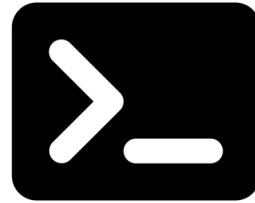
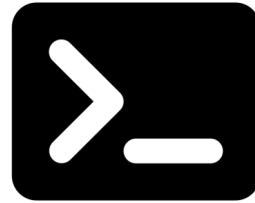


**Unser Python-Programm liest die  
Temperaturdaten von  
einem Temperatursensor aus und  
visualisiert diese in Echtzeit in  
einem Diagramm.**



```
cat /sys/bus/w1/devices/28-*/temperature
```



```
git clone https://github.com/oaxiento/girlsday.git
```

# Definiere wo das Programm die Sensordaten im Dateisystem findet

---

```
temperatur_sensor_ordner =  
glob.glob('/sys/bus/w1/devices/' + '28*')[0]  
temperatur_sensor_datei = temperatur_sensor_ordner  
+ '/temperature'
```

---

# Lese die Datei, die die Sensordaten enthält

---

```
sensor_datei = open(temperatur_sensor_datei, 'r')  
sensor_daten = sensor_datei.readlines()  
sensor_datei.close()
```

---

# **Lese den aktuellen Sensorwert aus den Sensordaten**

---

```
sensorwert = sensor_daten[0]
```

---

# Wandle den Sensorwert in Grad Celsius um

---

```
temperatur_celsius = float(sensorwert) / 1000.0
```

---

**Füge die aktuelle Zeit zur x-Achse und den  
Temperaturwert in Grad Celsius zur y-Achse  
des Diagramms hinzu**

---

```
xs.append(dt.datetime.now().strftime('%H:%M:%S'))  
ys.append(temperatur_celsius)
```

---



# Kürze die Werte des Diagrams auf maximal 10 Elemente

---

```
xs = xs[-10:]  
ys = ys[-10:]
```

---

# Schreibe die Werte auf die Achsen

---

```
ax.clear()  
ax.plot(xs, ys)
```

---

# Füge Titel und Achsenbeschriftungen zum Diagramm hinzu

---

```
plt.title('Temperaturdiagramm')  
plt.ylabel('Temperatur (C°)')  
plt.xticks(rotation=90, ha='center', fontsize=10)  
plt.subplots_adjust(bottom=0.20)
```

---

# Lege fest wie das Diagramm aussehen soll und wie groß das Programmfenster sein soll

---

```
fig = plt.figure(facecolor='white', figsize=(10,  
5))  
ax = fig.add_subplot(1, 1, 1)  
xs = []  
ys = []
```

---

**Erstelle eine Animation, die das formatierte Diagramm und die Temperaturdaten jede Sekunde (alle 1000 Millisekunden) aktualisiert**

---

```
ani = animation.FuncAnimation(fig, diagramm,  
fargs=(xs, ys), interval=1000)
```

---

**Öffne ein Fenster mit dem Diagramm und zeige es an bis es geschlossen wird**

---

```
plt.show()
```

---