

# Arbeitszeit-Auswertungs Programm

---

This project is intended to:

- Practice the complete process from **problem analysis to implementation**
- Apply basic **Python** programming concepts learned in the Programming Foundations module
- Demonstrate the use of **console interaction, data validation, and file processing**
- Produce clean, well-structured, and documented code
- Prepare students for **teamwork and documentation** in later modules
- Use this repository as a starting point by importing it into your own GitHub account.
- Work only within your own copy — do not push to the original template.
- Commit regularly to track your progress.

## Analysis

### Problem

Das Problem ist, dass der Vorgesetzte ein File erhält in dem alle Mitarbeiter ihre gestempelten Zeiten eintragen und er muss immer alles manuell berechnen. Er muss berechnen wie lange die Mitarbeiter gearbeitet haben und schauen, ob sie die vertraglichen Rahmenbedingungen verletzt haben. Er möchte eine Übersicht haben über die jeweiligen Mitarbeiter, in der gezeigt wird: Mitarbeiter, Pensum, Ist-Zeit, Soll-Zeit, Differenz-Stunden, Einhaltung der Rahmenbedingungen und falls verletzt, Welche Rahmenbedingung verletzt wurde und dies nicht immer manuell berechnen müssen.

### Scenario

Der User will eine Übersicht über die Stunden haben, indem er ein File importiert, welches die wöchentlichen Stempelungen der Mitarbeiter beinhaltet. Schlussendlich soll er als Output in der Konsole, eine Übersicht erhalten in der aufgeführt ist:

- Nachname, Vorname, Pensum
- Effektivstunden
- Soll-Zeit
- Differenz-Zeit
- Pausen-Zeit
- Vertragsbedingungen eingehalten?
- Begründung der Vertrags-Verletzung

### User stories:

1. Als User möchte ich, eine CSV-Datei einlesen können, in der die Mitarbeiter ihr Pensum und ihre Zeitstempelungen für eine Woche aufgeführt haben.
2. Als User möchte ich, eine Übersicht der Soll-Zeit jedes einzelnen Mitarbeiters erhalten.
3. Als User möchte ich, eine Übersicht der Differenz-Zeit jedes einzelnen Mitarbeiters erhalten.
4. Als User möchte ich, eine Angabe des Pensums des Mitarbeiters erhalten.
5. Als User möchte ich, bei Fehlern in der CSV-Datei, eine Information erhalten wo der Fehler ist und diese anpassen können.

6. Als User möchte ich, die Arbeitszeiteinhaltung an dem Pensum angepasst des Mitarbeiters erhalten.  
(Max 48h = 100%)
7. Als User möchte ich, eine Angabe erhalten ob die vertraglichen Rahmenbedingungen eingehalten wurden.
8. Als User möchte ich, gegebenenfalls eine Angabe erhalten welche Rahmenbedingung verletzt wurde.

**Use cases:**

- Speichern des Files mit Stempelungen der Mitarbeiter.
- Filepath, des Files welches man auswerten will, eingeben.
- Das Programm durchlaufen lassen.
- Bei Fehlern im File, erhalte ich die Information, wo der Fehler liegt. Dieser muss dann korrigiert werden.  
Dann das Programm wieder durchlaufen lassen.
- Output wird als Übersicht in der Konsole ausgegeben.

---

## ☒ Project Requirements

Each app must meet the following three criteria in order to be accepted (see also the official project guidelines PDF on Moodle):

1. Interactive app (console input)
2. Data validation (input checking)
3. File processing (read/write)

---

### 1. Interactive App (Console Input)

Der User Startet das Programm. Das Programm interagiert mit dem User in dem der User den Filepath der CSV-Datei in das Programm eingibt.

---

### 2. Data Validation

Das Programm muss überprüfen ob die angegebenen Daten korrekt sind:

- Ist der Mitarbeiter ein Name.
- Sind die Timestamps korrekte Zeiten. Im richtigen Format und überhaupt möglich.
- Ist das Pensum >0 und <100.
- Jeder User hat 4 Timestamps pro Tag (00:00 bei leeren Eingaben)
- Jeder User kann nur einen Tag gestempelt haben (!= 2 x Montag)
- Der Filepath der Usereingabe existiert.
- Wurden die Vertragsbedingungen verletzt.

### 3. File Processing

Das Programm liest die Daten, in dem es das Input CSV-File verwendet. Das Programm gibt Daten aus, in dem es die berechneten Resultate (Mitarbeiter, Überstunden, Minusstunden, Pensum, Rahmenbedingungen) in einem Output in der Konsole als Übersicht ausgibt.

## Implementation

### Technology

- Python 3.x
- Environment: GitHub Codespaces

### Repository Structure

```
My-Project-DEV-Container-included-/  
├── Data  
│   ├── Stempelzeiten KW_XX.csv      # Input File, Stempelungen  
│   └── Stempelzeiten KW_XY.csv      # Input File, Stempelungen  
├── Filehandling.py                  # Verarbeitung der Daten und Output  
├── Funktions_hierarchiebaum.pptx    # Wie die Funktionen zusammenarbeiten  
├── Arbeitszeit-Auswertungsprogram.pptx # Präsentation für die Applikation  
└── README.md                        # Projektbeschreibung und Meilensteine
```

### How to Run

1. Öffnen des repository in **GitHub Codespaces**
2. Input File in das Repository einfügen.
3. Öffnen des Terminals.
4. Run:

```
python3 FileHandling.py
```

5. Filepath in das Terminal schreiben
6. Im Terminal die Übersicht erhalten.

### Libraries Used

- CSV für CSV Verarbeitung

## Team & Contributions

Name	Contribution
Denis Silva	Stundenberechnung und Validierung, Rahmenbedingungen, Überarbeitung ReadME und Funktions_hierarchiebaum, Erstellung der Präsentation
Mehmedali Abdiu	CSV Einlesung und Validierung, Erstellung des ReadMe und Funktions_hierarchiebaum, Überarbeitung Präsentation, Docstrings im Code, Besprechungstermine organisieren

## Contributing

- Use this repository as a starting point by importing it into your own GitHub account.

- Work only within your own copy — do not push to the original template.
- Commit regularly to track your progress.

## License

This project is provided for **educational use only** as part of the Programming Foundations module.

[MIT License](#)