

Python

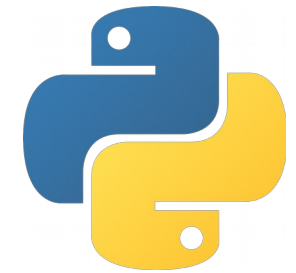
Datum und Zeit



- Elementare Zeitfunktionen – time
 - Das time-Modul setzt direkt auf den Zeitfunktionen der C-Bibliothek des Betriebssystems auf
 - Es speichert deshalb alle Zeitangaben als sogenannte Unix-Timestamps
 - Ein Unix-Timestamp beschreibt einen Zeitpunkt durch die Anzahl der Sekunden, die seit dem 01.01.1970 um 00:00 Uhr vergangen sind

Python

Datum und Zeit

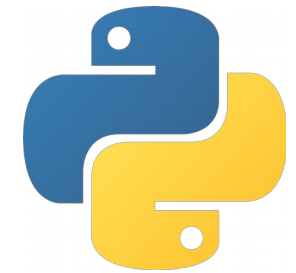


- Elementare Zeitfunktionen – time
 - Darstellung von Zeiten durch den Datentyp `struct_time`
 - Die Instanzen des Typs `struct_time` haben neun Attribute
 - wahlweise über einen Index
 - Oder ihren Namen angesprochen werden

Index	Attributname	Bedeutung	Wertebereich
0	<code>tm_year</code>	Die Jahreszahl des Zeitstempels ³	1970–2038
1	<code>tm_mon</code>	Nummer des Monats	1–12
2	<code>tm_mday</code>	Nummer des Tags im Monat	1–31
3	<code>tm_hour</code>	Stunde der Uhrzeit des Zeitstempels	0–23
4	<code>tm_min</code>	Minute der Uhrzeit des Zeitstempels	0–59

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Darstellung von Zeiten durch den Datentyp `struct_time`
 - Die Instanzen des Typs `struct_time` haben neun Attribute
 - wahlweise über einen Index
 - Oder ihren Namen angesprochen werden

Index	Attributname	Bedeutung	Wertebereich
5	<code>tm_sec</code>	Sekunde der Uhrzeit des Zeitstempels ⁴	0–61
6	<code>tm_wday</code>	Nummer des Wochentages (0 entspricht Montag)	0–6
7	<code>tm_yday</code>	Nummer des Tages im Jahr	0–366
8	<code>tm_isdst</code>	Gibt an, ob der Zeitstempel durch die Sommerzeit angepasst wurde.	0 für »Nein«, 1 für »Ja«, -1 für »Unbekannt«

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Attribute vom Modul → time
 - `accept2dyear`
 - Dieses Attribut enthält einen Wahrheitswert, der angibt, ob Jahreszahlen mit nur zwei statt vier Ziffern angegeben werden dürfen, wie sie gegen Ende des 20. Jahrhunderts üblich waren
 - Hinweis
 - Die Verwendung dieses Attributs wird seit Python 3.2 nicht mehr empfohlen.
 - Anwendungen, die zweistellige Jahreszahlen benötigen, sollen sich selbst um die Umwandlung kümmern

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Attribute vom Modul → time
 - altzone
 - Dieses Attribut speichert die Verschiebung der Lokalzeit von der UTC in Sekunden, wobei eine eventuell vorhandene Sommerzeit auch berücksichtigt wird
 - Liegt die aktuelle Zeitzone östlich vom Null-Meridian, ist der Wert von altzone positiv; liegt die lokale Zeitzone westlich davon, ist er negativ
 - Dieses Attribut sollte nur dann benutzt werden, wenn daylight nicht den Wert 0 hat

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Attribute vom Modul → time
 - daylight
 - Dieses Attribut hat einen Wert, der von 0 verschieden ist, wenn es in der lokalen Zeitzone eine Sommerzeit gibt
 - Ist für den lokalen Standort keine Sommerzeit definiert, hat daylight den Wert 0
 - Die durch die Sommerzeit entstehende Verschiebung lässt sich mit altzone ermitteln

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Attribute vom Modul → time
 - struct_time
 - Dies ist eine Referenz auf den eingangs besprochenen Datentyp struct_time
 - Sie können mit struct_time direkt Instanzen dieses Typs erzeugen, indem Sie dem Konstruktor eine Sequenz mit neun Elementen übergeben
 - Beispiel

```
>>> t = time.struct_time((2011, 9, 18, 18, 24, 56, 0, 0, 0))
>>> t.tm_year
2011
```

Python

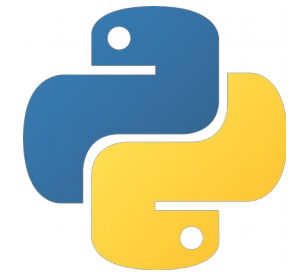
Datum und Zeit



- Elementare Zeitfunktionen – time
 - Attribute vom Modul → time
 - timezone
 - Dieses Attribut speichert die Verschiebung der Lokalzeit relativ zur UTC in Sekunden, wobei eine eventuell vorhandene Sommerzeit nicht berücksichtigt wird
 - tzname
 - Dieses Attribut enthält ein Tupel mit zwei Strings
 - Der erste String ist der Name der lokalen Zeitzone und der zweite der der lokalen Zeitzone mit Sommerzeit
 - Wenn die Lokalzeit keine Sommerzeit kennt, sollten Sie das zweite Element des Tupels nicht verwenden

Python

Datum und Zeit

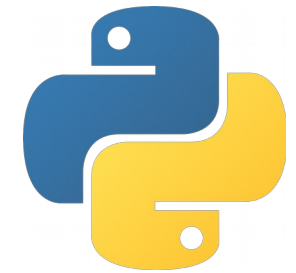


- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time

Funktion	Beschreibung
<code>asctime([t])</code>	Konvertiert die übergebene <code>struct_time</code> -Instanz in einen String.
<code>clock()</code>	Gibt die aktuelle Prozessorzeit zurück.
<code>ctime([secs])</code>	Konvertiert den übergebenen Unix-Zeitstempel in einen String. Falls kein Zeitstempel übergeben wurde, wird die aktuelle Systemzeit verwendet.
<code>gmtime([secs])</code>	Wandelt einen Unix-Timestamp in eine <code>struct_time</code> -Instanz um. Dabei wird die koordinierte Weltzeit zugrunde gelegt.
<code>localtime([secs])</code>	Wandelt einen Unix-Timestamp in eine <code>struct_time</code> -Instanz um. Dabei wird die Lokalzeit zugrunde gelegt.

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time

Funktion	Beschreibung
<code>mktime(t)</code>	Wandelt eine <code>struct_time</code> -Instanz in einen Unix-Timestamp um. Dabei wird die Lokalzeit zugrunde gelegt.
<code>sleep(secs)</code>	Unterbricht den Programmablauf.
<code>strptime(format[, t])</code>	Wandelt eine <code>struct_time</code> -Instanz nach den übergebenen Regeln in einen String um.
<code>strptime(string[, format])</code>	Interpretiert einen String nach den übergebenen Regeln als Zeitangabe und liefert eine passende <code>struct_time</code> -Instanz.
<code>time()</code>	Gibt den aktuellen Unix-Zeitstempel zurück. Dabei wird die Koordinierte Weltzeit zugrunde gelegt.

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `asctime([t])`
 - Diese Funktion wandelt eine `struct_time`-Instanz oder ein Tupel mit neun Elementen in einen String um
 - Die Funktion `asctime` liefert immer einen String zurück, der aus 24 Zeichen besteht und folgend formatiert ist
 - Beispiel

```
>>> time.asctime((1987, 7, 26, 10, 40, 0, 0, 0, 0))
'Mon Jul 26 10:40:00 1987'
```
 - Wird der optionale Parameter `t` nicht übergeben, gibt `asctime` einen String für die aktuelle Systemzeit zurück

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - clock()
 - Diese Funktion gibt die aktuelle Prozessorzeit zurück
 - Was dies konkret bedeutet, hängt von der verwendeten Plattform ab
 - Unter Unix gibt clock die Prozessorzeit zurück, die der Python-Prozess schon benutzt hat
 - Unter Windows ist es der zeitliche Abstand zum ersten Aufruf der Funktion

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - clock()
 - Wenn Sie die Laufzeit Ihrer Programme analysieren wollen, ist clock in jedem Fall die richtige Wahl
 - Beispiel

```
>>> start = time.clock()
>>> rechenintensive_funktion()
>>> ende = time.clock()
>>> print("Die Funktion lief "
...       "{0:1.2f} Sekunden".format(ende - start))
Die Funktion lief 7.46 Sekunden
```

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `ctime([secs])`
 - Diese Funktion wandelt den als Parameter übergebenen Unix-Timestamp wie `asctime` in einen String um
 - Wird der optionale Parameter nicht übergeben oder hat er den Wert `None`, wird die aktuelle Systemzeit verwendet

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `gmtime([secs])`
 - Diese Funktion wandelt einen Unix-Timestamp in ein `struct_time`-Objekt um
 - Dabei wird immer die koordinierte Weltzeit benutzt, und das `tm_isdst`-Attribut des resultierenden Objekts hat immer den Wert 0
 - Wird der Parameter `secs` nicht übergeben oder hat er den Wert `None`, wird der aktuelle Zeitstempel benutzt, wie er von der Funktion `time` zurückgegeben wird

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `gmtime([secs])`
 - Beispiel
 - `>>> time.gmtime()`
 - `time.struct_time(tm_year=2015, tm_mon=12, tm_mday=7, tm_hour=9, tm_min=42, tm_sec=4, tm_wday=0, tm_yday=341, tm_isdst=0)`
 - Das Beispiel wurde also nach UTC am 07.12.2015 um 09:42:04 Uhr ausgeführt

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - localtime([secs])
 - Genau wie gmtime, jedoch wandelt diese Funktion den übergebenen Timestamp in eine Angabe der lokalen Zeitzone um
 - mktime(t)
 - Diese Funktion wandelt eine struct_time-Instanz in einen Unix-Timestamp der Lokalzeit um
 - Der Rückgabewert ist eine Gleitkommazahl

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - localtime([secs]), mktime(t)
 - Die Funktionen localtime und mktime sind jeweils Umkehrfunktionen voneinander
 - Beispiel

```
>>> t1 = time.localtime()
>>> t2 = time.localtime(time.mktime(t1))
>>> t1 == t2
True
```

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - sleep(secs)
 - Die Funktion sleep unterbricht die Programmausführung für die übergebene Zeitspanne
 - Der Parameter secs muss dabei eine Gleitkommazahl sein, die die Dauer der Unterbrechung in Sekunden angibt
 - Wenn Sie ein Program mittels sleep unterbrechen, befindet es sich im Leerlauf und beansprucht den Prozessor nicht

Python

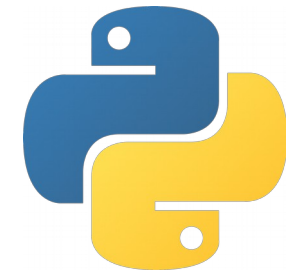
Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strftime(format[, t])`
 - Diese Funktion wandelt die `struct_time`-Instanz `t` oder ein neunelementiges Tupel `t` in einen String um
 - Dabei wird mit dem ersten Parameter namens `format` ein String übergeben, der das gewünschte Format des Ausgabestrings enthält
 - Ähnlich wie der Formatierungsoperator für Strings enthält der Format-String eine Reihe von Platzhaltern, die im Ergebnis durch die entsprechenden Werte ersetzt werden
 - Jeder Platzhalter besteht aus einem Prozentzeichen und einem Identifikationsbuchstaben

Python

Datum und Zeit

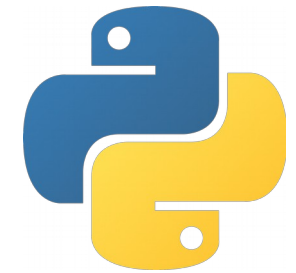


- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strftime(format[, t])`

Platzhalter	Bedeutung
%a	lokale Abkürzung für den Namen des Wochentags
%A	der komplette Name des Wochentags in der lokalen Sprache
%b	lokale Abkürzung für den Namen des Monats
%B	der vollständige Name des Monats in der lokalen Sprache
%c	das Format für eine angemessene Datums- und Zeitdarstellung auf der lokalen Plattform

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strftime(format[, t])`

Platzhalter	Bedeutung
%d	Nummer des Tages im aktuellen Monat. Ergibt einen String der Länge 2 im Bereich [01,31].
%H	Stunde im 24-Stunden-Format. Das Ergebnis hat immer zwei Ziffern und liegt im Bereich [00,23].
%I	Stunde im 12-Stunden-Format. Das Ergebnis hat immer zwei Ziffern und liegt im Bereich [01,12].
%j	Nummer des Tages im Jahr. Das Ergebnis hat immer drei Ziffern und liegt im Bereich [001, 366].
%m	Nummer des Monats, bestehend aus zwei Ziffern im Bereich [01,12]

Python

Datum und Zeit

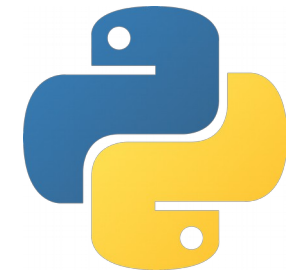


- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strftime(format[, t])`

Platzhalter	Bedeutung
%M	Minute als Zahl mit zwei Ziffern. Liegt immer im Bereich [00,59].
%p	die lokale Entsprechung für AM bzw. PM ⁵
%S	Sekunde als Zahl mit zwei Ziffern. Liegt immer im Bereich [00,61].
%U	<p>Nummer der aktuellen Woche im Jahr, wobei der Sonntag als erster Tag der Woche betrachtet wird. Das Ergebnis hat immer zwei Ziffern und liegt im Bereich [01,53].</p> <p>Der Zeitraum am Anfang eines Jahres vor dem ersten Sonntag wird als 0. Woche gewertet.</p>
%w	Nummer des aktuellen Tages in der Woche. Sonntag wird als 0. Tag betrachtet. Das Ergebnis liegt im Bereich [0,6].

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strftime(format[, t])`

Platzhalter	Bedeutung
%W	wie %U, nur dass statt des Sonntags der Montag als 0. Tag der Woche betrachtet wird
%x	Datumsformat der lokalen Plattform
%X	Zeitformat der lokalen Plattform
%y	Jahr ohne Jahrhundertangabe. Das Ergebnis besteht immer aus zwei Ziffern und liegt im Bereich [00,99].
%Y	komplette Jahreszahl mit Jahrhundertangabe
%Z	Name der lokalen Zeitzone oder ein leerer String, wenn keine lokale Zeitzone festgelegt wurde
%%	Ergibt ein Prozentzeichen % im Resultatstring.

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strftime(format[, t])`
 - Beispiel

```
>>> time.strftime("%d.%m.%Y um %H:%M:%S Uhr")
'07.12.2015 um 10:55:47 Uhr'
```
 - Ausgabe des aktuellen Zeitpunktes in einem für Deutschland üblichen Format

Python

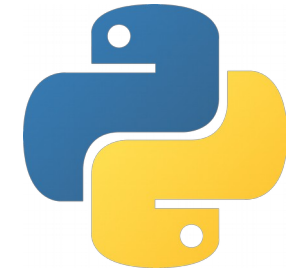
Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strptime(string[, format])`
 - Mit `strptime` wandeln Sie einen Zeit-String wieder in eine `time.struct_time`-Instanz um
 - Der Parameter `format` gibt dabei das Format an, in dem der String die Zeit enthält
 - Der Aufbau solcher Format-Strings ist der gleiche wie bei `strftime`

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - `strptime(string[, format])`
 - Beispiel

```
>>> zeit_string = '19.09.2011 um 00:21:17 Uhr'
>>> time.strptime(zeit_string, "%d.%m.%Y um %H:%M:%S
Uhr")
time.struct_time(tm_year=2011, tm_mon=9, tm_mday=19,
tm_hour=0, tm_min=21, tm_sec=17, tm_wday=0,
tm_yday=262, tm_isdst=-1)
```
 - Wenn Sie den optionalen Parameter `format` nicht angeben, wird der Standardwert `"%a %b %d %H:%M:%S %Y"` verwendet
 - Dies entspricht dem Ausgabeformat von `ctime`

Python

Datum und Zeit



- Elementare Zeitfunktionen – time
 - Funktionen vom Modul → time
 - time()
 - Dies gibt den aktuellen Unix-Zeitstempel in UTC als Gleitkommazahl zurück
 - Beachte
 - Nicht alle Systeme unterstützen eine höhere Auflösung als eine Sekunde und der Nachkommateil ist somit nicht unbedingt verlässlich

Python

Datum und Zeit



- Komfortable Datumsfunktionen – datetime
 - Funktionen vom Modul → datetime
 - Das Modul datetime ist im Vergleich zum time-Modul abstrakter und durch seine eigenen Zeit- und Datumstypen auch angenehmer zu benutzen
 - Hinweis
 - Es wird zwischen naiven und bewussten Zeitobjekten unterschieden
 - Ein bewusstes Zeitobjekt ist sich der Zeitzone, auf die es sich bezieht, bewusst
 - Ein naives Zeitobjekt kennt keine Informationen zu seiner Zeitzone

Python

Datum und Zeit



- Komfortable Datumsfunktionen – datetime
 - Funktionen vom Modul → datetime
 - Aufgabe
 - Befassen Sie sich mit dem Modul 'datetime'
 - Erstellen Sie ein Programm, welches die relativen Feiertage berechnet (z.B. Ostern, Pfingsten usw.)
 - Legen Sie ein Menü fest, welches den User darauf hinweist, welchen Wert er eingeben muss, um das Programm zu starten
 - Hinweis
 - Berechnungen und entsprechende Formeln finden Sie im Internet