

# Tutorium 0

## Vagrant, Jupyter Notebooks, Python

### Big Data Engineering

Prof. Dr. Jens Dittrich

[bigdata.uni-saarland.de](https://bigdata.uni-saarland.de)

25./26. April 2022

# VM Installation

## Frage

Wer hat schon die VM via `vagrant` installiert und ein Jupyter Notebook getestet?

- (a) Klar, alles läuft reibungslos!
- (b) Ja, die VM läuft aber ein Notebook habe ich noch nicht ausprobiert.
- (c) Naja, ich habe es versucht aber es hat leider nicht geklappt.
- (d) Was ist `vagrant`?

# Vagrant, Jupyter



ausführliche Anleitung sowie ein FAQ zum Vagrantfile im CMS unter  
*Informationen*

# Vagrant - Cheat Sheet

## Host (dein Rechner):

- `vagrant up`: startet die VM, konfiguriert die VM beim ersten Aufruf
- `vagrant halt`: beendet die VM
- `vagrant reload`: startet die VM neu
- `vagrant status`: zeigt den Status der VM an
- `vagrant ssh`: via SSH mit der VM verbinden
- `vagrant destroy`: entfernt die VM

## Guest (die VM):

- `exit`: schließt die SSH-Verbindung mit der VM
- `jupyter notebook --no-browser --ip=0.0.0.0`:  
startet einen Jupyter Server

# Jupyter - Workflow

1. lade die [Jupyter Notebooks](#) lokal auf deinen Rechner herunter
2. verschiebe sie in den shared-Ordner der VM
3. verbinde dich mit der VM  
`$ vagrant ssh`
4. starte einen Jupyter Server  
`$ jupyter notebook --no-browser --ip=0.0.0.0`
5. öffne die angezeigte URL (`http://127.0.0.1:8888/?token=`) in einem Browser deiner Wahl
6. schließe den Jupyter Server mit Strg-C und bestätige mit Y und Enter
7. trenne die Verbindung zur VM  
`$ exit`

## Materialien

[Zugriffsrechte hier bearbeiten...](#)

Neue Kategorie

Neue Datei

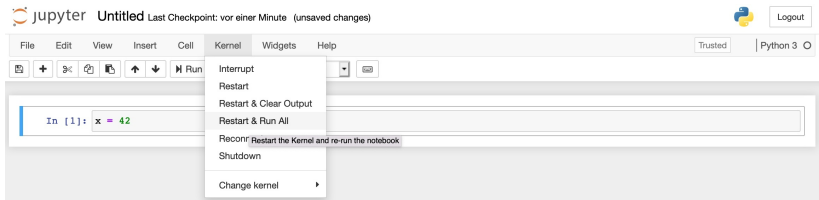
Neuer Link

Python		    
 Anleitung		 
 Einführung (Jupyter Notebooks)		 
 Einführung (Videos)	Einführung und Crashkurs behandeln die gleichen Jupyter Notebooks	 
 Vagrantfile (VM Definition)		 

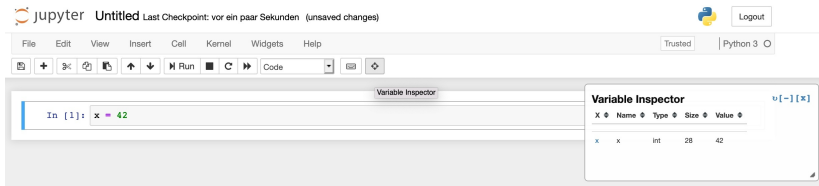
ausführliche Einführung sowie alle dazugehörigen Jupyter Notebooks im CMS unter *Informationen* → *Materialien*

# Jupyter - Notebooks

- einzelne Zelle ausführen mit Strg-Enter (oder Cmd-Enter unter macOS)
- alle Zellen ausführen über *Kernel* → *Restart & Run All*



- aktuelle Werte von Variablen anzeigen über den Variablen-Inspektor (Fadenkreuz)



# Aufgabe 1

## Frage

Implementieren Sie in einem neuen Jupyter Notebook eine Funktion *find\_duplicates*, die zwei Listen von Zahlen als Input erhält. Dabei soll sie überprüfen, welche Zahlen in beiden Listen enthalten sind und diese anschließend in einer zusätzlichen Liste ausgeben. Achten Sie hierbei auf eine lineare Laufzeit in der Länge der Listen.

Erstellen Sie in einer weiteren Jupyter Zelle die zwei Listen [15, 36, 0, 11, 2] und [2, 56, 111, 36, 90] und printen Sie diese. Führen Sie anschließend ihren Algorithmus mit den beiden Listen als Parametern aus und printen Sie das Ergebnis.



# Aufgabe 1

## Lösung

```
def find_duplicates(lst1, lst2):  
    lst1_set = set()  
    for element in lst1:  
        lst1_set.add(element)  
  
    new_list = []  
  
    for element in lst2:  
        if element in lst1_set:  
            new_list.append(element)  
  
    return new_list  
  
lst1 = [15, 36, 0, 11, 2]  
lst2 = [2, 56, 111, 36, 90]  
print(lst1, lst2)  
print(find_duplicates(lst1, lst2))
```

## Aufgabe 2

### Frage

Implementieren Sie eine Funktion *sort*, die eine Liste von Zahlen als Input erhält und diese mithilfe von BubbleSort sortiert.

Erstellen Sie in einer weiteren Jupyter Zelle die Liste [17, 40, 51, 101, 25] und printen Sie diese. Führen Sie anschließend ihren Algorithmus mit der Liste als Parameter aus und printen Sie die sortierte Liste. Sie können in dieser Aufgabe das Notebook aus der vorherigen Aufgabe verwenden.

## Aufgabe 2

### Lösung

```
def sort(lst):
    length = len(lst)

    for i in range(1, length):
        for j in range(0, length - i):
            if lst[j] > lst[j+1]:
                number = lst[j]
                lst[j] = lst[j+1]
                lst[j+1] = number

lst = [17, 40, 51, 101, 25]
print(lst)
sort(lst)
print(lst)
```