data
12    line.set_ydata(np.array([np.sin(x) for x in x_data]))
13    ax.set_autoscaley_on(True) # Behövs inte för exemplet
14    ax.set_xlim(min(x_data), max(x_data))
15    fig.canvas.draw()
```

Outline

Grafer

- Repetition

- Nyheter

Uppdaterande graf

- Uppdatera värden

- Lägg till nya värden

Övningar

Övningar

1. Återskapa graferna från övningarna
2. Annotera alla graferna
3. Prata med Calle för fler övningar