

# mug121 - Wissenschaftliche Datenverarbeitung: Hausaufgabe 04

Niklas Heidemann (heidemann@geo.uni-bonn.de)

## Aufgabe 1: print-container Funktion erweitern

Benennen Sie dieses IPython-Notebook um zu `mug121_HA04_Name-Vorname.ipynb` und entwickeln Sie in den Codezeilen weiter unten Ihren Code aus Hausaufgabe 03 weiter:

1. Erstellen Sie wieder einen auskommentierten Header (Kopfzeile) mit Zweck (HA04), Autor (Ihr Name), Datum und Programmiersprache.
2. Erstellen Sie eine Funktion, die den folgenden Code enthält:

```
def print_container(container):  
    """  
    Ihre Funktionsbeschreibung vom letzten Mal, überarbeitet und  
    erweitert für die folgenden Aufgaben (Docstring)  
    """  
  
    print("Container-Info:\n", container, "\n des Typs:",  
          type(container), "\n mit Anzahl Einträge:" , len(container))
```

3. Erweitern Sie die oben stehende Funktion auf Basis Ihrer Überlegungen in HA03. Welche Informationen können noch hilfreich sein? Welche weiteren Strings, Formatierungen etc. können Sie in dem Print-Statement unterbringen? Wollen Sie unterschiedliche Ausgaben für unterschiedliche Containertypen haben (bspw. eigene Ausgabe für build-in Container vs. NumPy-Array / Pandas Series/DataFrame)? Lassen Sie alle Ihre Überlegungen einfließen und bauen Sie eine Funktion, die zur Ausgabe vieler Container-Infos mit einem Aufruf pro Container genutzt werden kann.
4. Erstellen Sie wieder einen beliebigen Container (gerne alle bekannten Containertypen verwenden) und speichern Sie ihn als Variable `my_container` :

```
my_container =
```

Rufen Sie die in 1.2 erstellte und erweiterte Funktion auf und übergeben Sie ihr den erstellten Container `my_container` unter Nutzung des folgenden Codeschnipsels:

```
print_container(my_container)
```

5. **Wichtig:** Kommentieren Sie Ihren Code: Was passiert in den einzelnen Zeilen / Blöcken? Übernehmen Sie auch hier gerne Ihre Ergebnisse aus der letzten Hausaufgabe.

6. Ersetzen Sie die Zeichenkette *Beschreibung der Funktion* durch eine kurze Beschreibung, die die Frage beantwortet: Was macht die Funktion? Aktualisieren Sie die Beschreibung entsprechend zu Ihren Erweiterungen.
7. Speichern Sie Ihr Notebook (Diskette oben links im Notebook-Fenster, `File -> Save Notebook`, `Ctrl + S` oder `File -> Export Notebook As...`).

### Hinweise:

- Zum Kommentieren müssen Sie Ihrem Kommentar die Raute `#` voranstellen. Das kann in einer Zeile hinter Code durchgeführt werden.
- Achten Sie darauf, den Funktionskörper - alles nach dem Doppelpunkt - in 1.2 richtig einzurücken (Taste Tab). Der Code in 1.3 und 1.4 wird dann wieder linksbündig gesetzt.
- Sollten Sie auf Fehler stoßen: Nicht verzweifeln! Tippen Sie Ihren Code, überprüfen Sie, ob Sie die Tippfehler gemacht haben und schreiben Sie bei unlösbaren Fehlern einen kurzen Kommentar-Text, was Sie an dem Fehler nicht verstehen.
- Starten Sie bei Fehlern ggf. Ihren Kernel neu: Der Kernel sollte oben rechts auf *py3-general* eingestellt sein. Nutzen Sie `Run -> Restart Kernel and Run All Cells`.
- Auch nicht funktionierender Code wird in dieser Hausaufgabe als Abgabe akzeptiert, sofern Sie Ihre Antwort "sprechen" lassen: **Code kommentieren! Und zwar soviel wie möglich!**

```
In [1]: # # Codezeile Aufgabe 1

# Codezeile Header

# Codezeile Funktion

# Codezeile Container-Definition

# Codezeile Funktionsaufruf
```

---

### Aufgabe 2: Statistik an 10 Einträgen

Erstellen Sie sich einen Container mit **10 Einträgen**. Der Container kann eine Liste, ein Tupel oder ein Dictionary sein. Die Einträge müssen vom Typ `int` oder `float` sein.

Nutzen Sie built-in Funktionen wie `sum`, `min`, `max`, `len` um sich eine **simple "Statistik"** des Datencontainers ausgeben zu lassen. Denken Sie beispielsweise auch an Mittelwert und

Median. Formatieren Sie die Ausgabe möglichst allgemeingültig und schreiben Sie gegebenenfalls eine Funktion für diese Aufgabe.

In [2]: `# Codezeile Aufgabe 2`

---

### Aufgabe 3: Paper lesen

Laden Sie das Paper *Wilson-et-al\_2016\_1609.00037v2.pdf* aus dem Ordner *Literatur* auf eCampus herunter [Link auf das Paper in eCampus](#).

Das Paper trägt den Titel "Good Enough Practices in Scientific Computing" (Wilson et al. 2016).

Lesen Sie das Paper und notieren Sie Stichpunkte zu den folgenden Fragen:

1. Wie gehen Sie beim Durchlesen vor? Macht es Sinn, zuerst am Anfang und am Ende zu lesen und dann den gesamten Text? Was denken Sie?
2. Was enthalten die einzelnen Kapitel (inkl. der einleitenden Kapitel Abstract, Author Summary, Introduction, Acknowledgements)? Machen Sie einen Stichpunkt zu jedem Kapitel, falls möglich.
3. Was sind die wesentlichen Aussagen? Fassen Sie die wesentlichen Aussagen des Papers in maximal 5 Stichpunkten zusammen.
4. Warum behandeln wir dieses Paper?

Halten Sie die Antworten in der folgenden Markdown-Zeile fest:

### Antwort Aufgabe 3:

---

### Aufgabe 4: Eigener Dictionary Datencontainer

Erstellen Sie einen **Dictionary** und speichern Sie ihn als Variable `data`. Dieser Dictionary soll folgendermaßen aussehen:

- Zwei Einträge (Schlüssel-Wert-Paare)
- Schlüssel sind die Namen von Städten, formatiert als String
- Werte sind Tupel oder Listen mit zeitlich abhängigen Daten (Temperatur / Bevölkerung / Erdbebenstärke etc. / andere geowissenschaftlich relevante Daten)
- maximal 10 Einträge pro Tupel/Liste

Geben Sie die Werte aus, indem Sie `data` einzeln mit den Schlüsseln indizieren.

Hinweis: gewünschtes Schema

```
data = {"Stadt1": [Werte1], "Stadt2": [Werte2]}
```

Hinweis 2: Hier Arbeit hineinstecken wird sich ebenfalls lohnen: Den eigenen Datensatz werden wir demnächst plotten.

```
In [3]: # Codezeile Aufgabe 4
```

---

**Aufgabe 5 - Abgabe:** Schicken Sie das IPython-Notebook `mug121_HA04_Name-Vorname.ipynb` an die folgende Mailadresse ab (mit Rechtsklick->Download oder File->Download kann das Notebook auf Ihren Computer heruntergeladen werden):

heidemann@geo.uni-bonn.de mit dem Betreffspräfix `[mug121]`