



Beispielcode: Astronomische Berechnungen mit Skyfield

Python-Code für die Berechnung der Position von Merkur und Erdnähe

Der folgende Python-Code zeigt, wie man mit der Bibliothek `skyfield` die Position von Merkur relativ zur Erde berechnen und die Erdnähe über einen Zeitraum von 15 Jahren ermitteln kann:

```
from skyfield.api import load, wgs84, utc
from datetime import datetime, timedelta
from zoneinfo import ZoneInfo # Python 3.9+

# Beobachterstandort: südlich von Augsburg
latitude = 48.04
longitude = 10.886
elevation = 597 # Meter

# Skyfield-Zeit & Planeten laden
ts = load.timescale()
t_now = ts.now()
planets = load('de440.bsp') # ⚠ vollständiges Ephemeriden-File
earth, mercury = planets['earth'], planets[1]
observer = earth + wgs84.latlon(latitude, longitude, elevation_m=elevation)

# ==== Aktuelle Uhrzeit ====
utc_now = datetime.now(ZoneInfo("UTC"))
local_now = utc_now.astimezone(ZoneInfo("Europe/Berlin"))

# ==== Aktuelle Merkur-Position ====
d_now = earth.at(t_now).observe(mercury).apparent().distance()
astrometric_now = observer.at(t_now).observe(mercury).apparent()
alt_now, az_now, dist_now = astrometric_now.altaz()

# 📄 CSV-Ausgabe der aktuellen Werte
print(f"{alt_now.degrees:.2f},{az_now.degrees:.2f},{dist_now.km:.2f},{latitude:.2f},{longitude:.3f},{elevation:.2f}")

# Trenner für PHP
print("==ERDNEAECHSTE==")

# ==== Zukünftige Erdnähen über 15 Jahre ====
start = datetime.now(utc)
end = start + timedelta(days=15 * 365)
current = start
step = timedelta(days=1)

distances = []
while current <= end:
    t = ts.utc(current)
    d = earth.at(t).observe(mercury).apparent().distance().au
    distances.append((current, d))
    current += step

# Lokale Minima (Erdnächste)
closest_points = []
for i in range(1, len(distances) - 1):
    if distances[i][1] < distances[i - 1][1] and distances[i][1] < distances[i + 1][1]:
        closest_points.append(distances[i])

# Merkur-Äquatordurchmesser in km
MERCURY_DIAMETER_KM = 4879

# CSV-Ausgabe der Erdnächsten
for date, dist_au in closest_points:
    t = ts.utc(date)
    astrometric = observer.at(t).observe(mercury).apparent()
    alt, az, _ = astrometric.altaz()
    dist_km = dist_au * 149597870.7
    angular_diameter_arcsec = (MERCURY_DIAMETER_KM / dist_km) * 206265

# Lokale Zeit inkl. Sommerzeit
local_dt = date.replace(tzinfo=ZoneInfo("UTC")).astimezone(ZoneInfo("Europe/Berlin"))
local_str = local_dt.strftime('%Y-%m-%d %H:%M')

# 📄 CSV-Zeile
print(f"{local_str},{dist_au:.4f},{alt.degrees:.1f},{az.degrees:.1f},{angular_diameter_arcsec:.2f}")
```

Hinweise:

- Die Bibliothek `skyfield` wird für astronomische Berechnungen genutzt, insbesondere für die Berechnung von Planetenpositionen und Entfernungen.
- Der Code zeigt, wie man die Position von Merkur aus der Perspektive eines Beobachters in Augsburg berechnet.
- Es werden auch die Erdnähe von Merkur sowie der scheinbare Durchmesser auf Grundlage der aktuellen Entfernung ermittelt.
- Für eine vollständige Berechnung müssen die Ephemeriden-Daten geladen werden (dazu die Datei `de440.bsp`).