

9.3 Datums- und Zeit-Funktionen

Gambas stellt Ihnen für die Arbeit mit Datums- und Zeitwerten nicht nur den aktuellen Zeitstempel zur Verfügung sondern auch Funktionen u.a. zum Generieren eines Datums oder zur Isolation einzelner Datumskomponenten wie zum Beispiel Monat oder Jahr oder Wochentag oder Kalenderwoche aus einem gegebenen Datum oder Format-Funktionen zur Ausgabe von Datum und Zeit in frei wählbaren Formaten.

Einige Komponenten zur Ein- oder Ausgabe von Datum oder Zeit-Werten arbeiten mit validen Werten für Datum und Zeit (DateChooser) – bei anderen müssen Sie selbst prüfen, ob ein valides Datum oder eine valide Zeit vorliegt (ValueBox).

9.3.1 Übersicht zu den Datums- und Zeitfunktionen

Funktion	Datentyp	Beschreibung
Date (Expression)	Date	Gibt nur das Datum eines Ausdrucks zurück, der als Datum interpretiert werden kann.
Time (Expression)	Date	Gibt nur die Zeit eines Ausdrucks zurück, der als Datum interpretiert werden kann.
DateAdd (Date as Date , Period AS Integer , Count AS Integer)	Date	Addiert das Vielfache einer spezifizierten Periode zu einem gegebenen Datum und gibt das neue Datum zurück
DateDiff (Date1 AS Date , Date2 AS Date , Period AS Integer)	Integer	Gibt die Anzahl der spezifizierten Periode zwischen zwei Datumswerten zurück.
Year (Date AS Date)	Integer	Gibt das Jahr eines Datums als Integer-Zahl zurück.
Month (Date AS Date)	Integer	Gibt den Monat eines Datums als Integer-Zahl zwischen 1 und 12 zurück.
Day (Date AS Date)	Integer	Gibt den Tag eines Datums als Integer-Zahl zwischen 1 und 31 zurück.
Hour (Date AS Date)	Integer	Gibt die Stunde eines Datums als Integer-Zahl zwischen 0 und 23 zurück.
Minute (Date AS Date)	Integer	Gibt die Minuten eines Datums als Integer-Zahl zwischen 0 und 59 zurück.
Second (Date AS Date) AS Integer	Integer	Gibt die Sekunden eines Datums als Integer-Zahl zwischen 0 und 59 zurück.
Week (Date AS Date [, First-DayOfWeek AS Integer , FullWeek AS Boolean])	Integer	Gibt die Kalender-Woche als Integer-Zahl zwischen 0 und 52 zurück. Der Funktionswert wird durch die Parameter zur Angabe des ersten Wochentages und zur vollen Woche beeinflusst.
WeekDay (Date AS Date)	Integer	Gibt den Wochentag als Integer-Zahl zwischen 0 und 6 zurück.
Now [()]	Date	Gibt das aktuelle Datum und die aktuelle Zeit zurück.
Timer [()]	Float	Gibt die Anzahl der Sekunden nach dem Programmstart als reelle Zahl zurück.

Tabelle 9.3.1.1: Übersicht Datums- und Zeitfunktionen

Von besonderem Interesse wird die Date()-Funktion sein, mit der Sie aus der Angabe des Jahres, des Monats und des Tages ein valides Datum generieren können. Wenn Sie auch die optionalen Argumente Stunden, Minuten, Sekunden und Millisekunden angeben, dann werden auch die Zeitwerte im Datum gesetzt, sonst werden diese auf 00 gesetzt:

```
Datum = Date ( Jahr , Monat , Tag [ , Stunden , Minuten , Sekunden , Millisekunden ] ) AS Date
```

Im Zusammenhang mit der Date()-Funktion werden Sie auch diese Funktionen schätzen:

- IsDate(String), um zu prüfen, ob ein String sicher als Datum interpretiert werden kann,
- CDate (Expression AS Variant) zur Konvertierung einer Zahl (Float) oder eines Strings in einen Datum-String (ohne Beachtung der eingestellten Lokalisierung),

- Val (String), mit der Sie einen String u.a. in ein Datum konvertieren können, wobei diese Funktion die eingestellte Lokalisierung berücksichtigt oder
- Str (Expression), um ein Datum in einen String zu konvertieren. Auch diese Funktion berücksichtigt die eingestellte Lokalisierung!

Ausführliche Informationen zu diesen 4 Funktionen und weiteren Konvertierungsfunktionen können Sie im → Kapitel 9.10 Konvertierungsfunktionen nachlesen.

9.3.2 Hinweise

Für die Berechnung eines neuen Datums aus einem Start-Datum und dem addierten Zeit-Intervall können Sie diese Funktion einsetzen:

```
DateAdd ( Date as Date , Period AS Integer , Count AS Integer ) AS Date
```

- *Date* ist das Start-Datum,
- *Period* ist der Typ des Zeitintervalls: Jahr, Monat, Tag, ... ,
- *Count = c* ist die Anzahl der zu addierenden Zeitintervalle. Standard ist $c = 1$.

Liste der Konstanten für Zeitintervalle (Period):

Konstante	Effekt
gb.Second	Addiert c Sekunden (seconds).
gb.Minute	Addiert c Minuten (minutes).
gb.Hour	Addiert c Stunden (hours).
gb.Day	Addiert c Tage (days).
gb.Week	Addiert c Wochen (weeks).
gb.WeekDay	Addiert c Wochentage (Samstag und Sonntag werden ignoriert).
gb.Month	Addiert c Monate (months).
gb.Quarter	Addiert c Vierteljahre (quarters).
gb.Year	Addiert c Jahre (years).

Für die Berechnung von Differenzen zwischen Datumswerten ist diese Funktion einsetzbar:

```
DateDiff ( Date_1 AS Date , Date_2 AS Date , Period AS Integer ) AS Integer
```

- Es gilt: $Date_1 \leq Date_2$
- Das Datum 1 liegt vor dem Datum 2,
- *Period* ist der Typ des Zeitintervalls.

Der Funktionswert von DateDiff(..) für die Zeit-Differenz ist die *Anzahl der Zeitintervalle* zwischen Datum1 und Datum2. Es werden nur *vollständige* Intervalle gezählt; intern wird abgerundet. Die Konstanten für das Zeitintervall sind die gleichen wie für die DateAdd()-Funktion.

Bei der Funktion WeekDay(..) zur Ermittlung des Wochentages aus einem Datum

```
WeekDay ( Date AS Date ) AS Integer
```

ist zu beachten, dass eine Zahl zwischen 0 (Sonntag) und 6 (Samstag) zurückgegeben wird. Hier eine Liste der vordefinierten Konstanten, die den Wochentagen zugeordnet sind:

Konstante	Wochentag	Rückgabe-Wert
gb.Sunday	Sonntag	0
gb.Monday	Montag	1
gb.Tuesday	Dienstag	2
gb.Wednesday	Mittwoch	3
gb.Thursday	Donnerstag	4
gb.Friday	Freitag	5
gb.Saturday	Samstag	6

Beispiele:

```
Dim aTagesListe As String[]
Dim dDate As Date

aTagesListe = Split("Sonntag, Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag", ",")
dDate = Now()
```

```
Print "Heute ist "; aTagesListe[WeekDay(dDate)]; " ("; Format$(dDate, "dd.mm.yyyy"); ")."
dDate = DateAdd(Now, gb.day, 7 - WeekDay(Now))
Print "Der nächste Sonntag fällt auf den "; Format$(dDate, "dd.mm.yyyy"); "!"
```

Ausgabe in der Konsole der IDE:

```
Heute ist Sonntag (01.06.2014).
Der nächste Sonntag fällt auf den 08.06.2014!
```

9.3.3 Schaltjahre

Ob ein Jahr ein Schaltjahr ist, kann leicht mit den folgenden Quelltext-Ausschnitten überprüft werden:

```
Public Function IsLeapYear(iYear As Integer) As Boolean
    Return (iYear >= 1583 AND ((iYear Mod 4 = 0 And iYear Mod 100 <> 0) Or (iYear Mod 400 = 0)))
End ' IsLeapYear(..)
```

```
Public Sub btnSchaltjahr_Click()
    Print "2008: "; IsLeapYear(2008)
    Print Year(Now()); ": "; IsLeapYear(Year(Now()))
    Print "1500: "; IsLeapYear(1500)
    Print "2000: "; IsLeapYear(2000)
End ' btnSchaltjahr_Click()
```

In der Konsole der IDE zeigt sich:

```
2008: True
2014: False
1500: False
2000: True
```

9.3.4 Zeitstempel - Unix-TimeStamp

Ein Unix-Zeitstempel (Unix-TimeStamp) ist eine Zahl, welche die Anzahl der Sekunden seit dem 1. Januar 1970 repräsentiert.

So konvertieren Sie das aktuelle (Gambas-)Datum in einen Unix-Zeitstempel:

```
Print DateDiff(CDate("1/1/1970"), Now(), gb.Second)
Print "UNIX-TimeStamp : "; DateDiff(CDate("1/1/1970"), Now(), gb.Second); " Sekunden"
```

So funktioniert die Umwandlung eines Unix-Zeitstempels in ein (Gambas-)Datum:

```
Print DateAdd(CDate("1/1/1970"), UnixTimeStamp, gb.Second)
```

Auf der Website <http://www.timeanddate.com/worldclock/germany/berlin> finden Sie Interessantes zu den Themen Datum, Zeit und Zeitzonen. Der Zeitstempel begegnet Ihnen auch im Zusammenhang mit MySQL und deren Daten-Typen in der Konstanten: `TimeStamp As String = "TIMESTAMP"`.

9.3.5 Valide Datums- und Zeitwerte

Um ein Datum zu generieren, können Sie verschiedene Steuerelemente einsetzen. In der nächsten Übersicht finden Sie das Steuerelement und das Kapitel, in dem das Steuerelement beschrieben wird.

- `DateChooser` → Kapitel 17.13
- `ValueBox` → Kapitel 16.9
- `MaskBox` → Kapitel 16.7
- `ButtonBox` → Kapitel 16.8.2 (mit Maske)
- `DateBox` → Kapitel 16.6.2
- `Eingabefeld` → Kapitel 19.6.5.2 Prüfung der Syntax von Zeichenketten

Von allen Steuerelementen gibt es einen Favoriten: Es ist die `DateBox`, denn die `DateBox` als spezialisierte `ButtonBox` mit einer Datum/Zeit-Maske setzt neben der Eingabe und Anzeige eines Datums auf den Datum/Zeit-Auswahl-Dialog im `DateChooser`.

Wenn Sie Datums- und Zeit-Werte im Quelltext direkt verwenden wollen, dann sollten Sie sich die Beispiele im nächsten Abschnitt ansehen.

9.3.6 Beispiele

Die folgenden Beispiele sollten Sie in ein kleines Projekt übernehmen und die Argumente sowie Konstanten vielfach variieren. Viele Ausgaben werden zusätzlich in unterschiedlicher Weise formatiert. Die Beschreibung der Format-Funktion *Format(..)* finden Sie im → Kapitel 9.1 und eine Übersicht von Format-Festlegungen für Datums- und Zeit-Werte im → Kapitel 9.2.

```
Dim aTagesListe As String[]
Dim dDate As Date

aTagesListe = Split("Sonntag, Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag", ",")
dDate = Now()
Print "Heute ist "; aTagesListe[WeekDay(dDate)]; "."
Print "Heute ist "; Format$(Now(), "dddd.")
dDate = DateAdd(Now, gb.day, 7 - WeekDay(Now))
Print "Der nächste Sonntag fällt auf den "; Format$(dDate, "dd.mm.yyyy"); "!
Print "Bis Weihnachten sind noch "; DateDiff(Now(), Date(Year(Now()), 12, 24), gb.Day); " Tage."

Print "UNIX-Zeitstempel (aktuell): "; DateDiff(CDate("1/1/1970"), Now(), gb.Second); " Sekunden"
Print "UNIX-TimeStampToDate (aktuell): "; DateAdd(CDate("1/1/1970"), gb.Second, DateDiff(CDate("1/1/1970"), Now(), gb.Second))
Print "UNIX-Zeitstempel vom 5.6.2014: "; DateDiff(CDate("1/1/1970"), Date(2014, 6, 5), gb.Second); " Sekunden"

Print "Heute ist "; Format$(Now(), "dddd"); ", der "; Format$(Now(), "d. mmmm yyyy.")
Print "Es war "; Format$(Now, "hh:nn"); " Uhr!"
Print "Es war genau "; Format$(Now, "hh:nn:ss.uuu"); " Uhr!"
Print "Es war "; Format$(Now, "hh:nn"); " Uhr"; Format$(Now, " (t)")
Print "Zeit = "; Format(Now(), "hh:nn:ss"); " Uhr!" ' Aktuelle Küchen-Zeit
Print "UTC = "; Format(Time(DateAdd(Now(), gb.second, System.TimeZone)), "hh:nn:ss"); " Uhr"
Print "Aktuelle Kalenderwoche: "; Week(Now(), gb.Monday, True)
```

```
Heute ist Donnerstag.
Heute ist Donnerstag.
Der nächste Sonntag fällt auf den 08.06.2014!
Bis Weihnachten sind noch 202 Tage.
UNIX-Zeitstempel (aktuell): 1401964266 Sekunden
UNIX-TimeStampToDate (aktuell): 05.06.2014 10:31:06
UNIX-Zeitstempel vom 5.6.2014: 1401926400 Sekunden
Heute ist Donnerstag, der 5. Juni 2014.
Es war 10:31 Uhr!
Es war genau 10:31:06.349 Uhr!
Es war 10:31 Uhr (CET)
Zeit = 10:31:06 Uhr!
UTC = 09:31:06 Uhr
Aktuelle Kalenderwoche: 22
```