

Herzlich willkommen auf dieser Infoseite

Diese Seite richtet sich vor allem an **Neueinsteiger** bei der Wetterauswertesoftware WSWIN (<http://www.pc-wetterstation.de>) von Werner Krenn.

Auf den folgenden Seiten soll **nicht** die Arbeit mit WSWIN erklärt werden, sondern **die Bedeutung und Verwendung der vielen Variablen in WSWIN**.

Das **Gesamtdokument** (ca 550 kByte) erhalten Sie hier: [WSWIN-Praxis](#)

Nur wer in der Verwendung der Variablen sicher ist, wird im Umgang mit den benutzerdefinierten Dateien keine oder wenig Probleme haben.

Fortgeschrittene Benutzer werden in den Menüpunkten weiter unten weiterführende Hinweise finden.

Wer die im linken Abschnitt erläuterten Punkte alle durchgelesen hat, sollte einmal auch die folgenden Seiten besuchen:

- [Arbeit mit benutzerdefinierten Dateien](#)
- [Erstellen eines Wetter-Tickers](#)

Letztmals bearbeitet am 22.03.2009

Inhaltsverzeichnis

| Thema | Seite |
|---|-------|
| <u>Grundlagen</u> | 3 |
| <u>Variablenarten</u> | 5 |
| <u>Variablen ohne Index</u> | 14 |
| <u>Variablen zur Text- u. Bildausgabe</u> | 15 |
| <u>Variablen mit Benutzervorgabe</u> | 17 |
| <u>Zusammenfassung</u> | 20 |
| <u>Anwendungshilfen</u> | 21 |

Grundlegende Anhaltspunkte

WSWIN stellt jede Menge **Variable** zur Verfügung.

Ihr Zweck ist es, Wetterdaten und andere Werte, die in WSWIN gewonnen werden oder festgehalten sind, für html-Dateien, die man beispielsweise im Internet veröffentlichen möchte, zur Verfügung zu stellen.

Was damit gemeint ist, soll weiter unten erklärt werden.

Doch zunächst sollte man wissen, wie solche Variablen aufgebaut sind.

Variablen sind daran zu erkennen, dass sie mit "%" beginnen und mit "%" enden.

Eine der vielen Variablen in WSWIN lautet beispielsweise

```
%ws_date%
```

Diese Variable beinhaltet das aktuelle **Datum** (date).

Ein andere Variable heisst

```
%ws_time%
```

Diese Variable beinhaltet die aktuelle **Uhrzeit** (time).

Wann immer man in einer anderen Datei (häufig ist es die Datei custom.txt und daraus erzeugt die Datei custom.html) das aktuelle Datum oder die aktuelle Uhrzeit benötigt, werden diese Variablen verwendet.

(s. dazu auch meine anderen Hilfe-Seiten über die Verwendung der [custom.txt](#))

Man könnte also so zusammenfassen:

1. WSWIN **liest** Wetterdaten aus dem Interface (beispielsweise WS2000 PC) aus, **speichert** diese (in der Datei ws_02_03.dat für Februar 2003) und stellt diese Daten **grafisch** und in **Tabellen** dar (im Programm selbst - ohne weiteres Zutun).
2. Darüberhinaus erzeugt WSWIN standardmäßig eine Menge von html-Dateien für das Internet (bei entsprechender Einstellung), z.B. aktuell.gif - aktuell.htm - minidisplay.gif - current.html usw. (ebenfalls selbständig).

Für alle diese beiden Punkte ist keine Kenntnis über Variablen nötig, denn dies erledigt das Programm von allein.

Warum benötigt man dann Variablen?

1. Wer darüber hinaus noch **zusätzliche** Dateien (z.B. für seine Internetpräsenz) erstellen will, kommt über die Arbeit mit der Datei **custom.txt** (s. oberer Link) nicht hinweg
2. In dieser Datei custom.txt werden nämlich mit Hilfe der Variablen von WSWIN die Daten bereitgestellt, auf die man Wert legt, z.B.
 - die aktuelle Außentemperatur
 - die aktuelle Windgeschwindigkeit
 - der aktuelle Luftdruck
 - Minimal- u. Maximalwerte
 - **seit Version 2.8.3.0** auch Variablen zum **Vergleich** (z.B. *"ist die Temperatur des Nordsensors höher als die des Südsensors"*)
 - Variablen zur **Berechnung** (z.B. *km/ in m/s*)
 - mit **Speicherfunktion** (*Variableninhalt merken und an custom.txt*)

übergeben)

- seit **Version 2.8.5.1** auch Variable für **beliebige zurückliegende Werte**
(z.B. *"wie war die Temperatur heute vor einem Jahr"*)

Um hier die richtigen Werte zu erhalten, ist die Kenntnis der Variablen und ihrer Bedeutung nötig.
Dies soll in den folgenden Seiten erläutert werden.

Lesen Sie später dazu auch meine Bemerkungen über eben diese custom.txt und die Arbeit mit ihr.

Die vielen verschiedenen Variablen von WSWIN

Wer sich informieren will, welche Variablen es in WSWIN gibt, sollte unbedingt die **Hilfedatei von WSWIN** öffnen und hier den Punkt "Inhalt - Variablen" wählen. Hier sind alle verfügbaren Variablen benannt und beschrieben.

Dennoch sollen hier ein paar Erläuterungen erfolgen.

Es gibt in WSWIN Variablen für unterschiedliche Zwecke:

- Darstellung aktueller Werte (aktuelle Außentemperatur)
- Darstellung von Durchschnittswerten (Durchschnittstemperatur des Tages, der Woche, des Monats usw)
- Darstellung allgemeiner Werte (Namen von Sensoren, Status von Sensoren usw.)
- Darstellung von Langzeitwerten (Maximaltemperatur über ein Jahr usw.)
seit Version 2.8.3.0 neu
- Vergleich von Werten aus Variablen
- Berechnung von Werten aus Variablen
- Speicherung von Werten
seit Version 2.8.5.1 neu
- Darstellung beliebiger zurückliegender Werte
- **seit Version 2.94.0 neu**
- Berechnung von statistischen Werten (Schnitt, Maximum, Minimum usw.) **für beliebige Zeiträume** + Speichermöglichkeit

Dies soll nachfolgend erläutert werden.

Die wohl am häufigsten verwendeten Variablen sind diejenigen, die aktuelle Werte, Maximal- u. Minimalwerte, Durchschnittswerte oder Langzeitwerte erzeugen.

Dies sind Variablen, die in der Regel in der sog. *custom.txt* verwendet werden. (zum Erstellen der Datei *custom.html*)

Stellvertretend wird hier auf eine Auswahl eingegangen

| Variablenname | erzeugt | Bedeutung von "X" |
|--------------------|--|--|
| %curval[x]% | aktuellen Wert | Der Wert von X (der sog. Index) gibt an, welchen Sensor man wünscht, z.B. X=2 --> Temperatur 1 <i>oder</i> X=33 --> Luftdruck s. hierzu unbedingt die Hilfedatei, Stichwort "Indexe" |
| %avgval[x]% | Durchschnittswert | |
| %minval[x]% | Minimalwert | |
| %mintime[x]% | Zeit des Minimalwertes (je nach Einstellung aus Normalanzeige oder 24h-Anzeige) | |
| %mintimecurday[x]% | Zeit des Minimalwertes des aktuellen Tages (auch bei 24h-Ansicht) | |

Ein Beispiel soll dies zeigen

| Inhalt der <i>custom.txt</i> | Inhalt der erzeugten <i>custom.html</i> |
|--|---|
| Außentemperatur aktuell: %curval[2]% | Außentemperatur aktuell: 3,6 °C |
| Durchschnittstemperatur: %avgval[2]% | Durchschnittstemperatur: 1,4 °C |
| Minimum heute: %minval[2]% um %mintime[2]% | Minimum heute:-2,3 °C um 07:56 |

| | |
|--|--|
| Luftdruck aktuell: %curval[33]% | Luftdruck aktuell: 1023 hPa |
| durchschn. Luftdruck: %avgval[33]% | durchschn. Luftdruck: 1019 hPa |
| minimaler Luftdruck: %minval[33]% um %mintime[33]% | minimaler Luftdruck: 1016 hPa um 03:44 |

Hiermit soll auch deutlich werden:

Der Variablenname curval erfüllt **zwei** Funktionen:

- er erzeugt den **aktuellen** Wert
- es ist je nach Indexzahl (**2** oder **33**) entweder der Wert der **Außentemperatur** oder des **Luftdrucks**

Genauso verhält es sich mit allen anderen Variablen und Indexen.

Auf eine **neue Variable** (seit Version **2.82.0**) soll hier besonders eingegangen werden:

| Variablenname | erzeugt | Bedeutung von |
|--|---|---|
| %curminmaxhour[a,h,x]% %curminmaxhour[a,h,x]% | aktuelle Werte, Minimalwerte, Maximalwerte, Durchschnittswerte und Differenzwerte | "a" hat Wert 0: aktueller Wert "a" hat Wert 1: Minimalwert "a" hat Wert 2: Maximalwert "a" hat Wert 3: Durchschnitt "a" hat Wert 4: Differenz |
| %curminmaxhour[a,h,x]% | | "h" kann die Werte von 1 - 24 für die letzten 1 - 24 Stunden haben; hat "h" den Wert "0", so wird (z.B. bei Minimum oder Maximum) der Wert des aktuellen Tages genommen, auch wenn die 24h-Ansicht gewählt ist |
| %curminmaxhour[a,h,x]% | | Der Wert von x (der sog. Index) gibt an, welchen Sensor man wünscht, z.B. X=2 -->Temperatur 1 oder X=33 --> Luftdruck |

Zugrunde sollen folgende Daten liegen:

| Zeit | Sensor 1 |
|-------|----------|
| 10:50 | 23,2 |
| 10:20 | 22,7 |
| 9:50 | 20,7 |
| 9:20 | 19,1 |
| 8:50 | 19,4 |
| 8:20 | 19,7 |
| 7:50 | 18,4 |
| 7:20 | 18,3 |
| 6:50 | 17,8 |

Wie diese Daten ausgewertet werden, soll wiederum an einem Beispiel aufgezeigt werden (als Sensor wird hier immer Sensor 1 [ID2] genommen):

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|--|---|
| Minimaltemperatur letzte 4 Stunden: %curminmaxhour[1,4,2]% | Minimaltemperatur letzte 4 Stunden: 17,8°C |
| Durchschnittstemperatur letzte 2 Stunden: %curminmaxhour[3,2,2]% | Durchschnittstemperatur letzte 2 Stunden: 21,0°C |
| Maximaltemperatur letzte 3 Stunden: %curminmaxhour[2,3,2]% | Maximaltemperatur letzte 3 Stunden: 23,2°C |
| Temperaturdifferenz der letzten 2 Stunden: %curminmaxhour[4,2,2]% | Temperaturdifferenz der letzten 2 Stunden: 3,8°C |
| aktueller Wert der vergangenen Stunde: %curminmaxhour[0,1,2]% | aktueller Wert der vergangenen Stunde: 20,7 |
| aktueller Wert vor 2 Stunden: %curminmaxhour[0,2,2]% | aktueller Wert vor 2 Stunden: 19,4°C |
| Maximalwert des aktuellen Tages: %curminmaxhour[2,0,2]% | Maximalwert des aktuellen Tages: 23,2°C |

Wie man sieht, ist hiermit jede nur erdenkliche Auswertung möglich. Man beachte, dass vor allem mit den letzten beiden Beispielen in der Tabelle **aktuelle** Werte genau festgelegter Zeitabstände möglich sind (im Gegensatz dazu gibt es ja immer schon die Variable curval[x], die aber nur in der Lage ist, den letzten (aktuellen) Wert auszugeben.

Was häufig zu Missverständnissen führt:

Für welchen **Zeitraum** gelten die Variablen? (gemeint sind hier alle Variablen **außer** der "Sondervariablen" %curminmaxhour[a,h,x]%)

Sie gelten immer für den **eingestellten Betrachtungszeitraum** in WSWIN.

- ist im Menü "Ansicht - Tag - Normalanzeige" gewählt, so gelten sie für den Zeitraum 00:00 Uhr bis aktuelle Uhrzeit (z.B. 16:00 Uhr)
- ist im Menü "Ansicht - Tag - 24h Anzeige" gewählt, so gelten sie für die letzten 24 Stunden (z.B. 12:00 Uhr Vortag bis 12:00 Uhr aktueller Tag)

Wiederum neu ist eine spezielle Variable (ab Version 2.83.0):

Wer bisher Minimal- u. Maximalwerte eines Monats oder des Jahres ermitteln und ausgeben wollte, war häufig auf die template-Dateien angewiesen. Hier wurde eine neue Variable eingeführt, die in der custom.txt eingesetzt werden kann:

`%ws_m_XXXval[%ws_month%,x]%`

| Variablenname | erzeugt | Bedeutung von |
|---|--|--|
| <code>%ws_m_minval[%ws_month%,x]%</code> <code>%ws_m_mintime[%ws_month%,x]%</code> <code>%ws_m_minday[%ws_month%,x]%</code> <code>%ws_m_maxval[%ws_month%,x]%</code> | Minimalwerte, Maximalwerte und Durchschnittswerte von Wert, Zeit und Tag | <code>%ws_month%</code> wird durch den aktuellen Monat ersetzt, es kann aber auch ein fester Zahlenwert (z.B. "3" für März) eingesetzt werden. Wird der Wert "0" eingesetzt, so werden die Werte des Jahres ermittelt. |
| <code>%ws_m_avg[%ws_month%,x]%</code> | | "x" entspricht der jeweiligen Sensor-ID, also z.B. "2" für Außensensor |

Praktisches Beispiel:

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|--|--|
| Minimaltemperatur von <code>%ws_month%</code> : <code>%ws_m_minval[%ws_month%,2]%</code> | Minimaltemperatur von Januar: -10,2 °C |
| Maximaltemperatur von Oktober : <code>%ws_m_maxval[10,2]%</code> | Maximaltemperatur von Oktober : 19 °C |
| Durchschnittstemperatur Jahr : <code>%ws_m_avg[0,2]%</code> | Durchschnittstemperatur Jahr : 9,2 °C |
| Hierbei ist zu beachten, dass bei Einsatz der Variablen <code>%ws_month%</code> (erstes Beispiel) die Werte immer auf den aktuellen Monat bezogen sind, während bei Angabe einer Zahl (1-12) ein ganz bestimmter Monat ausgegeben werden kann. | |

Neu hinzugekommen ist eine Variable zur Ausgabe beliebig zurückliegender Werte(ab Version 2.85.1):

| Variablenname | erzeugt | Bedeutung von |
|---|---|---|
| <code>%avgvalpast[id~dd~mm~yyyy]%</code> <code>%minvalpast[id~dd~mm~yyyy]%</code> <code>%mintimepast[id~dd~mm~yyyy]%</code> <code>%mindatepast[id~dd~mm~yyyy]%</code> <code>%maxvalpast[id~dd~mm~yyyy]%</code> <code>%maxtimepast[id~dd~mm~yyyy]%</code> <code>%maxdatepast[id~dd~mm~yyyy]%</code> <code>%t_sensdaypast[dd~mm~yyyy]%</code> <code>%sundaypast[dd~mm~yyyy]%</code> | Durchschnittswert, Minimalwert, Maximalwert, Zeitsensorwert und Tagessonnenzeit des gewählten Sensors (id), Tages (dd), Monats (mm) und Jahres (yyyy) | id entspricht den Sensoren 1 - 46, also z.B. "2" für Außensensor dd den Tagen 00 bis 31 oder zurückliegenden Tagen (-8 = vor 8 Tagen) mm den Monaten 00 bis 12 oder zurückliegenden Monaten (-3 = vor 3 Monaten) |

Folgendes ist gut zu wissen: Die Sensor ID "0" entspricht dem für den Taupunkt eingestellten Sensor, also in der Regel dem Außensensor

Praktische Beispiele:

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|---|---|
| Maximum heute %maxvalpast[2~00~00~0000]% | Maximaltemperatur des Außensensors am heutigen Tag |
| Maximum vor einem Jahr %maxvalpast[2~00~00~-001]% | Maximaltemperatur des Außensensors heute vor einem Jahr |
| Maximum vor 4 Tagen %maxvalpast[2~-4~00~0000]% | Maximaltemperatur des Außensensors vor 4 Tagen |
| Maximum vor 2 Monaten %maxvalpast[2~00~-2~0000]% | Maximaltemperatur des Außensensors vor 2 Monaten |
| Maximum am 8.1.05 %maxvalpast[2~08~01~2005]% | Maximaltemperatur des Außensensors genau am 8.1.05 (also ein konkretes Datum) |
| Maximum vor einem Jahr %maxvalpast[2~00~00~-001]% also um %maxdatepast[2~00~00~-001]% | gibt nicht nur die Maximaltemperatur vor einem Jahr aus, sondern auch das dazugehörige Datum |

Hierbei ist folgendes zu beachten:

Die bis hierher genannten Beispiele funktionieren auch jahresübergreifend, was bedeutet: Wenn jetzt gerade Januar 2005 ist, so bringt %maxvalpast[2~00~-2~0000]% den Wert von November 2004

Was **nicht** funktioniert:

Maximum vor einem Monat **und** vor einem Jahr %maxvalpast[2~00~-1~001]% geht nicht, da Priorität Monat das Jahr auf aktuelles Jahr setzt.

ebenso nicht:

Maximum vor vier Tagen **und** vor einem Jahr %maxvalpast[2~-4~00~-001]% geht nicht, da das Jahr wegen "-4" bei "Tag" auf aktuelles Jahr gesetzt wird.

aber das braucht man auch nicht, oder?

Zuletzt:

Damit diese Variablen funktionieren, sollte die Erzeugung der neuen "Jahresdateien" (ws_yyyy.dat) nicht ausgeschaltet sein.

Wer aber darauf verzichten will, kann dies unter "Ansicht - Einstellungen" machen, indem er den Haken setzt bei "keine Jahr Datei"

Neu hinzugekommen ist eine Variablenart zur Ausgabe in beliebigen Einheiten und beliebiger Sprache (ab Version 2.91.4):

Einheitenwahl

| Variablenname | Umschaltwert | Bedeutung von |
|-----------------------------------|---|---|
| %ws_newunit[x]=u% + Variablenname | 0= °C - hPa - l/m ² - km/h 1= °F - mmHG - mm - m/s Sonderwerte: 2= Druck in mbar - Wind in mph 3= Wind in Knoten 4= Wind in Beaufort (vollständige Liste in WSWIN-Hilfe) | x entspricht der Sensor-ID u schaltet die Einheiten um |

Folgendes ist gut zu wissen: Diese "Umschaltung" gilt nur für die betreffende **Zeile** und darf pro Sensortyp auch nur **einmal** vorkommen. Der Einsatz dieser neuen Variablen erspart die bis dahin nötige Umwandlung durch die "Berechnungsvariablen".

Praktische Beispiele:

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|---|---|
| %ws_newunit[0]=1%Temperatur: %curval[0]% °F | Temperatur: 36.1 °F |
| %ws_newunit[0]=0%Temperatur: %curval[0]% °C | Temperatur: 2.3 °C |
| %ws_newunit[0]=1%%ws_newunit[35]=1% Temperatur: %curval[0]% °F und Windgeschw. %curval[35]% m/s | Temperatur: 36.1 °F und Windgeschw. 3 m/s |
| %ws_newunit[35]=3%Windgeschw. %curval[35]% Knoten | Windgeschw. 4 Knoten |

Wir ersehen hieraus folgendes:

`%ws_newunit[0]=0%` setzt den **Sensor**

`%ws_newunit[0]=0%` setzt die **Ausgabereinheit**

Es können in einer Zeile **verschiedene** Sensoren und **verschiedene** Einheiten gewählt werden
Was **nicht** funktioniert:

In ein und derselben Zeile kann die Temperatur nicht in verschiedenen Einheiten ausgegeben werden.

Falsch wäre also:

`%ws_newunit[0]=1%%Temperatur %curval[0]% °F` und jetzt `%ws_newunit[0]=0% Temperatur %curval[0]% °C`

in aufeinanderfolgenden Zeilen dagegen ist es durchaus möglich.

Sprachwahl

| Variablenname | Umschaltwert | Bedeutung von |
|--------------------------------|--|---|
| <code>%forecast_txt[x]%</code> | 0= deutsch 1= englisch 2= französisch (vollständige Liste in WSWIN-Hilfe) | x entspricht der Sprache # in WSWIN-Hilfe bedeutet, dass die Variable mehrsprachig ist |

Folgendes ist gut zu wissen: Diese "Umschaltung" der Sprache kann auch in einer Zeile für mehrere Sprachen erfolgen.

Praktische Beispiele:

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|---|----------------------------------|
| Wettervorhersage-Text englisch <code>%forecast_txt[1]%</code> | unstable |
| Wettervorhersage-Text deutsch <code>%forecast_txt[0]%</code> | veränderlich |

alle Variablen, die diese Funktionalität bieten, sind mit dem vorangestellten "#" gekennzeichnet. Dieses aber nicht verwenden, also **nicht**: `##%forecast_txt[1]%`

Ab Version 2.94.0 verfügt WSWIN über eine **Season-Variable**, die statistische Werte über beliebige Zeiträume ermittelt

| Variablenname | erzeugt | Bedeutung von |
|---|---|---|
| %season_avg[id~ddmmyyyy~ddmmyyyy]% %season_sum[id~ddmmyyyy~ddmmyyyy]% %season_min[id~ddmmyyyy~ddmmyyyy]% %season_mintime[id~ddmmyyyy~ddmmyyyy]% %season_mindate[id~ddmmyyyy~ddmmyyyy]% %season_max[id~ddmmyyyy~ddmmyyyy]% %season_maxtime[id~ddmmyyyy~ddmmyyyy]% %season_maxdate[id~ddmmyyyy~ddmmyyyy]% %season_days% | Durchschnittswert, Summe , Minimalwert, Minimumzeit, Minimumdatum, Maximalwerte wobei id der gewünschte Sensor ist und ddmmyyyy dem Anfangs- bzw. Enddatum entspricht | id entspricht den Sensoren -1 bis 46, also z.B. "2" für Außensensor dd den Tagen 00 bis 31 mm den Monaten 00 bis 12 yyyy dem Jahr (oder -001 ein Jahr zurück) |

Folgendes ist gut zu wissen: Die Sensor ID "0" entspricht dem für den Taupunkt eingestellten Sensor, also in der Regel dem Außensensor, die Sensor ID "-1" dem Feuchtesensor

Praktische Beispiele:

| Inhalt der custom.txt | die erzeugte custom.html gibt aus |
|--|---|
| Maximum Herbst 2008 %season_max[0~01092008~30112008]% | Maximaltemperatur vom 1.9.2008 bis 30.11.2008 |
| Maximum Herbst Vorjahr %season_max[0~0109-001~3011-001]% | Maximaltemperatur des vergangenen Herbstes |
| Schnitt Herbst 2007 %season_avg[0~01092007~30112007]% | Durchschnitt des Herbst 2007 |
| Schnitt Sommer 2007 %season_avg[0~01062007~31082007]% und Tage dieser Saison %season_days% | Schnitt des Sommers 2007 und Anzahl der Tage in diesem Zeitraum |
| Hierbei ist folgendes zu beachten: Diese Variable funktioniert auch jahresübergreifend , was bedeutet: Wer aus irgendeinem Grund den Schnitt vom 1.7.2006 bis 31.3.2008 benötigt, verwendet eben %season_avg[0~01072006~31032008]% Wichtig: Die Variable %season_days% muss in der Zeile verwendet werden, in der der entsprechende Zeitraum ausgewertet wird. | |
| Auf eine wichtige Besonderheit soll ausdrücklich hingewiesen werden | |
| Schnitt Herbst 2008 %season_avg[0~01092008~30112008]% | dieses Beispiel sollte nun bereits klar sein |
| %season_avg[0~01090000~30110000]% | was passiert aber, wenn das Beispiel dieser Zeile verwendet wird? |

Hier hängt es nun vom **aktuellen Datum** des PC's ab. Wenn wir uns bereits im aktuellen Datumsbereich befinden, also zwischen dem 1.9. und 30.11., dann wird der Durchschnittswert des Herbstes im **aktuellen Jahr** ermittelt.

Wenn wir aber noch davor sind, als z. B. im Bereich 12.4. des aktuellen Jahres, dann wird der Schnitt ermittelt vom 1.9. des **Vorjahres** bis zum aktuellen Datum.

Dies mag etwas verwirrend sein, ist aber nötig, um ungültige Werte zu vermeiden, denn es kann ja nicht im März des aktuellen Jahres der Schnitt des kommenden Herbstes ermittelt werden.

Fazit:

Wenn also keine konkreten Jahres-Datumswerte eingesetzt werden, sondern die Werte für aktuelles Jahr (0000) oder Vorjahr (-001), dann ist etwas Sorgfalt geboten..

Besonders praktisch: Diese Variablen können auch im Programm selbst genutzt werden unter "Wetter - Wettervorhersage - Statistikdaten"

Ein besonderes Schmankerl dort:

Durch Drücken des Buttons "WriteVar101" werden die dort berechneten Werte in einer Datei namens **ws_store.cfg** im Programmverzeichnis gespeichert und stehen dadurch ohne ständige Neuberechnung zur Verfügung, denn sie können mühelos in WSWIN wieder sichtbar gemacht werden im Menüpunkt

"Wetter - Wettervorhersage - Text" durch Eingabe der Variablennummer, in unserem Fall also "101"

Wer nicht nur den **Wert** speichern will, sondern auch gleich die **zugehörige Variable**, setzt neben dem Button "WriteVar101" noch einen Haken bei "C"

Auf der folgenden Seite soll auf Variablen eingegangen werden, die [keinen Index](#) benötigen.

Variablen ohne Index

Wie im vorhergehenden Beispiel gezeigt, erfüllen manche Variablen einen **zweifachen** Zweck: Ihr **Name** gibt den **Zweck** an (minval = Minimalwert) und ihr **Index** bezeichnet den **Sensor** (33 = Luftdruck)

Es gibt nun in WSWIN eine Vielzahl von Variablen, die nur eine **eindeutige** Funktion haben:

| Name | erzeugt |
|-------------|----------------------------|
| %rain1h% | Regen der letzten Stunde |
| %rainday% | Regen des aktuellen Tages |
| %rainweek% | Regen der aktuellen Woche |
| %rainmonth% | Regen des aktuellen Monats |

Ein Beispiel soll dies zeigen:

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|---------------------------------|---|
| Regen aktuelle Stunde: %rain1h% | Regen aktuelle Stunde: 1,8 l/m ² |
| Regen heute: %rainday% | Regen heute: 4,9 l/m ² |
| Regen diese Woche: %rainweek% | Regen diese Woche: 12,3 l/m ² |
| Regen diesen Monat: %rainmonth% | Regen diesen Monat: 24,8 l/m ² |

Dieses Beispiel soll außerdem verdeutlichen, dass hier der Wert der Variablen **unabhängig** vom eingestellten **Betrachtungszeitraum** ermittelt wird.

Dies alles gilt sinngemäß für ähnlich strukturierte Variablen wie

- %ws_baro1h% --> Luftdrucktendenz der letzten Stunde
- %ws_icedays_m% --> Eistage im aktuellen Monat

Hier ist dann immer im Namen der Variablen ein Hinweis auf ihren Zweck enthalten:

- %ws_baro**1**h% --> lh für letzte Stunde
- %ws_icedays_**m**% --> m für Monat

Eine gewisse **Sonderstellung** nehmen die Langzeitvariablen ein, da sie zwar **unabhängig** vom eingestellten Betrachtungszeitraum sind, aber **abhängig** von der Einstellung im Programm im Menüpunkt "Wetter - Statistikdaten - Langzeitwerte".

Ist hier unter "Aufzeichnungsbeginn" das aktuelle Jahr (z.B. 2003) eingegeben, so werden alle Werte von 2003 ermittelt.

Ist dagegen bei "Aufzeichnungsbeginn" **2002** eingegeben, so werden die Werte aus den Jahren **ab** 2002 ermittelt.

| Inhalt der custom.txt | erzeugt | Inhalt der erzeugten longtime.html |
|-----------------------|---|------------------------------------|
| %LT_TempMax% | Maximaltemperatur von Sensor 1 im ganzen Jahr | 33,4 °C |
| %LT_BaroMax% | maximaler Luftdruck im ganzen Jahr | 1033 hPa |

Wiederum eine andere Art von Variablen sind diejenigen, die keinen Zahlenwert, sondern **Text** erzeugen, wie auf der [Seite "Variablen zur Text- u. Bildausgabe"](#) gezeigt wird.

Variablen zur Text- und Bildausgabe

Den bisher erwähnten Variablen war eines gemeinsam:
Sie erzeugen einen **zahlenmäßigen Wert** (Temperatur, Luftdruck usw.)

Es gibt aber auch Variable, die **Text** erzeugen.

| Variablenname | erzeugt |
|----------------------|-----------------------|
| %longslabel[x] | Sensorname |
| %monthtxt% | aktueller Monatsname |
| %ws_barotendencytxt% | Luftdrucktendenz-Text |

Konkret:

| Inhalt der custom.txt | Inhalt der erzeugten custom.html |
|--------------------------|---|
| %longslabel[2] | Temperatur 1 |
| %monthtxt% | Februar (oder anderer aktueller Monat) |
| %ws_barotendencytxt % | fallend (oder je nach Luftdruckverhalten "steigend - gleichbleibend" usw.) |

Dies schafft die Möglichkeit, in selbst erzeugten Dateien Beschriftungen **aktuell** zu erzeugen. Diese beiden Beispiele stehen nur stellvertretend für viele weitere Variable, die zur Verfügung stehen.

Eine weitere, sehr nützliche Art von Variablen sind diejenigen, die eine **Bildausgabe** ermöglichen. Hier ist der Zusammenhang allerdings etwas komplexer.

| Variablenname | erzeugt | mögliche Dateien |
|--------------------|---|---|
| %windd_gra% | Namen der Bilddatei für die Windrichtung (Standard v. WSWIN) | e.gif / e-ne.gif / o.gif usw. |
| %winddrange_gra% | Namen der Bilddatei für die Windrichtung mit Schwankung für die Icons v. Marcus Pautsch | O1.gif / O2.gif / O3.gif für Wind aus Ost mit verschiedenen Schwankungen (entsprechend auch für die anderen Richtungen) |
| %thermicstressgif% | Namen der Bilddatei für "persönliches Wohlbefinden" | thermic.gif / thermic1.gif usw. |
| %ws_baro_icon% | Namen der Bilddatei für die Luftdrucktendenz | p_do.gif / p_up.gif usw. |

Was in diesem Zusammenhang häufig zu Missverständnissen führt:

Wie in der Tabelle gezeigt, erzeugt ein- und dieselbe Variable (z.B. %windd_gra%) **unterschiedliche** Dateinamen, in diesem Beispiel je nach Windrichtung die Dateien e.gif, s.gif, w.gif.n.gif und andere.

Nun muss dafür gesorgt werden, dass in der custom.txt das dem Namen entsprechende Bild (icon) geladen wird.

Dies geschieht folgendermaßen:

| Variablenname | erzeugt | wie wird das dazugehörige Bild eingefügt? |
|----------------|----------|---|
| %ws_baro_icon% | p_do.gif | |

Erläuterung:

Sie fügen in Ihrer custom.txt **nicht** %ws_baro_icon% ein, sondern ****

Wenn die custom.txt dann abgearbeitet wird, steht im Code der custom.html z.B. oder und damit ist das Problem gelöst: **Je nach Luftdrucktendenz wird ein unterschiedlicher Pfeil angezeigt**

| | |
|--------------|--------------------------|
| für p_do.gif | <input type="checkbox"/> |
| für p_up.gif | <input type="checkbox"/> |

Das hier Gesagte gilt sinngemäß für alle gleichartigen anderen Variablen.

Um beim Beispiel der Windrichtungsgrafiken zu bleiben: Je nachdem, welche Art von Windrichtungsgrafik verwendet wird, müssen natürlich auch die **passenden** Icons im HTML-Verzeichnis vorhanden sein:

für

| | |
|------------------|--|
| %windd_gra% | e.gif / e-ne.gif / o.gif. usw |
| %windd10_gra% | wind_00.gif, wind_01.gif bis wind_36.gif |
| %windd22_gra% | wind00.gif, wind01.gif bis wind16.gif |
| %winddrange_gra% | N0.gif, N1.gif, N2.gif, N3.gif für Nordrichtungen und entsprechend für alle anderen Richtungen |

Hinweisen möchte ich noch auf die (ganz tolle) Möglichkeit, mit Hilfe der Variablen eigene Namen für Dateien zu bilden.

Beispiel:

Sie brauchen fürs Internet eine Datei, die ihren Namen dem Monat anpasst, z.B. m200302.jpg (m= Monatsdatei; Jahr= 2003; Monat =Februar).

Kein Problem, denn sie bilden den Namen so:

m%ws_year%%ws_month2%.jpg

und bauen diese "Variable" in die custom.txt ein.

Bitte beachten: Die Variablennamen sind oft sehr ähnlich, hier wird beispielsweise **nicht** %ws_month% verwendet, sondern eben **%ws_month2%**, wodurch der Monatsname zweistellig ausgegeben wird, wie man es hier eben braucht.

Beim Abarbeiten der custom-Datei ersetzt WSWIN dann **%ws_year%** durch **2003** und **%ws_month2%** durch **02**, wodurch ein Dateiname m200302.jpg entstanden ist.

Auf ähnliche Weise funktioniert die (seit Version 2.82.0) neue Variable **%ws_yearfolder%**, die zusätzlich zum Jahresnamen auch noch den slash (/) erzeugt. Hier wird beim Abarbeiten dann beispielsweise **%ws_yearfolder%** durch **2003/** ersetzt.

Doch dies ist schon fast für "Spezialisten" gedacht.

Damit sind der Gestaltung einer eigenen custom.txt und damit custom.html fast keine Grenzen gesetzt.

Auf einige Besonderheiten soll in der nachfolgenden Seite noch hingewiesen werden:

[Variablen mit Benutzervorgabe](#)

Variablen mit Benutzervorgabe

Eine Sonderstellung nehmen die Variablen mit Benutzervorgabe ein, da sie, wie schon der Name sagt, in ihrer Ausgabe nicht festgelegt sind, sondern der **Benutzer bestimmt, was** die Variable bei bestimmten Gelegenheiten anzeigen soll.

Ein Beispiel mag dies zeigen:

Es gibt eine Variable, die so aussieht:

```
%ws_isdaynight=Tag,Morgendämmerung,Abenddämmerung,Nacht%
```

Hier ist entscheidend, dass die Variable aus zwei Teilen besteht:

- aus dem Variablennamen (ws_isdaynight)
- den sog. Parametern "Tag,Morgendämmerung,Abenddämmerung,Nacht" (hier **vier**)

Was geschieht nun, wenn diese Variable abgearbeitet wird?

Je nach Tageszeit wird eine **unterschiedliche** Textmeldung ausgegeben:

Tag, Morgendämmerung, Abenddämmerung, oder Nacht

Nun kommt der entscheidende Punkt:

Diese vier Angaben können vom Benutzer **selbst** gestaltet werden, also beispielsweise:

```
%ws_isdaynight=guten Tag,es dämmt gerade,bald wird es Zeit fürs Bett,gute Nacht%
```

Wenn nun die Datei gerade abends um 23:00 Uhr abgearbeitet wird, erscheint auf dem Bildschirm "gute Nacht", wird sie dagegen vormittags um 10:00 Uhr erstellt, erscheint "guten Tag".

Zusammenfassend kann man es sich so vorstellen:

| Ursprungscode | vom Benutzer veränderter Code | Ergebnis (abhängig v. Situation) |
|--|--|--|
| %ws_isdaynight=Tag,Morgendämmerung,Abenddämmerung,Nacht% | %ws_isdaynight=guten Tag,hallo Morgenmuffel,Kinder jetzt ins Bett,eine gute Nacht% | hallo Morgenmuffel |
| %ws_icewarning=Ja,Nein% | %ws_icewarning=Glatteis möglich,keine Glatteisgefahr% | keine Glatteisgefahr |

Es gibt hier noch einige **wesentliche** Dinge zusätzlich zu beachten (Komma als Trennzeichen erforderlich - Parameter können auch htmCode sein usw.), bitte hier **dringend die Hilfedatei des Programms ansehen**.

Dennoch soll auf die zwei häufigsten Fehler bereits hier hingewiesen werden:

- Parameter**zahl** stimmt nicht (im Beispiel ws_isdaynight **müssen** es **vier** sein)
- die Parameter sind nicht durch Komma getrennt

Seit der Version 2.8.3.0 gibt es neue Variablenarten, die die Möglichkeiten des Programms ganz entscheidend erweitern.

Es sind dies Variablen

- zur (benutzerdefinierten) **Berechnung**
- zum **Vergleich** zweier Werte und Lieferung des Ergebnisses
- Variablen mit "**Speicherfunktion**"

Dies soll in der nachfolgenden Tabelle verdeutlicht werden:

Hier ist zu beachten, dass als Trennzeichen der Parameter die Tilde (~) verwendet wird.

| Variable zur Berechnung | vom Benutzer einstellbar | Bedeutung |
|---|--|---|
| %ws_calc[x]=w1~w2~Dez% oder %ws_calc1[x]=w1~w2~Dez% | Für den Parameter x können nun Rechenoperationen gewählt werden: [+] oder [-] oder [*] oder [/] | w1 = 1.Wert, w2 = 2.Wert Dez= Zahl der Kommastellen Dabei können w1 und w2 entweder echte Zahlen oder Variable sein |

Wozu ist die 2. Variable ws_calc1 ?

Sie wird benötigt, wenn in einer Zeile **zwei** Berechnungen erfolgen sollen, also z.B. Bildung einer Summe **und** dann die Durchschnittsermittlung. Ich verwende im Nachfolgenden ein Beispiel aus der Hilfedatei des Programms:

| was soll erreicht werden | Variable | Erklärung |
|---|--|--|
| aus zwei Temperaturen (Sensor 1 und Sensor 2) soll zuerst die Summe gebildet und dann der Durchschnitt errechnet werden | %ws_calc1[/]= %ws_calc[+]= %curval[2]~ %curval[3]~3%~2~3% | roter Teil: <ul style="list-style-type: none"> die Temperaturen 1 und 2 werden zusammengezählt und mit 3 Dezimalstellen berechnet grüner Teil: <ul style="list-style-type: none"> das Ergebnis wird durch 2 geteilt und mit drei Dezimalstellen ausgegeben |

praktisches Beispiel

ergibt

%ws_calc[-]=%curval[2]~%curval[3]~2%

Temperaturdifferenz von Aussensensor 1 und Aussensensor2 auf 2 Dezimalstellen genau
wenn curval[2]=15°C und curval[3]=12°C erscheint als Ergebnis 3,00

Nutzen dieser Variablenart: Aus zwei Variablen wird ein dritter Wert errechnet und ausgegeben.

| Variable zum Vergleich | vom Benutzer einstellbar | Bedeutung |
|--|---|--|
| %ws_compare[x]=w1~w2~richti g~falsch% | Hier bedeutet: x=1: Wert1 > Wert2 x=2: Wert1 >= Wert2 x=3: Wert1 = Wert2 x=4: Wert1 <= Wert2 x=5: Wert1 < Wert2 x=6: Wert1 <> Wert2 | Wenn der Vergleich zutrifft, wird der Text ausgegeben, der bei " richtig " steht, ist der Vergleich nicht zutreffend, wird der Text bei " falsch " ausgegeben. |
| praktisches Beispiel | | ergibt |

| | |
|---|--|
| <pre>%ws_compare[1]=%curval[2]%-~%curval[3]%-Temp. AußenNord ist höher als AußenSüd~Temp. AußenNord ist niedriger als AußenSüd%</pre> | <pre>bei TempNord 7,5°C und TempSüd 6,1°C -- > Temp. AußenNord ist höher als AußenSüd bei TempNord 4,5°C und TempSüd 5,1°C -- > Temp. AußenNord ist niedriger als AußenSüd</pre> |
|---|--|

Nutzen dieser Variablenart: zwei Variablen werden miteinander verglichen und je nach Vergleich wird unterschiedlicher Text ausgegeben.

Bitte beachten: Wert1 und Wert2 müssen **Zahlenwerte** darstellen (echte Zahlen oder Variable, aber keinen Text)

Ausnahme: Der Vergleich auf "ungleich" (x=6) erlaubt für den 2. Parameter auch andere Werte (z.B. Text), doch Parameter 1 muss eine Zahl sein (z.B. -255)

| Variable zur Speicherung | vom Benutzer einstellbar | Bedeutung |
|---|---|---|
| <pre>%ws_setmem[x]=zzzzz%</pre> | <p>Für den Parameter x können Werte von 1 - 32 gewählt werden</p> | <p style="color: #008000;">zzzzz können Variable oder auch Text sein. Diese Variable macht nur Sinn im Zusammenhang mit der Variablen %ws_getmem[x]%</p> |
| praktisches Beispiel | | ergibt |
| <pre>%ws_setmem[1]=heute scheint die Sonne%</pre> | | |
| <pre>%ws_getmem[1]%</pre> | | heute scheint die Sonne |

Hier stehen dem Benutzer insgesamt **32 Speichervariablen** zur Verfügung.

Nutzen diese Variablenart: Es kann also jede Menge Variablen (vor allem aus templates) zwischengespeichert werden und in einer anderen Datei (z.B. custom.txt) ausgegeben werden. Dazu gibt es noch eine Besonderheit:

| | | |
|-----------------------------|---|--|
| <pre>%ws_getmem_yest%</pre> | <p>entsprechend steht die Variable für Woche, Monat, Jahr zur Verfügung , z.B. %ws_getmem_y%</p> | <p>es wird abgefragt, ob das Yesterdaytemplate (template_yest.txt) seit Programmstart bereits abgearbeitet wurde</p> |
|-----------------------------|---|--|

Weitere Anwendungsbeispiele zu dieser neuen Variablenart sind [hier](#) zu finden.

Zusammenfassung

Die vorliegenden Seiten sind nur eine Zusammenfassung der allerwichtigsten Punkte. Hier wurde beispielsweise **nicht** auf Variable eingegangen, die

- besonders für die Verwendung in den sog. Template-Dateien geeignet sind (z.B. %ws_avgmintemp%)
Dies soll im nächsten Menüpunkt "*Anwendungshilfen*" geschehen
- als "Steuervariablen" eingesetzt werden (z.B. %unit_off%)

Wer tiefere Einblicke gewinnen will, kommt nicht darum herum, sich mit der Hilfe-Datei des Programms (Stichwort "Variablen") auseinanderzusetzen.

Ergänzend möchte ich auf zwei Links zu PC-Wetterstation hinweisen:

- [eine Zusammenfassung aller Variablen mit Erläuterung](#)
- [die Ergebnisse daraus \(ist natürlich bei Variablen mit Index nicht möglich\)](#)

Wer nun Lust bekommen hat, sich noch etwas tiefer einzuarbeiten oder mit dem Sinn verschiedener Variablen nicht ganz klar kommt, sollte den letzten Menüpunkt "**Anwendungshilfen**" durcharbeiten.

Ansonsten hoffe ich, mit dieser Zusammenstellung den Einblick in die Verwendung der Variablen von WSWIN etwas vertieft zu haben.

Anwendungshilfen

Es soll hier der Versuch unternommen werden, für Standardanwendungen einige Beispiele zusammenzustellen, damit leichter entschieden werden kann, welche Variable in Frage kommt. Dies soll aber **keine Einweisung in die Erstellung von Template-Dateien darstellen**.

Das Problem ist, dass manche Variable nur in der **custom.txt** Sinn machen, andere dagegen nur in den sog. **Template-Dateien** (Monatstemplate - Jahrestemplate).

Damit ist folgendes gemeint:

Eine Variable wie %ws_avgmaxtemp% bringt zwar auch in der custom.txt einen Wert, dieser bezieht sich aber **nicht** wie gewünscht auf den Monat, ist also hier letztlich sinnlos.

Hier ist daher immer darauf zu achten, ob in der **Hilfdatei des Programms** angegeben ist *"Variable nur bei Woche - Monat - Jahr gültig"*. Diese Variablen dann **nicht** in der custom.txt verwenden.

Was sind nun einige wichtige Variablen, die in der **custom.txt** verwendet werden können?

Hier können nun nicht alle Möglichkeiten aufgezählt werden, denn was für einen Maximalwert gilt, gilt entsprechend natürlich auch für den Minimalwert. Die Beispiele sollten also exemplarisch verstanden werden.

| Absicht | Variable | Anmerkung |
|--|---|---|
| aktueller Wert der Außentemperatur | %curval[x]% | x entspricht in allen Beispielen der Sensor ID |
| aktueller Wert der Außentemperatur | %curvaloutstemp% seit Version 2.8.4.6 auch %curval[0]% | hier ist keine Sensorangabe erforderlich, denn es wird der Sensor genommen, der für den Taupunkt angegeben ist |
| Maximalwert der Außentemperatur | %maxval[x]% | ergibt unterschiedliche Werte, je nachdem ob die Tages- oder 24h-Ansicht gewählt ist |
| Maximalwert der Außentemperatur | %maxvaloutstemp% seit Version 2.8.4.6 auch %maxval[0]% | ergibt unterschiedliche Werte, je nachdem ob die Tages- oder 24h-Ansicht gewählt ist, aber vom Wert des Taupunktsensors |
| Maximalwert der Außentemperatur | %maxtempday% | nimmt den Wert des Taupunktsensors und ist auf die Tagesansicht bezogen |
| Aus dem oben Beschriebenen sollte ersichtlich sein, dass es mehrere Variable gibt, die denselben Zweck erfüllen (z.B. maximale Außentemperatur), aber unter verschiedenen Voraussetzungen (einmal wird der Taupunktsensor verwendet, ein anderes Mal wird der Sensor durch die ID gewählt) | | |
| entsprechendes gilt für die "Zeitwerte" des Maximums | a) %maxtime[x]%, b) %maxtimecurday[x]%, c) %maxtempdaytime%, d) %maxtimeoutstemp% seit Version 2.8.4.6 auch %maxtime[0]% | alle bringen die Zeit der Maximaltemperatur, aber a) Sensorwahl und je nach gewählter Ansicht den Tageswert oder 24h-Wert b) Sensorwahl und unabhängig von der Ansicht den Tageswert c) unabhängig von der Ansicht den Tageswert , aber den des Taupunktsensors d) nimmt Taupunktsensor und ist abhängig von der Ansicht (Tag/24h) |
| Hinweis zur Verwendung der "outstemp" und "outshum"-Variablen | wie bereits oben gezeigt, kann die Variablenart "outstemp" (z.B. %curvaloutstemp% durch %curval[0]%) ersetzt werden | entsprechendes gilt für die Variablenart "outshum", bei der jedoch der Parameter "-1" gilt, also anstatt %curvaloutshum% gilt nun %curval[-1]% |
| Zuletzt soll noch auf eine Besonderheit der Variablen curminmaxhour eingegangen werden. Diese ist auch bei den "Variablenarten" eingehend beschrieben. | | |

| | | |
|---|------------------------|--|
| Minimalwert der letzten 5 Stunden von Sensor 2 | %curminmaxhour[1,5,2]% | hier kann also die Art des Wertes (aktuell, Minimum usw) , die Zahl der zu berücksichtigenden Stunden und die Sensornummer angegeben werden |
| Minimalwert des ganzen aktuellen Tages von Sensor 2 | %curminmaxhour[1,0,2]% | Hier werden also, ohne Berücksichtigung einer festen Stundenzahl, die Werte des ganzen aktuellen Tages berücksichtigt |

Eine ganze Reihe von Variablen macht dagegen nur Sinn, wenn sie in den sog. **Monatstemplates** (template_m.txt) oder **Jahrestemplates** (template_y.txt) verwendet werden. Nun kann es nicht Sinn dieses Artikels sein, hier eine Einführung in die Verwendung von Templates zu geben. Doch es soll versucht werden zu zeigen, was mit einigen Variablen in den Templates erreicht werden kann bzw. wie sie sich von der Verwendung in der custom.txt **unterscheiden**.

| Absicht | Variable | Anmerkung |
|---|--|---|
| Durchschnittstemperatur jedes einzelnen Tages eines ganzen Monats | %curval[x]% | Was hier oft für Verwirrung sorgt: curval[x] bringt in der <i>custom.txt</i> den aktuellen Wert (z.B. um 15 Uhr), in der Monats-Templatedatei jedoch den Durchschnittswert des betreffenden ganzen Tages im Monat |
| Maximalwert des jeweiligen Tages im Monat | %maxvaldayoutstemp% seit Version 2.8.4.6 auch %maxvalday[0]% | Es wird hier die höchste, am jeweiligen Tag des Monats aufgetretene Temperatur angezeigt |
| Durchschnittstemperatur aller Tage des ganzen Monats | %avgval[x]% | avgval[x] bringt hier nicht den Durchschnittswerte des Tages (wie in der <i>custom.txt</i>), sondern den des Monats |
| durchschnittliche Maximaltemperatur eines Monats | %ws_avgmaxtemp% | hier wird der Schnitt aller Maximaltemperaturen gebildet |
| <p>Wie man aus dem oben Beschriebenen sieht, kommt es bei diesen Variablen darauf an, - in welchem Zusammenhang sie verwendet werden: custom.txt oder Monatstemplate (template_m.txt) --> Beispiel 1 und 3 oder - dass sie nur im Monatstemplate Sinn machen (Beispiel 2 und 4) Ferner sollte klar sein, dass dies, was ich hier für die Maximalwerte angegeben habe, entsprechend auch für die Minimalwerte gilt.</p> | | |
| Maximaltemperatur des ganzen Monats | %maxvaloutstemp% seit Version 2.8.4.6 auch %maxval[0]% | hier wird als Sensor wiederum der Sensor für den Taupunkt genommen |
| Maximaltemperatur des ganzen Monats | %maxval[x]% | hier wird als Sensor jeder beliebige Sensor genommen |
| Regen im Monat | %rainmonth% | All diesen Sensoren wiederum ist gemeinsam, das sie sowohl in der custom.txt als auch im Monatstemplate verwendet werden können und gültige Werte bringen |
| Regentage > 0 mm im Monat | %rain0month% | |
| Sommertage im Monat | %ws_warmdays_m% | |

Unter dem folgenden Link kann in einer Monatstabelle, die mit einer Datei template_m.txt erzeugt wird, betrachtet werden, welche Variablen für welchen Wert "zuständig" sind.

[Monatsübersicht1](#) und [Monatsübersicht2](#)

Wer sich die Datei zum Ansehen laden will, klickt hier: [Monatstemplate](#)

Bitte beachten: Die Datei wurde von mir unter dem Namen **template_mMuster.zip** gespeichert.

Wer sie nicht nur ansehen, sondern versuchsweise verwenden will, muss sie natürlich nach dem Laden in `template_m.txt` umbenennen (eine bereits vorhandene `template_m.txt` zuvor **sichern**). Diese Template-Datei erzeugt dann eine HTML-Datei namens **tabelle.html**

An dieser Stelle sollen einige Beispiel zu den seit **Version 2.8.3.0 neu hinzugekommenen Variablen** gebracht werden:

Variable zur Berechnung

| Absicht | Variable | Anmerkung | Ergebnis |
|---|--|---|---|
| Beispiele zu Variablen mit Berechnung : <code>%ws_calc[x]=w1~w2~Dez%</code> | | | |
| Windgeschwindigkeit von km/ in m/s umrechnen | <code>%ws_calc[/]= %curval[35]~3.6~3%</code> | <ul style="list-style-type: none"> • <code>curval[35]</code> liefert die aktuelle Windgeschwindigkeit in km/h • Dieser Wert wird durch 3.6 geteilt • das Ergebnis wird mit einer Genauigkeit von 3 Stellen ausgegeben | bei <code>curval[35]=65</code> km/h ausgegeben wird 8.055 |
| Temperaturdifferenz zweier Aussensensoren berechnen | <code>%ws_calc[-]= %curval[2]~% %curval[3]~1%</code> | <ul style="list-style-type: none"> • <code>curval[2]</code> liefert Temp. Sensor 1 • <code>curval[3]</code> liefert Temp. Sensor 2 • die Werte werden subtrahiert • das Ergebnis wird mit einer Kommastelle ausgegeben | bei <code>curval[2]=18</code> und <code>curval[3]=16</code> ausgegeben wird 2.0 |
| Regentage in Prozent berechnen beachten: Hier werden ws_calc1 und ws_calc verwendet | <code>%ws_calc1[*]= %ws_calc[/]=%raindays %~%ws_dayyear %~3%~100~2%</code> | <ul style="list-style-type: none"> • <code>raindays</code> liefert die Zahl der Regentage • <code>ws_dayyear</code> liefert die Zahl der Tage des Jahres bis jetzt • <code>%ws_calc[/]</code> dividiert die Regentage durch die Gesamtage des Jahres • <code>%ws_calc1[*]</code> multipliziert dann mit 100 | bei <code>raindays = 60</code> und <code>ws_dayyear = 144</code> ausgegeben wird 41,66 |
| Inhalt der custom.txt | | Ausgabe in custom.html | |
| Windgeschw.: <code>%ws_calc[/]= %curval[35]~3.6~3%</code> m/s | | Windgeschw.: 8.055 m/s | |
| Temperaturdifferenz Nord/Süd: <code>%ws_calc[-]= %curval[2]~%curval[3]~1%</code> Grad | | Temperaturdifferenz Nord/Süd: 2.0 Grad | |

Regentage in Prozent: %ws_calc1[*]=
%ws_calc[/]=%raindays%~%ws_dayyear
%~3%~100~2%

Regentage in Prozent: 41,66

Variable zum Vergleich

| Absicht | Variable | Anmerkung | Ergebnis |
|--|--|---|--|
| Beispiele zu Variablen mit Vergleich: <code>%ws_compare[x]=w1~w2~richtig~falsch%</code> | | | |
| Temperaturdifferenz zweier Sensoren soll in Worten bewertet werden | <code>%ws_compare[5]= %curval[2]~ %curval[3]~Temp. AußenNord ist niedriger als AußenSüd~Temp. AußenNord ist höher als AußenSüd%</code> | <ul style="list-style-type: none"> • curval[2] liefert die Temp. von Sensor 1 • curval[3] liefert die Temp. von Sensor 2 • beide Werte werden auf kleiner verglichen (Sensor 1 < Sensor 2) | bei curval[2]=18 und curval[3]=20 ausgegeben wird Temp. AußenNord ist niedriger als AußenSüd bei curval[2]=21 und curval[3]=19 ausgegeben wird Temp. AußenNord ist höher als AußenSüd |
| Es soll ausgegeben werden, ob heute schon Niederschlag fiel | <code>%ws_compare[1]= %rainday%~0~heute bereits Niederschlag~heute noch kein Niederschlag%</code> | <ul style="list-style-type: none"> • rainday liefert den Tagesniederschlag • dieser Wert wird auf größer Null verglichen | bei rainday=5 ausgegeben wird heute bereits Niederschlag bei rainday=0 ausgegeben wird heute noch kein Niederschlag |
| Es soll ausgegeben werden, ob heute schon Niederschlag fiel, aber mit Angabe des Niederschlags | <code>%ws_compare[1]= %rainday%~0~heute bereits %rainday% mm Niederschlag~heute noch kein Niederschlag%</code> | es erfolgt derselbe Vorgang wie im obigen Beispiel, jedoch wird bei einem Parameter nicht nur Text , sondern auch eine Variable ausgegeben | ausgegeben wird heute bereits 5 mm Niederschlag |
| Es soll die momentane Windgeschwindigkeit ausgegeben werden und bei Windstille ein Text | <code>%ws_compare[1]= %curval[35]~0~Windges chwindigkeit. %curval[35]~ %unitname[35]~Moment an Windstille.%</code> | es soll die momentane Windgeschw. ausgegeben werden oder bei Windstille ein Text | bei 0 km/h ausgegeben wird Momentan Windstille |

Wir ersehen daraus:

- für die Parameter w1 und w2 können Variable eingesetzt werden, es kann also eine Variable mit einer **anderen Variablen** verglichen werden
- für die Parameter w1 und w2 können Variable und echte Werte (Zahlen) eingesetzt werden, es kann also eine Variable mit **einer Zahl** (z.B. dem Wert Null) verglichen werden
- vor Verwendung dieser Variablen sollte die Ausgabe der Einheiten mit %unit_off% ausgeschaltet werden, also z.B. %unit_off% %ws_compare[1]=%rainday%~0~heute bereits Niederschlag~heute noch kein Niederschlag%

Eine gewisse Sonderstellung nimmt die "Speichervariable" ws_setmem[x] ein.

| Absicht | Variable | Anmerkung | Ergebnis |
|--|--|---|---|
| Beispiele zu Variablen mit Speicherung : %ws_setmem[x]% und %ws_getmem[x]% | | | |
| Die maximale durchschnittliche Monatstemperatur soll ausgegeben werden. Die notwendige Variable, nämlich %ws_avgmaxtempmonth% macht jedoch nur in einer Template-Datei Sinn. | In die Monats-Template-Datei (template_m.txt) wird diese Variable eingebaut: %ws_setmem[1]= %ws_avgmaxtempmonth% % | Damit die Variable zugewiesen wird, muss natürlich zuerst die Monats-Template-Datei abgearbeitet werden. Dadurch wird der Wert von %ws_avgmaxtempmonth% in der Variablen %ws_setmem[1]% gespeichert | der Wert einer Template-Variablen wird an %ws_setmem[1]% übergeben, dort gespeichert und kann in einer anderen Datei (hier custom.txt) verwendet werden |
| Der Wert soll aber nicht in der Template-Datei, sondern in der custom.txt verwendet werden | Das steht dann in der custom.txt: Die durchschnittliche monatliche Maximaltemperatur beträgt %ws_getmem[1]% Grad | jetzt wird sie in der custom.txt eingesetzt | mit der Variablen %ws_getmem[1]% wird der Wert ausgegeben |
| Eine Sonderstellung nimmt die Variable %ws_getmem_X% ein, da sie nicht gesetzt werden muss, sondern bei der Abarbeitung eines Templates automatisch gesetzt wird und dann abgefragt werden kann. | | | |
| Es soll abgefragt werden, ob das yesterday-template abgearbeitet wurde | %ws_getmem_yest% | wenn es nicht abgearbeitet wurde wenn es abgearbeitet wurde | -255 12:21:47 04.12.2003 |
| sinngemäß kann so auch abgefragt werden, ob das Wochentemplate (%ws_getmem_w%), das Monatstemplate (%ws_getmem_m%) oder das Jahrestemplate (%ws_getmem_y%) bereits abgearbeitet wurde | | | |
| Was auch gut ist zu wissen: Die Speichervariablen werden mit -255 initialisiert, d.h. vor irgendeiner Zuweisung haben sie den Wert -255 | %ws_compare[6]=-255~%ws_getmem[1]~Variable ist gesetzt~Variable ist nicht gesetzt% | wenn sie gesetzt wurde wenn sie noch nicht gesetzt wurde | Variable ist gesetzt Variable ist nicht gesetzt |

Unter den folgenden zwei Links kann eine Textdatei betrachtet werden und wie die aus ihr erzeugte html-Datei aussieht.:

[neuvar.txt](#) und [neuvar.html](#)

Auf die zwei **häufigsten Fehlerquellen** soll hingewiesen werden:

- die Zahl der "%" stimmt nicht, z.B. `%ws_setmem[1]=%ws_avgmaxtempmonth%` --> es fehlt ein %-Zeichen
 - das Trennungszeichen zwischen den Variablen (~) wurde vergessen oder steht an falscher Stelle
 - am Anfang fehlt %unit_off% um die Einheiten auszuschalten
 - Zuweisung und Abfrage erfolgen in derselben Zeile, hier besser eine **neue** Zeile beginnen
-

Weitere praktische Beispiele

In dieser letzten Folge von Tabellen werden einige Variablenanwendungen vorgestellt, die **Kombinationen** aus den verschiedenen Möglichkeiten aufweisen. Diese Beispiele wurden teilweise auch von Usern aus dem Forum entwickelt.

| Absicht | Variable | Anmerkung |
|---|--|--|
| <p>auf Grund der Regenwerte und der Temperatur soll entschieden werden, ob Schnee oder Regen fällt</p> <p>Bedingung: Niederschlag und weniger als 1.3°C = Schnee, ansonsten Regen</p> | <pre>%unit_off% %ws_setmem[1]=%ws_compare[1]= %curval[34]%-0~5~1% %</pre> | wenn es geregnet hat, enthält setmem[1] den Wert 5 (5 und 1 sind willkürlich gewählt) |
| | <pre>%ws_setmem[2]=%ws_compare[5]= %curval[22]%-1.3~5~4% %</pre> | wenn es kälter als 1.3 °C ist, enthält setmem[2] den Wert 5 (5 und 4 sind willkürlich gewählt) |
| | <pre>%ws_setmem[3]=%ws_compare[3]=%ws_getmem[1]%- %ws_getmem[2]%- momentaner Schneefall~ momentaner Regen% %</pre> | die beiden Variablen werden verglichen. Wenn beide den Wert 5 enthalten (Niederschlag und kälter als 1,3°C), wird "momentaner Schneefall" ausgegeben, ansonsten "momentaner Regen" |
| | <p>Wem dies genügt, der kann jetzt mit %ws_getmem[3]% den Text ausgeben. Beispielsweise: <i>Augenblicklich</i> <pre>%ws_getmem[3]%</pre> Wer eine differenziertere Ausgabe möchte, setzt die folgenden Zeilen noch ein:</p> | |
| | <pre>%ws_setmem[5]=%ws_compare[2]= %curval[34]%-0.360~3~5% %</pre> | es wird verglichen, ob mehr als 0.36 l/m ² gefallen sind (Parameter 3 und 5 sind willkürlich gewählt) |
| | <pre>%ws_setmem[6]=%ws_compare[2]= %curval[34]%-0.7~3~4% %</pre> | es wird verglichen, ob mehr als 0.7 l/m ² gefallen sind (Parameter 3 und 4 sind willkürlich gewählt) |
| | <pre>%ws_setmem[7]=%ws_compare[3]=%ws_getmem[5]%- %ws_getmem[6]%-mäßiger~leichter% %</pre> | wenn in getmem[5] und getmem[6] der gleiche Wert 3 ist, haben wir mäßigen, sonst leichten Niederschlag |
| | <pre>%ws_setmem[8]=%ws_compare[4]=%curval[34]%-1.08~ %ws_getmem[7]%-starker% %</pre> | wenn der aktuelle Niederschlag <= 1.08 l/m ² ist, wird der Wert von getmem[7] in setmem[8] gemerkt, ansonsten das Wort "starker" |
| | <pre>%ws_compare[1]=%curval[34]%-0~%ws_getmem[8]%- %ws_getmem[3]%-trocken% %</pre> | hier erfolgt die endgültige Ausgabe: der aktuelle Niederschlag (curval[34]) wird mit "0" verglichen. Wenn es Niederschlag gab, erfolgt die entsprechende Bemerkung, ansonsten das Wort "trocken" |

| | | |
|---------------------------|--|---|
| Zusatzanmerkung | Es werden also folgende Fälle unterschieden: kein Niederschlag --> trocken bis 0.36 l/m ² und Temperatur über 1,3°C --> leichter momentaner Regen bis 0.7 l/m ² und Temperatur über 1,3°C --> mäßiger momentaner Regen über 1.08 l/m ² und Temperatur über 1,3°C --> starker momentaner Regen bis 0.36 l/m ² und Temperatur unter 1,3°C --> leichter momentaner Schneefall bis 0.7 l/m ² und Temperatur unter 1,3°C --> mäßiger momentaner Schneefall über 1.08 l/m ² und Temperatur unter 1,3°C --> starker momentaner Schneefall Diese Variablenfolge wurde vom User "adris" entwickelt. | Dabei sollte folgendes klar sein: die Temperaturgrenze von 1,3°C und die Regenmengen wurden willkürlich gewählt und können natürlich verändert werden |
| Dringend beachten: | wenn zwei Prozentzeichen in den Variablen aufeinanderfolgen, dazwischen keine Leerstelle lassen wie in den obigen Beispielen (ich muss dies wegen meines Editors machen) | |

Eine interessante **Variante** (die sich erst in der Praxis noch bestätigen muss), geht davon aus, dass es dann schneit, wenn die **Summe aus Temperatur und Taupunkt ≤ 0 ist** und dass Schneeregen bei ≤ 0.5 °C zu erwarten ist, ansonsten Regen.

| Variable | Anmerkung | |
|--|---|---|
| %unit_off% %ws_setmem[1]=%ws_compare[1]=%curval[34]%-0~5~1% % | wenn es geregnet hat, enthält setmem[1] den Wert 5 | |
| %ws_setmem[9]=%ws_calc[+]=%curval[2]%-~%curval[43]~1% % | Summe aus Temperatur und Taupunkt als Maßstab für Prognose | |
| %ws_setmem[2]=%ws_compare[4]=%ws_getmem[9]%-0.5~5~4% % | wenn obige Summe ≤ 0.5 ist, enthält setmem [10] den Wert 5 | |
| %ws_setmem[10]=%ws_compare[4]=%ws_getmem[9]%-0~5~6% % | wenn obige Summe ≤ 0 ist, enthält setmem [10] ebenfalls den Wert 5 | |
| %ws_setmem[11]=%ws_compare[3]=%ws_getmem[2]%-~%ws_getmem[10]%-Schneefall~Schneeregen% % | wenn setmem[2] und setmem[10] den gleichen Wert (5) enthalten, gibt es Schneefall, sonst Schneeregen | |
| %ws_setmem[3]=%ws_compare[4]=%ws_getmem[9]%-0.5~%ws_getmem[11]%-Regen% % | wenn Summe Temp.+Taupunkt $<$ als 0.5 sind, kommt entweder Schneefall oder Schneeregen, ansonsten Regen | |
| %ws_setmem[5]=%ws_compare[2]=%curval[34]%-0.360~3~5% % | bei den restlichen Variablen gilt Gleiches wie oben | |
| %ws_setmem[6]=%ws_compare[2]=%curval[34]%-0.7~3~4% % | | |
| %ws_setmem[7]=%ws_compare[3]=%ws_getmem[5]%-~%ws_getmem[6]%-mäßiger~leichter% % | | |
| %ws_setmem[8]=%ws_compare[4]=%curval[34]%-1.08~%ws_getmem[7]%-starker% % | | |
| %ws_compare[1]=%curval[34]%-0~%ws_getmem[8]%-~%ws_getmem[3]%-trocken% | Ausgabe des Ergebnisses | |
| Es wird sich zeigen, ob damit (oder durch gewisse Anpassungen), eine einigermaßen zuverlässige Ansage des momentanen Zustands möglich ist. | auch dieser Algorithmus wurde von Adris zusammengestellt | |
| Absicht | Variable | Anmerkung |
| | damit die nachfolgenden Variablen ausgewertet werden können, muss folgende Einstellung in WSWIN getroffen werden: | Wetter - Statistikdaten - Einstellungen: ≤ 19.9 °C einstellen |

| | | |
|---|---|--|
| die Anzahl der tropischen Nächte im Jahr und Vorjahr soll ermittelt werden (entwickelt von Adris) | <pre>%ws_setmem[14]=%ws_calc[-]=%ws_dayyear%~ %ws_uldays_y%~0% % %ws_setmem[15]=%ws_calc[-]= %ws_getmem[14]%~1~0% %</pre> | die aktuelle Tagnummer des Jahres minus die Tage ohne trop. Nacht ergibt die tropischen Nächte; da die Zahl wegen des aktuellen Tages um eins zu hoch ist, wird anschließend der Wert 1 abgezogen |
| | <pre>%ws_setmem[16]=%ws_calc[-]=365~ %ws_ly_uldays_y%~0% %</pre> | wie oben, doch da komplettes Jahr, werden die Tage von 365 subtrahiert (Schaltjahr beachten) |
| | <pre>trop.Nächte: %ws_getmem[15]% trop. Nächte Vorjahr: %ws_getmem[16]%5</pre> | Ausgabe der Werte |

Ein Trost zum Schluss: (sozusagen das "Wort zum Sonntag")

Die Verwendung mancher Variablen (vor allem in den Template-Dateien) ist nicht ganz einfach, also nicht unbedingt für einen gedacht, der das Programm gerade mal eine Woche besitzt.

Denn gerade die Vielzahl an Möglichkeiten sorgt letztlich auch für eine gewisse Verwirrung, wenn man sich noch nicht lange damit beschäftigt hat.

Dafür sind die Möglichkeiten, die sich dadurch bieten, nahezu unerschöpflich.