**Leitpfaden – Coding:**

* **General Aspects:**
  + **Comments**
    - Ziele:
      1. Beschreibung der Motivation des Codes  
         („Erklärende Comments“, keine reine Code-Beschreibung)
      2. Erläuterung einzelner Variable
      3. Gliederung des Codes
      4. „Lessons learned“ mit Informationen für die zukünftige Instandhaltung
    - Arten:
      1. „Inline Comments“:
         * Hinter der Zeile – beginnend mit zwei Leerzeichen
         * Example:   
           a = np.array([0.5,0.5]) # Weights of PF for Optimization
      2. Block-Kommentar:
         * Längerer Kommentar der mehrere Zeilen umfasst.
         * Startet in jeder Zeile mit „# “
         * Example
    - No Gos:
      1. Professionell bleiben (Sprachgebrauch, etc.)
      2. Nicht die Zeilengrenze übersch
      3. Keine reine Wiederholung der Code-Operationen.  
         [Fokus auf die Motivation und das Resultat]
      4. ToDo Kommentare begrenzen,   
         Besser Codetags in Gitlab nutzen
* **Functions:**
  + **Naming**
    - Explizit halten, wenn nicht teil einer konkreten Funktionsgruppe / Modul / Klasse
  + **Naming - Format**
    - small letters
    - snake\_style
  + Annotations
    - Arguments: Format and Dimension
    - Output: Format and Dimension
  + **Umfang der Funktionen:**
    - Kompromiss zwischen Intra-Funktionskomplexität und Inter-Funktionskomplexität
    - Zu kurz wie möglich, aber…
    - Keine Vollständige „Aufsplittung“ in Unterfunktionen
    - Ausgliederung in Unterfunktionen bei der Vermutung, dass diese auch für andere Aspekte / Funktionen nützlich sein können.
    - Ideal: 30 Zeilen, Max. 100 Zeilen
  + **Docstring:**
    - Ziel:
      1. 1-2 Sätze zum Ziel des Ansatzes
    - Parameter
      1. Kurze Beschreibung der Parameter  
         Dimension + Inhalt + Datentyp
    - Returns
      1. Kurze Beschreibung der Parameter  
         Dimension + Inhalt + Datentyp
    - Notes:
      1. Optional
      2. Kurzes Beschreiben von Besonderheiten
  + **Initial Dimension / Consistency Checks**
    - Starting Point should be a Check of the Dimensions of the Inputs   
      (if necessary, esp. when Matrix Multiplication is used afterwards)
  + Eindeutige Datentyp der RÜckgabe:
    - Eine Funktion soll möglichst immer denselben Datentyp zurückgeben
  + **Debugging:**
    - Keine „None“ für Fehlerrückgabe
    - Ausnahmen bei Fehlern zurückgeben und keine Fehlercodes
* **Module:**
  + **Docstring**
    - **Kurze Beschreibung des Modulinhalts**

**Gitlab – Aufbau:**

* **U**nterteilung: Entw / Test / Prod
* Venv?
* Unit Tests für automatische Überprüfung
* Tags und Issues