

# Einleitung

**d\_t\_a\_i** ist ein Programm um sich etwas die tägliche Arbeit zu erleichtern. Das Programm kann Dateien "beobachten" und bei einer Änderung eine Kopie automatisch abspeichern.

Kurz gesagt: Wenn sich die Datei ändert, wird automatisch eine Kopie auf dem Dateisystem abgelegt.

Diverse Betriebssysteme unterstützen bereits solche Mechanismen. Für diejenigen, welche dennoch eine einfache Alternative suchen oder nicht immer "Copy & Paste" im Dateimanager Ihrer Wahl machen wollen ist dieses kleine Programm gedacht.

Das Programm ist nicht dafür gedacht ein Versioncontrollsystem (bei Softwareprojekten) oder ein Dokumenten-Management-System zu ersetzen.

## Was bedeutet die Abkürzung?

**d\_t\_a\_i** = **D**on't **t**hink **a**bout **i**t.

Sprichwörtlich: Kümmer Dich nicht drum'.

## Vorraussetzungen

Für den Betrieb des Programm wird Python benötigt. Aktuell getestet mit Version 2.7.

## Download Python

<https://www.python.org/downloads/release/python-2712/>

## Grundsätzliche Arbeitsweise

1. Man definiert in der Datei `file_to_observe.json` die Dateien, welche man gerne "versioniert" hätte.
2. Startet das Programm
3. Sobald eine der beobachteten Dateien sich ändert (i.d.R. wenn man Strg+S drückt oder "Speichern" aus dem Menü wählt) wird im gleichen (oder einem definierten Verzeichnis) auf der Festplatte eine weitere Datei mit gleichem Dateinamen angelegt und dem Zusatz "v0001".

## Beispiel

Beobachtete Datei: C:\Temp\important\_notes.txt

Nach einer Änderung (Speicherung) gibt es dann im Verzeichnis: C:\Temp\important\_notes\_v0001.txt, C:\Temp\important\_notes\_v0002.txt, ...

## Definieren der Dateien welche zu beobachten sind

Die Konfiguration ist in der Datei **file\_to\_observe.json**.

Diese Datei ist im JSON Format.

## Beispielinhalt

```
{
  "1":{"key_Filename":"C:\\users\\foo_bar\\documents\\File_A.doc", "key_Save_To_Same_Folder" : "True"},
  "2":{"key_Filename":"C:\\Temp\\important_notes.txt", "key_Save_To_Same_Folder": "True"},
  "3":{"key_Filename":"C:\\config\\server.ini", "key_Save_To_Same_Folder": "False",
"key_Alternate_Backup_Folder": "C:\\Backup\\my_server_configs\\"},
  "4":{"key_Filename":"C:\\dev\\some_project\\important_file.txt", "key_Save_To_Same_Folder": "False",
"key_Alternate_Backup_Folder": "C:\\users\\foo_bar\\documents\\dropbox\\history_dev\\"}
}
```

Wichtig ist zu beachten, dass die letzte Zeile am Ende nicht durch ein Komma getrennt wird.

```
{
  "3":{"key_Filename": "<Dateiname mit Pfad. mit \\ unter Windows>",
"key_Save_To_Same_Folder": "False", "key_Alternate_Backup_Folder": "<Pfad mit \\ unter Windows>"},
  "4":{"key_Filename": "C:\\dev\\some_project\\important_file.txt", "key_Save_To_Same_Folder": "True"}
}
```

Sollte sich die Konfiguration ändern, muss das Programm neu gestartet werden (da diese nur am Start eingelesen wird).

## Programm-Konfiguration

Grundsätzliche Programmeinstellungen befinden sich in der Datei **config.json**.

Aktuell gibt es folgende Parameter:

```
{
  "params": {
    "check_intervall": "2",
    "fileversion_sep": "v",
    "init_number": "0001",
    "method_change_detect": "content"
  }
}
```

Parameter	Erläuterung
check_intervall	Häufigkeit der Prüfung auf Änderung mit Angabe in Sekunden
fileversion_sep	Trenner im Dateinamen zwischen Originalnamen und Versionsnummer (z.B. Datei_v0001.txt)
init_number	Nummer mit welcher begonnen werden soll zum Zählen
method_change_detect	Parameter zum Erkennen einer Dateiänderung. Folgende Optionen stehen zur Verfügung: content=md5Sum der Datei fileattr=Datum und Größe der Datei

Stand der ReadMe\_DE  
2016-09-02 - 21:07