

Diese Seite richtet sich an alle Benutzer der Wetterauswertesoftware WSWIN
(<http://www.pc-wetterstation.de>)
von Werner Krenn, besonders aber an Neueinsteiger.

Auf den folgenden Seiten soll **nicht** die Arbeit mit WSWIN erklärt werden, sondern

die Erstellung eines

Wetter-Tickers

(Laufband)

mit Hilfe von benutzerdefinierten Dateien.

Ich halte mich dabei an den von Werner Krenn im Userbereich angebotenen Ticker (Davisticker).
Selbstverständlich können andere Java-Applets sinngemäß genauso verwendet werden.

Wie Anfragen im Forum von WSWIN zeigen, ergeben sich vor allem für Einsteiger in diesem
Zusammenhang häufig Fragen:

1. Wie wird ein solcher Ticker überhaupt realisiert?
2. Wie erreiche ich, dass meine in WSWIN verfügbaren Daten dort dargestellt werden?
3. Was muss bei der Übertragung auf die Homepage beachtet werden?

Diese und weitere Fragen sollen hier erörtert werden.
In der linken Navigationsleiste erreichen Sie die dazu notwendigen Seiten.

Letztmals bearbeitet am 23.03.2009

Inhalt

[Grundlagen](#)

[wap.txt](#)

[Ticker einbinden](#)

Grundlegende Anhaltspunkte

Um einen Witterticker zu erstellen, müssen mehrere Vorgänge erfolgen:

1. Es müssen die Wetterdaten, die im Ticker gezeigt werden sollen, zuerst aus WSWIN **gewonnen** werden
Dies geschieht über eine benutzerdefinierte Datei (**wap.txt**)
2. Die mit Hilfe der wap.txt gewonnenen Daten (**data.htm**) müssen an einen Ticker übergeben werden
3. Ein Java-Applet (der eigentliche Ticker) sorgt für die Darstellung der Daten als Laufband

Diese ganzen Zusammenhänge sollen mit Hilfe der Menüpunkte links erläutert werden.

Hier noch ein paar allgemeine Hinweise:

Es könnte zwar ein solcher Ticker auch rein durch html erzeugt werden (Marquee), doch läuft dieser Ticker manchmal recht "ruppig" und vor allem die Einbindung der gewonnenen Daten ist umständlich zu bewerkstelligen. Der Vorteil allerdings wäre, dass er auf vielen verschiedenen Systemen auch tatsächlich läuft.

Mir persönlich erscheint die Lösung mit Hilfe eines Java-Applets als die geeignetere Lösung:

- die Daten können problemlos aus einer externen Datei (data.htm) eingebunden werden
- die Datei, die die Tickerdaten enthält (data.htm) ist recht **klein**
- der Ticker zeigt ein ruckelfreies Laufverhalten

Ein Nachteil soll nicht verschwiegen werden:

Ein User, der Java deaktiviert hat, wird den Ticker auch **nicht** sehen; außerdem gibt es auch Browser, die einen solchen Ticker nicht darstellen.

Hier sollte also jeder entscheiden, was er für richtig hält.

Wie können die Daten für den Ticker gewonnen werden?

Hierzu ein grundlegender Hinweis:

Um benutzerdefinierte Dateien zu erstellen, stehen in WSWIN grundsätzlich die Dateien
custom.txt und wap.txt

zur Verfügung. (nähere Hinweise auch in einer anderen Hilfedatei von mir: [Hinweise zum Umgang mit benutzerdefinierten Dateien](#))

Für die Erstellung der Daten für einen Wetteticker ist **nur** die **wap.txt** geeignet.

Wie ist hier vorzugehen?

Es müssen zwei grundlegende Unterschiede beachtet werden:

- wer die wap.txt **nicht** benötigt, um eine wap-Seite (z.B. index.wml) zu erstellen, kann vorgehen, wie unter "[Ticker](#)" beschrieben
- wer mit der wap.txt **zwei** Dinge erledigen will, nämlich
 1. eine wap-Seite erstellen **und**
 2. die Daten für einen Ticker gewinnen

der muss vorgehen, wie unter "[wap+Ticker](#)" beschrieben

Ticker

Wie gewinne ich die Daten für den Ticker?

Mit einem einfachen Editor (notepad u.a.) eine Datei erstellen, die den Text der mittleren Tabellenspalte enthält und diese Datei dann unter dem Namen wap.txt im WSWIN-Verzeichnis speichern

Tabelle 1

	Text in Datei	was dadurch erreicht wird
. Zeile	<!-- %customfile=data.htm% -->	aus der Datei wap.txt soll eine Datei data.htm erstellt werden
2. Zeile	aktuelles Wetter aus Meinstadt %ws_date%, %ws_time% Uhr ++ %longslabel[2]% %curval[2]% ++ %longslabel[18]% %curval[18]% ++ %longslabel[33]% %curval[33]% ++ Luftdrucktendenz: %ws_barotendencytxt% ++ %longslabel[34]% %curval[34]% ++ Niederschlag am Tag %rainday %	aus den hier eingefügten Variablen erzeugt WSWIN die tatsächlichen Wetterwerte selbstverständlich können und müssen diese Variablen an die eigenen Verhältnisse angepasst werden

Sobald diese Datei von WSWIN abgearbeitet wird, erhalten wir die Datei **data.htm** (im html-Verzeichnis), die dann so aussieht:

aktuelles Wetter aus Meinstadt 11.02.2003, 23:12 Uhr ++ Außentemp. -4,6 °C ++ Feuchte 78 % ++
Luftdruck 1026 hPa ++ Luftdrucktendenz: gleichbleibend ++ Niederschlag aktuell 0,0 l/m² ++
Niederschlag am Tag 0,0 l/m²

Damit ist der erste Teil des Problems gelöst:

Durch WSWIN wurden mit Hilfe der wap.txt die aktuellen Daten erzeugt, die im Ticker verwendet werden sollen.

Lesen Sie nun weiter bei "[Ticker einbinden](#)"

wap+Ticker

Geringfügig anders muss vorgegangen werden, wenn mit Hilfe der wap.txt sowohl eine **wap-Seite** erstellt als auch die Daten für den **Ticker** gewonnen werden sollen.

Mit Hilfe eines Editors wird dieselbe Datei erstellt, wie oben in Tabelle 1 beschrieben. Doch diese Datei wird nicht unter dem Namen wap.txt, sondern unter dem Namen **daten.txt** im WSWIN-Verzeichnis gespeichert.

(der Name "daten.txt" ist willkürlich, aber ich möchte bei diesem Beispiel bleiben)

Nun wird eine bereits vorhandene wap.txt geöffnet und in die 2. Zeile folgender Befehl (hier **rot** dargestellt) eingefügt:

	Text in Datei	was dadurch erreicht wird
. Zeile	<!-- %customfile=index.wml% -->	aus der wap.txt wird eine Datei index.wml erzeugt (die wap-Seite)
2. Zeile	<!-- %openfile=daten.txt% -->	die Datei daten.txt wird geöffnet und abgearbeitet
3. Zeile	<?xml version="1.0" encoding="ISO-8859-1" ?> <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml"> <wml> <card id="Aktuell" title="Wetter in %ws_location%"> <p align="left"> %ws_location% %ws_altitude% usw.	aus diesen (unvollständigen) Zeilen wird die wap-Seite erzeugt

Damit ist auch hier der erste Teil des Problems gelöst:

Durch WSWIN wurden mit Hilfe der wap.txt die aktuellen Daten erzeugt, die im Ticker verwendet werden sollen sowie die wap-Seite.

Lesen Sie nun weiter bei "[Ticker einbinden](#)"

Wie binde ich den Ticker in meine Seite ein?

Um die Voraussetzung für die Datenanzeige zu schaffen, muss **an der Stelle**, an der der Ticker erscheinen soll, der **Code des Applets** eingebunden werden.

Dieser Code lautet folgendermaßen:

```
<applet code="davisticker.class" width="650" Height= "40">
<param name="dataFile" value="data.htm">
<param name="fps" value="250">
<param name="dataInterval" value="60">
<param name="font" value="sans-serif">
<param name="style" value="PLAIN">
<param name="size" value="15">
<param name="fontColor" value="255,000,051">
<param name="borColor" value="153,000,051">
<param name="borWidth" value="4">
<param name="backColor" value="000,000,051">
</applet>
```

Hier ist folgendes von Bedeutung:

Die Datei "davisticker.class" (sie kann auch aus dem user-Verzeichnis von Werner Krenn geladen werden) **und** die Datei "data.htm" (ihre Erzeugung ist unter "wap.txt" beschrieben), müssen sich im **gleichen** Verzeichnis befinden wie die Seite, in die der Applet-Code eingebunden ist (auf der Festplatte und **auf dem Server**).

Dies ist eine der **häufigsten Fehlerquellen** wenn ein Ticker nicht läuft, denn zunächst landet (in den Grundeinstellungen) die Datei data.htm im html-Verzeichnis der Festplatte, also in \WSWIN\html

(durch Pfadangaben können natürlich Anpassungen vorgenommen werden, wenn jemand eine andere Verzeichnisstruktur wünscht)

Wenn nun alle Voraussetzungen erfüllt sind, sollte der Ticker so aussehen, wie in folgendem Beispiel gezeigt:zur:

[Ticker-Demo](#)

Bedeutung der einzelnen Parameter im Applet:

Parameter Name	Parameter Beschreibung
width	Breite des Tickers (in Pixel)
height	Höhe des Tickers (in Pixel)
dataFile	Name der Datei, die die Tickerdaten enthält
dataInterval	nach diesem Zeitraum (in Sekunden) wird überprüft, ob sich die Datei "data.htm" geändert hat. Sehr praktisch, denn zur Aktualisierung des Tickers muss dann nicht die ganze Seite neu geladen werden
font - style- size	Aussehen des Textes (Schriftart - Stil - Größe)
fps	Durchlaufgeschwindigkeit des Tickers
borWidth u. borColor	Breite und Farbe des Rahmens
backColor	Hintergrundfarbe
fontColor	Schriftfarbe