

# Fortbildungsmodul M4:

## Datenprojekte und Datenexploration: Umweltdaten mit Sensoren sammeln und analysieren

Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny



## Fortbildung – Termine

- **08.03.22 9-16 Uhr Präsenztag 1**
- **16.03.22, 14:30-16 Uhr Online Termin 1 (Zoom)**
- **23.03.22, 14:30-16 Uhr Online Termin 2 (Zoom)**
- **ERPROBUNGSPHASE**
- **22.06.22, 9-16 Uhr Präsenztag 2**

# Vorstellung

## 1. Das Team

## 2. Die Teilnehmenden

Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny

## Vorstellung des Teams

#name

#DenisePallerberg

#schule

#Wittekind-Gymnasium Lübbecke

#fächer

#IfPaM

#erwartungen

#neugierdeAufWasNeues

## Vorstellung des Teams

#name

#SusannePodworny

#schule

#UniversitätPaderborn

#fächer

#MD

#erwartungen

#interessierteLehrkraefte

## Vorstellung des Teams

#name

#HaraldNeubeck

#schule

#GymnasiumSchlossNeuhaus

#fächer

#infoGesch

#erwartungen

#neueBlickeAufNeueThemen

## Vorstellung des Teams

#name

#SvenHüsing

#schule

#UniversitätPaderborn

#fächer

#IfM

#erwartungen

#spannendeDiskussionenUndAustausche

# Vorstellung der Teilnehmenden

#name

#schule

#fächer

#erwartungen

# Fortbildung – gesamter Überblick

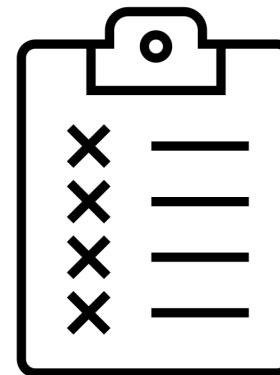
- **08.03.22 Präsenztag 1**
  - Umweltdaten mit Sensoren erfassen
  - Umweltdaten in Computational Essays explorieren
- **16.03.22 Online Termin 1 (Zoom)**
  - Sensebox programmieren
  - Sensebox kaufen oder selbst bauen?
- **23.03.22 Online Termin 2 (Zoom)**
  - Kernlehrplanbezug
  - Wiederholung MyBinder
- **ERPROBUNGSPHASE**
- **22.06.22 Präsenztag 2**
  - Erfahrungsaustausch
  - Tipps und Tricks für die Durchführung
  - Anschlussmöglichkeiten

## Ablauf des heutigen Termins

- Jetzt **Grundlagen für den Unterricht**
  - Überblick über die Unterrichtsreihe
  - Hintergrundwissen Computational Essay/Epistemisches Programmieren
- 10:30 **Pause**
- 10:45 **Das Computational Essay als Worked Example I**
  - PPDAC als Hintergrund inkl. Fragen stellen ist nicht schwer (?!)
  - Daten mit der Sensebox sammeln
- 12:15 **Mittagspause**
- 13:15 **Das Computational Essay als Worked Example II**
  - Daten einlesen und im Jupyter Notebook explorieren und auswerten
- 14:45 **Pause**
- 15:00 **Ergebnisse interpretieren und präsentieren**
  - Schlussfolgerungen aus den Datenexplorationen
- 15:30 **Abschlussdiskussion**

## Bevor es losgeht...

Wir sind Euch sehr dankbar, wenn Ihr die Umfrage hinter dem folgenden Link ausfüllen könntet: [go.upb.de/VuE-Programmieren](http://go.upb.de/VuE-Programmieren)



Ihr helft uns damit sehr bei unserer Forschung und der Erstellung passender Unterrichtsprojekte und –vorhaben.

Vielen Dank 😊

## Kurze Einordnung „Projekt Data Science und Big Data in der Schule“ (ProDaBi)

- Projektkurs für die Oberstufe „Data Science und KI – das Parkplatzprojekt“
- 4 Module für Klassenstufen 8-10
  - Datendetektive (Data Science und künstliche Intelligenz ohne Programmieren)
  - Data Science und künstliche Intelligenz mit Jupyter Notebooks
  - Datenbewusstsein (mit Jupyter Notebooks)
  - Umwelt-Datenprojekte mit Jupyter Notebooks
- 2 Module für Klassenstufen 5/6
  - Entscheidungsbäume unplugged
  - Datenbewusstsein

## Kontakt

- Bei weiteren Fragen, schreibt uns gerne eine Mail an

[prodabi@campus.upb.de](mailto:prodabi@campus.upb.de)

- ...oder besucht uns im Internet unter

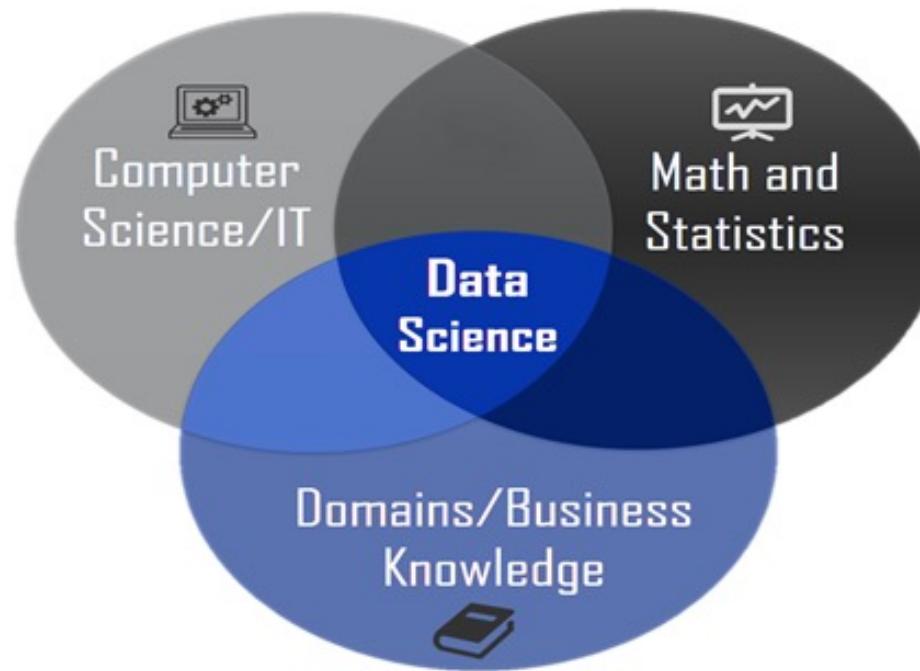
[www.prodabi.de](http://www.prodabi.de)

- ...oder tragt Euch in unsere Mailingliste ein für weitere Informationen zu Veröffentlichungen von Material, Lehrkräftefortbildungen, Workshops etc.:

<https://lists.uni-paderborn.de/mailman/listinfo/prodabi-schule>

## Data Science

**Data Science, Data Literacy und epistemisches Programmieren als fächerübergreifende Ideen für den Schulunterricht**



Enge Anbindung an Lehrplan Informatik

Dazu auch: Data Literacy Charta: <https://www.stifterverband.org/charter-data-literacy>

## Data Science

- Einsatz eines **Datenanalysetools** aufgrund **großer Datenmengen** unvermeidbar für Data Science
- **Multivariate** und **reale Daten** aus der Umwelt-Umgebung von Schülerinnen und Schülern
- Reale **Fragestellungen**, die zu neuen Erkenntnissen und/oder Handlungen führen

09.12.2021, 15:17 Uhr

An Schulen wird zu wenig gelüftet

## Dicke Luft im Klassenraum

In Klassenzimmern wird laut einer Studie trotz Corona nicht genug gelüftet. Technische Anlagen schneiden überraschend schlecht ab.

21. Juni 2020  
**Studie beschreibt schlechte Luftqualität in Klassenzimmern – Forscher warnen: Dicke Luft erhöht die Viruslast**

## Dicke Luft in deutschen Klassenzimmern

In den meisten Klassenräumen herrschte eine ungesunde Luft, was gesundheitlich ungünstig erkannt wurden: Die Luft ist zu trocken und zu feucht. Schulbehörden konzentrieren sich auf die Lüftung, obwohl sie nur in wenigen Schulen funktionieren.

Von Michael Hartwig  
22.10.2010, 08.37 Uhr

BUNDESPROGRAMM GEFORDERT

## Wider die dicke Luft an Schulen

AKTUALISIERT AM 01.04.2021 - 12:36

Schlechte Luft vermindert die Lerngeschwindigkeit und beschleunigt die Ausbreitung von Viren, sagt eine Initiative aus Wissenschaft und Stiftungen. Ein professionelles Lüftungskonzept an Schulen rechne sich.

n Unterricht  
setzt eine  
n wurden.  
cht. Einem  
aptisch

### Linkliste:

<https://www.tagesspiegel.de/wissen/an-schulen-wird-zu-wenig-gelueftet-dicke-luft-im-klassenraum/27878822.html>

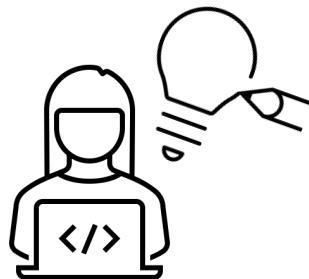
<https://www.news4teachers.de/2020/06/schlechte-luftqualitaet-in-klassenzimmern-dicke-luft-erhoeht-die-viruslast/>

<https://www.spiegel.de/lebenundlernen/schule/zu-viel-co2-dicke-luft-in-deutschen-klassenzimmern-a-724364.html>

<https://www.faz.net/aktuell/karriere-hochschule/bundesprogramm-gefordert-wider-die-dicke-luft-an-schulen-17272540.html>



# Die Unterrichtsreihe



# Die eigene Umwelt erforschen:

## Analyse von Umweltdaten mit Arduinos und Jupyter Notebooks

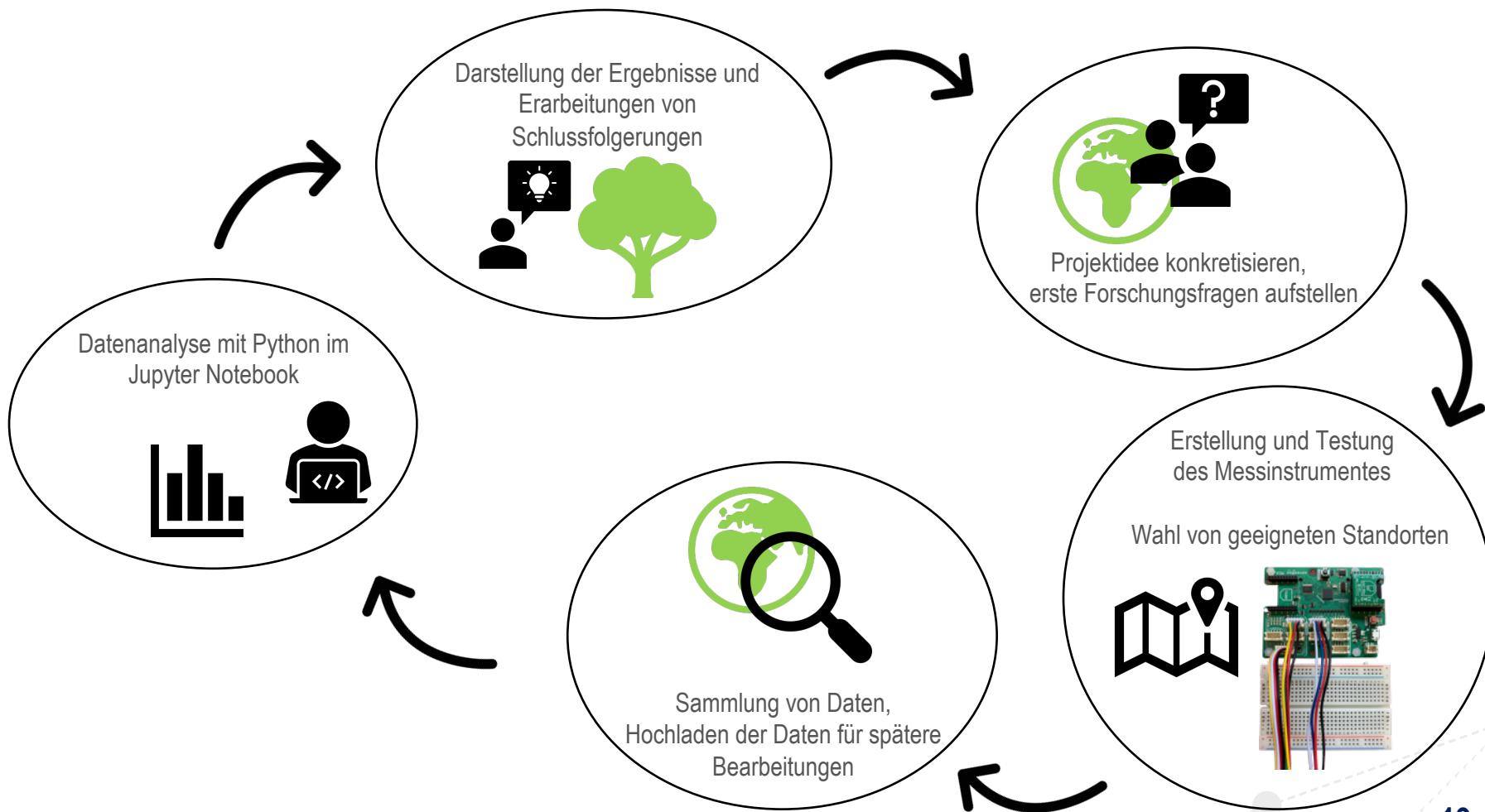
Wie lassen sich „Umweltdaten“ erheben und auswerten?  
Vorstellung eines Data-Science-Projektes

Aktuelles Thema:  
Luftqualität/Klimadaten  
im Klassenraum

Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny



# Ablauf des Projektes



# Datenanalyse mit Python im Jupyter Notebook (epistemisches Programmieren)

Welche Schritte werden dabei ausgeführt?

- Bibliotheken importieren, Daten einlesen und anzeigen lassen
- Daten bereinigen und indizieren
- Daten auswerten und visualisieren
  - Fragestellung entwickeln
  - Befehl/Visualisierung finden
  - ausprobieren
  - “tunen”
  - Dokumentieren
  - interpretieren



# Überblick über die Unterrichtsreihe

## Stunde 1

- Einführung, Problemaufriss „Dicke Luft im Klassenzimmer“

## Stunde 2 und 3

- Eigene Datenerhebung vorbereiten (Senseboxen)
- Datensammeln
- Fragen erarbeiten

## Stunde 4-7

- Daten in Jupyter Notebooks einlesen und auswerten
- Computational Essay erstellen

## Stunde 8-9

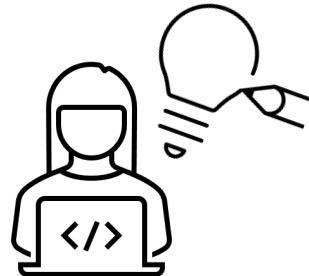
- Peerfeedback

## Stunde 10

- Erkenntnisse zusammentragen
- Handlungsstrategien

# Input:

## Computational Essays und epistemisches Programmieren



# Epistemisches Programmieren

Motivation:

- Programmieren nicht nur als Selbstzweck, sondern auch als Möglichkeit/Werkzeug wahrnehmen,...
  - (persönlich relevante) Erkenntnisse zu gewinnen
  - anderen die gewonnenen Erkenntnisse zugänglich zu machen
  - sich selbst auszudrücken

# Epistemisches Programmieren

Welche Erkenntnisse lassen sich gewinnen?

- Externe Erkenntnisse
  - Erkenntnisse, die sich auf die „reale Welt“ beziehen und die die Programmierer:innen beim Programmieren oder anhand der Programmierergebnisse gewinnen können
- Internale Erkenntnisse
  - Erkenntnisse, die sich auf den adäquaten Umgang mit digitalen Artefakten beziehen, um Erkenntnisse zu gewinnen
  - Beispiel: Verwendung spezifischer Bibliotheken/Methoden

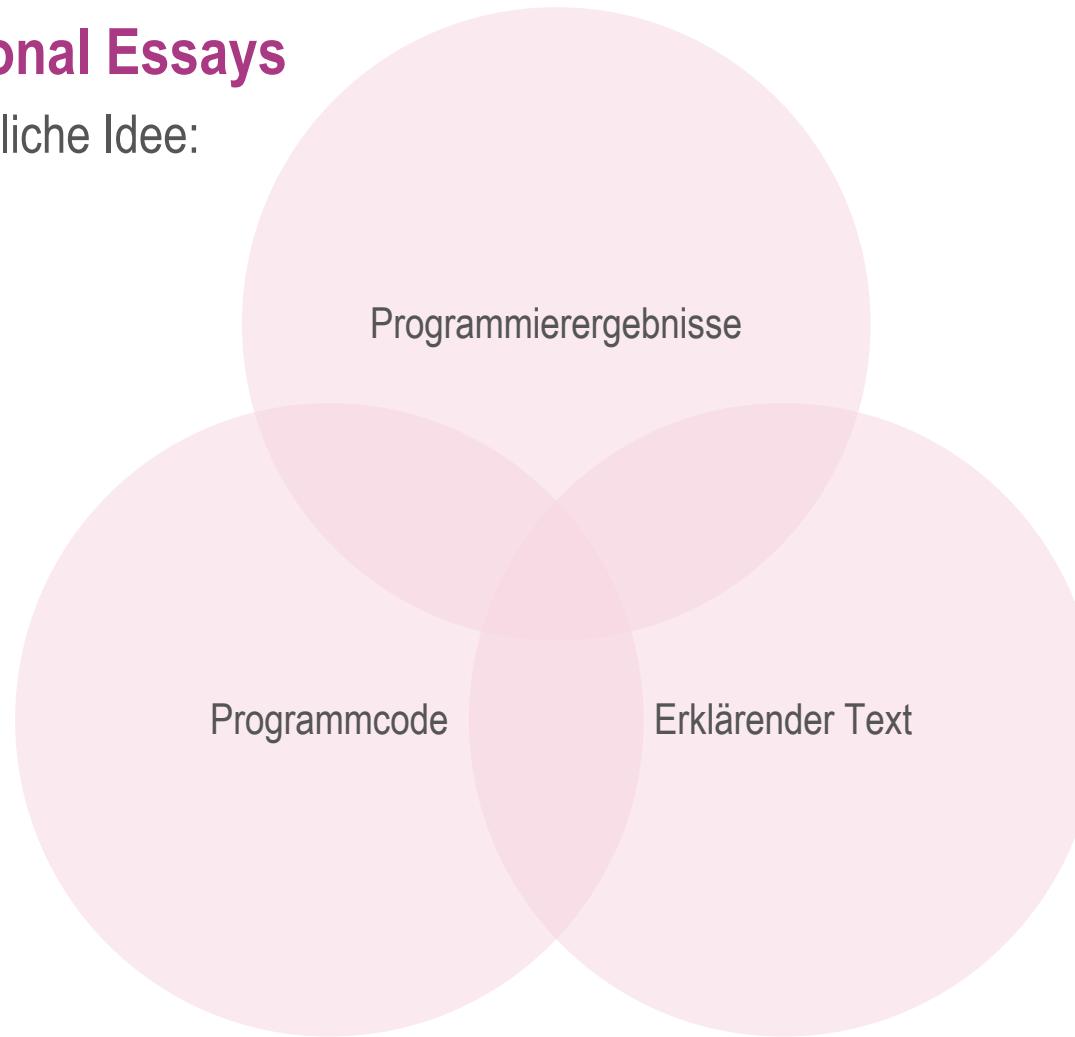
# Epistemisches Programmieren

Produkte:

- Simulationen (Agent-Based-Modeling)
  - Programmieren einzelner digitaler Akteure innerhalb eines größeren Szenarios, aus dem heraus sich dann Erkenntnisse ergeben
- Visualisierungen
  - (datenbasiertes) Erstellen von Visualisierungen, aus denen dann Erkenntnisse gewonnen werden können
- Text-Basierte Produkte (Epistemisches Schreiben)
  - Festhalten von Erkenntnissen und Zusammenhängen, woraus wiederum weitere Erkenntnisse gewonnen werden können
- Computational Essays
  - Digitale Essays, mit denen ein:e Leser:in interagieren kann (siehe nächster Abschnitt)

## Computational Essays

- Grundsätzliche Idee:



# Computational Essays

## Ziele:

- Programmierergebnisse und daraus resultierende Erkenntnisse für die Lesenden **nachvollziehbar** und somit **reproduzierbar** machen
- Einladen, mit den Programmierergebnissen **herumzutüfteln**, um diese besser zu verstehen
- Vorgehensweise **protokollieren** und **dokumentieren**

## Aspekte eines Computational Essays:

- Programmcode (in „Code-Zellen“)
- Ausgaben des Programmcodes (Visualisierungen, Textausgaben,...)
- Erläuternder Text (in „Markdown-Zellen“)
  - Dokumentation des Codes
  - Beschreibung des Vorgehens
  - Interpretation der Ergebnisse

## Computational Essays

# Beispiele

- Umweltdaten
- Parkplatzdaten

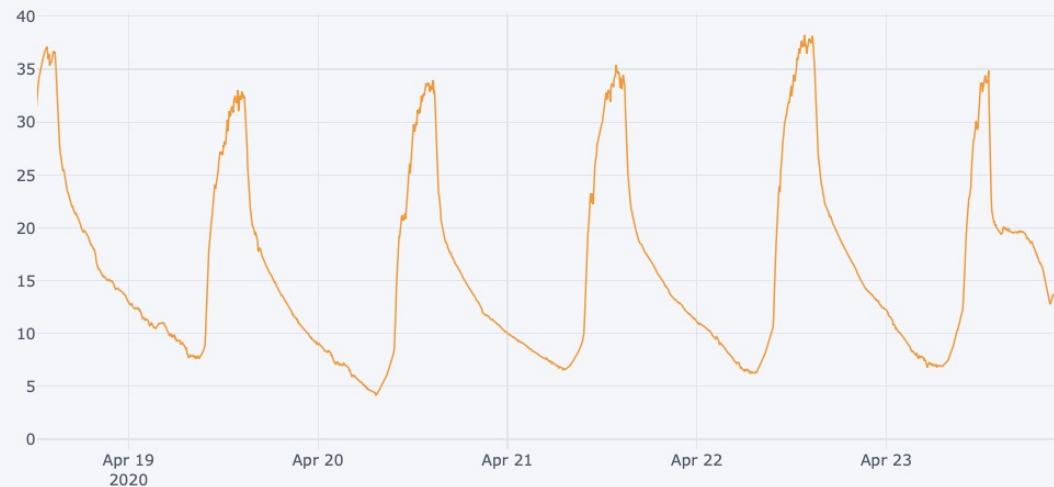
### Datenvisualisierung

#### Gesamten Datensatz visualisieren

##### Erklärung

Nun wollen wir uns einen groben Überblick über den Datensatz verschaffen, indem wir uns eine Zeitreihendarstellung angucken. Dazu ruft man die Methode `iplot` mit dem Parameter `kind='line'` auf der Tabellenspalte 'value' auf:

```
df['value'].iplot(kind='line')
```



Bezeichnung des Wochentages sowie die gefilterten Datensätze des jeweiligen Wochentages enthalten.

[Export to plotly](#)

## Computational Essays in Jupyter Notebooks

- Jupyter Notebooks als Möglichkeit, Programmier-Umgebungen vorzubereiten, in denen SuS Data Analyses durchführen können
- Verschiedene Zell-Typen:
  - Technical-Code-Cells („unsichtbare“ Voraussetzungen für die Datenanalyse schaffen – z.B. Datensätze einlesen)
  - Output-Code-Cells (Output (z.B. Visualisierungen) generieren, womit Betrachter interagieren können)
  - Explaining-Markdown-Cells (Code/Prozess erläutern und Outputs interpretieren/kommentieren)

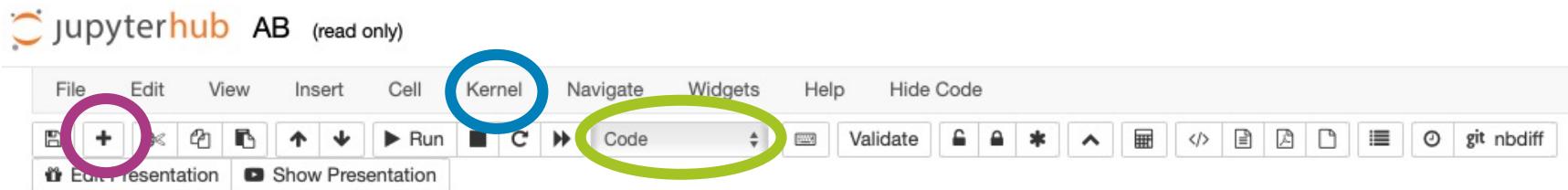
# Computational Essays in Jupyter Notebooks

- Jupyter Notebooks als Möglichkeit, Programmier-Umgebungen vorzubereiten, in denen SuS Data Analyses durchführen können



# Erstellen eines Computational Essays in Jupyter Notebooks

Ein paar grundlegende Tipps dazu:



1. Neue Zellen erstellt man über das „+“-Symbol oben in der Leiste
2. Über das Dropdown-Menü lässt sich der Zell-Typ von Code- auf Markdown-Zelle und umgekehrt ändern
3. Um allen Output zu löschen, klickt man im Menü auf „Kernel“ => „Restart & Clear Output“

## Epistemisches Programmieren

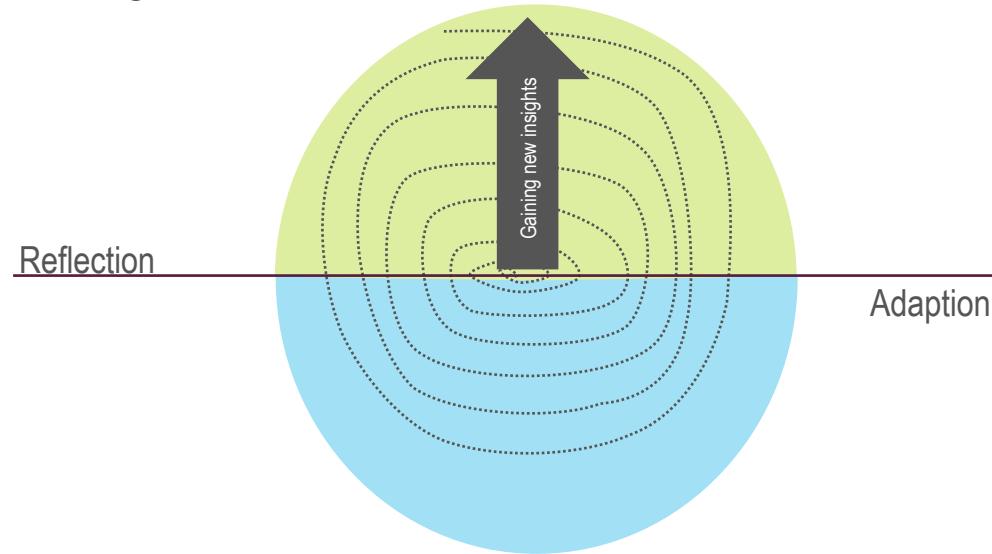
Wie läuft so ein epistemischer Programmierprozess im Kontext des Erstellens von Jupyter Notebooks als Computational Essays ab?

- Ziel: Bedeutungsvolle/Interessante Erkenntnisse
  - Bedarf an Scaffolds/Unterstützungsmöglichkeiten
    - Nutzung mächtiger Bibliotheken
    - Adaption von Worked Examples:
- Worked Examples:
  - Zeigen eine (professionelle) Problemlösung auf, die von den SuS betrachtet/untersucht werden können
  - Darauf aufbauend: Adaption/Erweiterung dieser Produkte zu individualisierten Lösungen zu (individuellen) Problemen
  - Siehe auch:  
Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research*, 70(2), 181–214. <https://doi.org/10/csm67w>

## Epistemisches Programmieren

Wie läuft so ein epistemischer Programmierprozess ab?

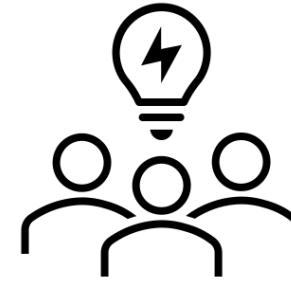
- Vorgehen: Trial & Error, Ausprobieren und Tüfteln:
  - Grundsätzliche Idee: Erkenntnisse und Erkenntnissinteressen entwickeln sich aus einem Prozess des Ausprobierens und sequentiellen Anpassens von Programmcode und der Untersuchung der sich daraus ergebenen Programmierergebnisse:



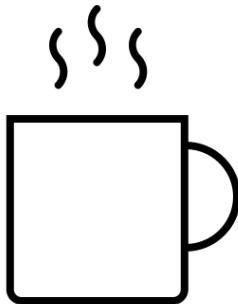
# Epistemisches Programmieren

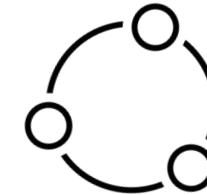
Wozu das Ganze überhaupt?

- Verdeutlichen: Programmieren lernen nicht nur zum Selbstzweck/um des Programmierens Willen, sondern weil ich es als **Werkzeug** benutzen kann, um **mir die Welt zu erschließen** und die so gewonnenen **Erkenntnisse mit anderen teilen** kann
- Aufzeigen des Bildungsgehalts, der durch Programmieren vermittelt wird
- Die realweltliche Anbindung hört nicht irgendwann auf, sondern ist hier dauerhaft zentraler Bestandteil
- Programmieren als Möglichkeit verstehen, eigene Interessen zu verfolgen und sich entsprechendes Wissen anzueignen
- Dabei: Wissen ...
  1. aus dem (realweltlichen) Interessenbereich
  2. bezogen auf das Programmieren/Adaptieren/Umgehen mit digitalen Werkzeugen an sich



# (Kaffee-)Pause (eine Ebene höher)



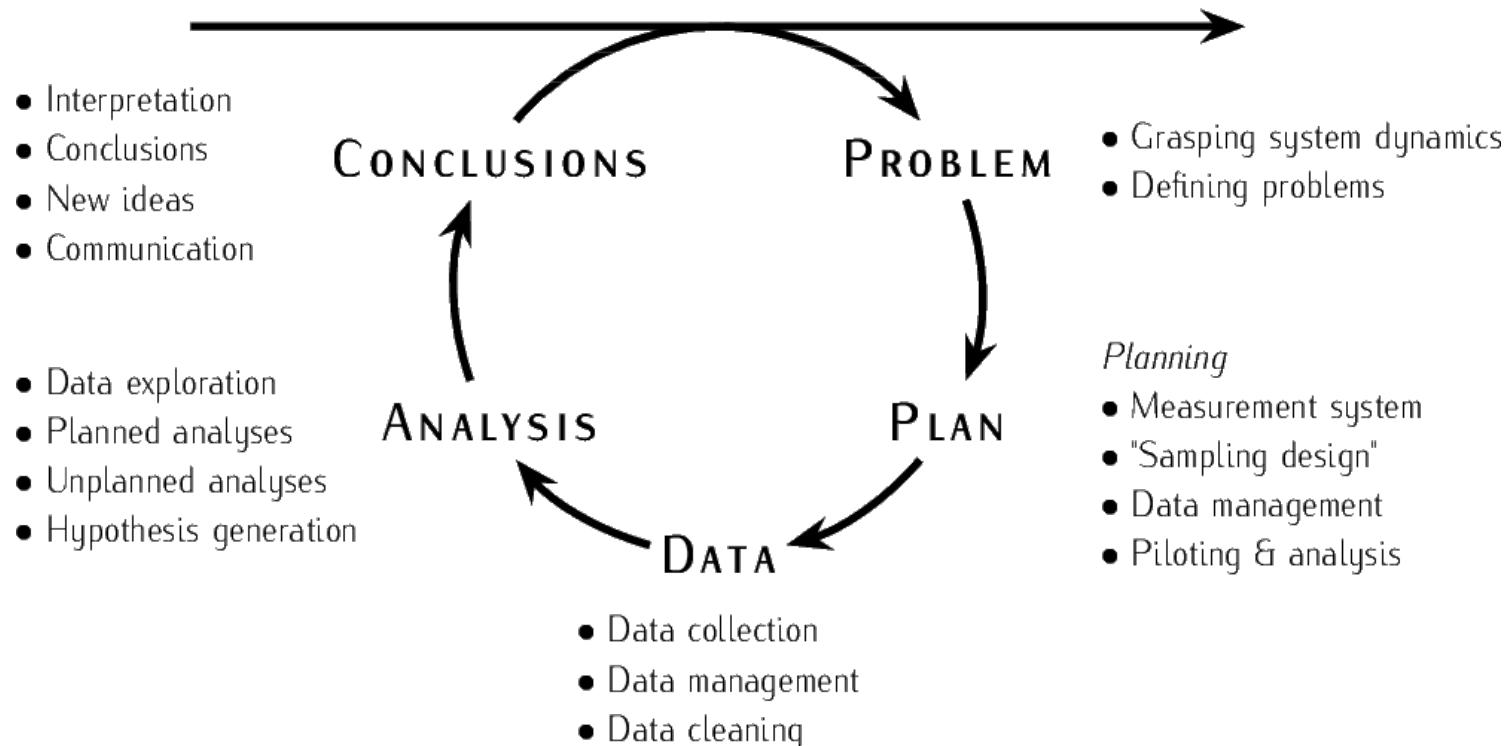


# Theoretischer Input: PPDAC-Cycle



## Datenexploration mit System: Der PPDAC Kreislauf

### DIMENSION 1: The Investigative Cycle (PPDAC)



Wild, C., & Pfankuch, M. (1999). Statistical Thinking in Empirical Enquiry. *International Statistical Review*, 67(3), 223-265.  
Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny

# Datenprojekte mit System

Ziel: Durchführen eines Datenanalyseprojekts und Präsentation

Epistemisch programmiert: Neue Erkenntnisse

Erstellung von Computational Essays  
Entlang Worked Examples

Datenanalyse mit Jupyter Notebooks, Exploration der Fragestellung  
Erstellen von aussagekräftigen Graphiken

Statistische Auswertungsmethoden:  
Diagramme  
Kennzahlen  
Filter

Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny

Generierung von statistischen Fragestellungen  
Was sind gute stat. Fragestellungen?

Generieren von statistischen Fragestellungen rund um Luftqualität im Klassenzimmer

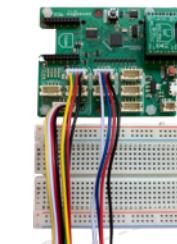
Problem

(C) Konklusion

Plan

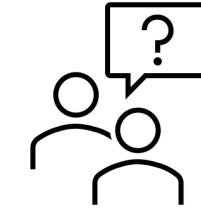
Analyse

Daten

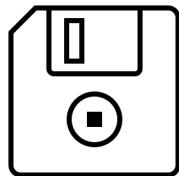


Import der Umwelt-Daten in Jupyter Notebook, Bereinigung der Daten

Datenanalyse mit Jupyter Notebooks und Python



# Workshop-Phase 1: Computational Essay – Fragestellung und Daten einlesen



# Datenprojekte mit System

Ziel: Durchführen eines Datenanalyseprojekts und Präsentation

Epistemisch programmiert: Neue Erkenntnisse

Erstellung von Computational Essays  
Entlang Worked Examples

(C) Konklusion

Datenanalyse mit Jupyter Notebooks, Exploration der Fragestellung

Erstellen von aussagekräftigen Graphiken

Statistische Auswertungsmethoden:  
Diagramme  
Kennzahlen  
Filter

Analyse

Problem

Generieren von statistischen Fragestellungen rund um Luftqualität im Klassenzimmer

Daten

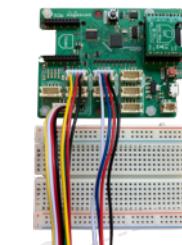
Plan

Datenerhebung planen und durchführen

Import der Umwelt-Daten in Jupyter Notebook, Bereinigung der Daten

Datenanalyse mit Jupyter Notebooks und Python

Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny



## Die Phase „Problem“ oder Worum geht es eigentlich?

- Luftqualität in unserem Raum

### >>Problembeschreibung<<

Eine gute Raumluftqualität ist Voraussetzung dafür, dass sich Menschen in Innenräumen gut konzentrieren können (siehe [Empfehlungen des österreichischen Bundesministeriums für Bildung und Frauen und der Plattform MeineRaumluft](#)).

Da insbesondere in der Schule viele Menschen für eine lange Zeit in einem Klassenraum lernen, sollte besonders auch hier sichergestellt werden, dass die Raumluftqualität gut ist, sodass sich die Personen im Klassenraum gut konzentrieren können.

In dieser Analyse wollen wir deshalb in unseren Klassenraum schauen und anhand von Daten zur unserer Raumluft überprüfen, wo es möglicherweise Verbesserungsbedarf gibt. Aufbauend darauf sollen dann Handlungsmöglichkeiten gefunden werden, um die Raumluftqualität in unserem Klassenraum zu verbessern.

## Die Phase „Problem“ und Die richtigen Fragen stellen!

| Kriterien für Fragestellungen  | Erfüllt? |
|--|----------|
| Ist die Frage für Dich und deine Mitschüler:innen relevant?                        |          |
| Lassen sich durch Beantwortung der Frage Erkenntnisse über das Raumklima gewinnen? |          |
| Ist die Frage klar und eindeutig gestellt?   |          |
| Vermeidet die Frage „ja“/„nein“ Antworten?   |          |
| Kannst Du Deine Frage mit den gesammelten Daten beantworten?                       |          |

→ Fragen stellen nach der Think-Pair-Share Methode!

# Die Phase „Problem“ oder Die richtigen Fragen stellen!

## Forschungsfragen:

- Wie sieht der zeitliche Verlauf der CO<sub>2</sub>-Werte an den verschiedenen Wochentagen in meinem Klassenraum aus?
- Zu welchen Zeitpunkten liegt der CO<sub>2</sub>-Wert oberhalb der empfohlenen Grenze von 1000 ppm?
  - Welche Ursachen könnte dies haben?
  - Was können wir tun?

| Kriterien für Fragestellungen  | Erfüllt? |
|--|----------|
| Ist die Frage für Dich und deine Mitschüler:innen relevant?                        |          |
| Lassen sich durch Beantwortung der Frage Erkenntnisse über das Raumklima gewinnen? |          |
| Ist die Frage klar und eindeutig gestellt?   |          |
| Vermeidet die Frage „ja“/„nein“ Antworten?   |          |
| Kannst Du Deine Frage mit den gesammelten Daten beantworten?                       |          |

## Die Think-Pair-Share Methode

**Phase „Think“ (einzelne oder zu zweit)**

Formulieren von ein bis zwei Fragen!

**Phase „Pair“ (jeweils zwei Personen oder Paare)**

1. (Tausch) Fragen an ein anderes Paar weitergeben, diese geben Feedback zur Verbesserung der Fragen
2. (Rücktausch) Einarbeiten des Feedbacks

**Phase „Share“ (alle gemeinsam)**

1. Besprechen ausgewählter Fragen im Plenum
2. (Eventuell) Einarbeiten von Feedback

## Die Think-Pair-Share Methode

Wir sammeln heute Daten zur Raumluftqualität hier.

**Bitte erstellen Sie 1-2 Forschungsfragen, die später mit den Daten beantwortet werden können!**

Dazu: Think-Pair-Share Methode!

# Datenprojekte mit System

Ziel: Durchführen eines Datenanalyseprojekts und Präsentation

Epistemisch programmiert: Neue Erkenntnisse

Erstellung von Computational Essays  
Entlang Worked Examples

(C) Konklusion

Problem

Generierung von statistischen Fragestellungen  
Was sind gute stat. Fragestellungen?

Generieren von statistischen Fragestellungen rund um Luftqualität im Klassenzimmer

Datenanalyse mit Jupyter Notebooks, Exploration der Fragestellung

Erstellen von aussagekräftigen Graphiken

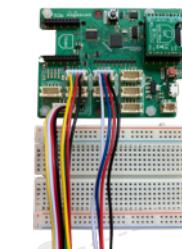
Analyse

Statistische Auswertungsmethoden:  
Diagramme  
Kennzahlen  
Filter

Daten

Plan

Datenerhebung planen und durchführen



Import der Umwelt-Daten in Jupyter Notebook, Bereinigung der Daten

Datenanalyse mit Jupyter Notebooks und Python

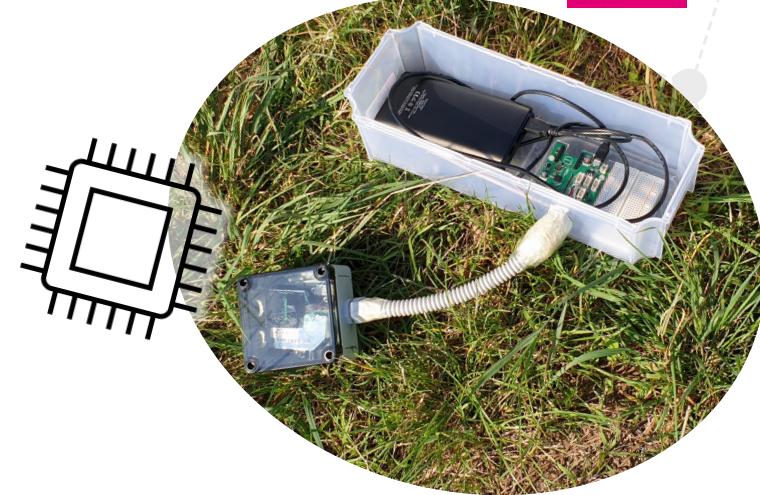
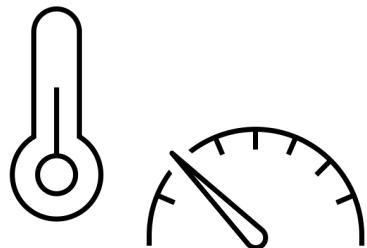
## Die Phase „Plan“

- **Welche Daten benötigen wir?**
- → Welche Daten wollen wir erheben?
- Wie können wir die Daten erheben?

Messprotokoll:

- Wo soll die Sensebox aufgestellt werden? (Ort im Raum; Höhe, etc.)
- Wie lange sollen Daten gesammelt werden?

# Input: Sensebox – Programmieren und Aufbauen

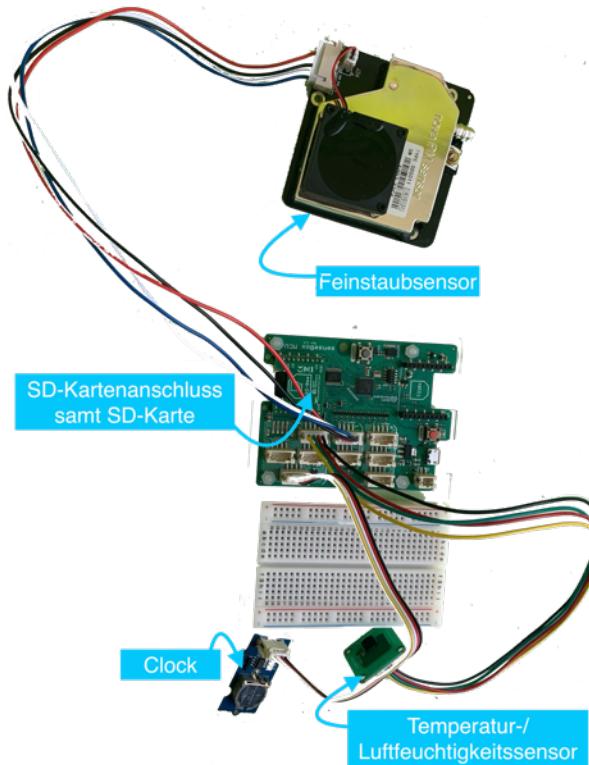


## Input: Sensebox – Programmieren und Aufbauen

Sensebox:

- Mikrocomputer (Arduino), an den sich mehrere Sensoren anschließen lassen,  
z.B.:
  - Temperatursensor,
  - Luftfeuchtigkeitssensor
  - Luftdrucksensor
  - Helligkeitssensor
  - CO<sub>2</sub>-Sensor
  - Feinstaubsensor
  - ...

## Input: Sensebox – Programmieren und Aufbauen



## Input: Sensebox – Programmieren und Aufbauen

Wie wir die Sensebox nutzen:

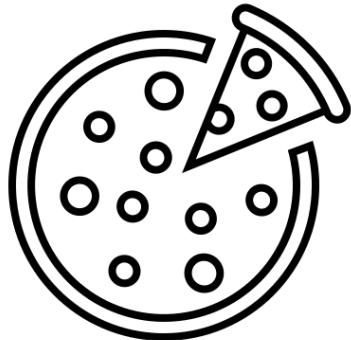
- Sammlung an Daten in bestimmten definierten Abständen
- Speicherung der Daten auf einer SD-Karte oder alternativ Senden der Daten via Wifi oder Bluetooth
- Anzeige der Daten auf einem Display

Programmierung der Sensebox:

- Mithilfe von Blockly: <https://blockly-react.netlify.app>
  - In Blocksprache; Ähnlich zu Scratch  
(einfacher Einstieg, aber Limitierung bezüglich der Individualisierung)
- Mit der Arduino-Software: <https://www.arduino.cc/en/software>
  - In (geschriebenem) Programmcode  
(komplexer, aber dafür stärker individualisierbar)
- Hybrid (zunächst in Blockly und dann detaillierter in der Arduino-Software)

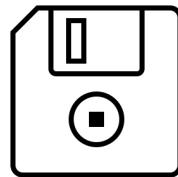


# Mittagspause 😊





# Kurzinput: OpenSenseMap und Daten einlesen



## Input: OpenSenseMap als Alternative zur Programmierung der Sensebox

OpenSenseMap:

- hilfreich, um Daten für die Datenauswertung zu finden
- Plattform für offene Sensordaten, auf der jede\*r Daten veröffentlichen und sammeln kann
- Zugriff auf Daten von aktuell mehr als 5000 Sensebox in Deutschland und 7500 weltweit
- entwickelt im Rahmen von GI@School am Institut für Geoinformatik in Münster, wo es zusammen mit der senseBox eingesetzt wird
- Link: <https://www.opensensemaps.org>
- => Mehr dazu am 23.03.2022

## Kurzinput: Daten einlesen

- Schritt 1: Daten als csv-Datei hochladen (durch Klick auf „Upload“)



Visit repo Copy Binder link Quit

Files Running Clusters

Select items to perform actions on them.

Upload New ▾

| <input type="checkbox"/> | 0 | <input type="button" value=".."/>                           | / JupyterNotebooks / Unterrichtsreihe / WorkedExamples | Name | Last Modified         | File size |
|--------------------------|---|---|--|------|-----------------------|-----------|
| <input type="checkbox"/> |   | <input type="button" value=".."/>                           |  |      | vor ein paar Sekunden |           |
| <input type="checkbox"/> |   | <input type="button" value="res"/>                          |  |      | vor 23 Minuten        |           |
| <input type="checkbox"/> |   | <input type="button" value="Basisinformationen.ipynb"/>     |  |      | vor 23 Minuten        | 2.68 kB   |
| <input type="checkbox"/> |   | <input type="button" value="ComputationalEssay_CO2.ipynb"/> |  |      | vor 23 Minuten        | 65.2 kB   |
| <input type="checkbox"/> |   | <input type="button" value="data_complete_hPa.csv"/>        |  |      | vor 23 Minuten        | 1.07 MB   |

## Kurzinput: Daten einlesen

- Schritt 2: Im Jupyter Notebook müssen (nach Import der benötigten Bibliotheken) die Daten in einem sog. DataFrame gespeichert werden – dazu gibt man in der Methode `read_csv` den Pfad an, unter dem sich die Daten befinden (hier liegen sie im selben Ordner):

```
In [ ]: import pandas as pd
import numpy as np
import plotly.graph_objects as go
import plotly.subplots as sp
from datetime import datetime as dati
```

### Daten einlesen

Nun müssen wir erst einmal die Daten hier in das Jupyter Notebook hineinbekommen, um diese dann später analysieren und interpretieren zu können. Dazu lesen wir die Daten als sogenannte **csv-Datei** (csv = comma-separated values, zu Deutsch: durch Komma getrennte Werte) in ein **Dataframe** ein, welches wir `df` nennen. Ein Dataframe kann man sich vorstellen wie eine Tabelle, in der die Zeilen jeweils den aufgenommenen Datensätzen entsprechen und die Spalten die einzelnen Attribute/Variablen - wie z.B. "Datum" oder "gemessener Temperaturwert" darstellen.

```
In [ ]: df = pd.read_csv('data_complete_hPa.csv', sep =';')
```

Navigation zu Unterordner mit NameUnterordner/  
Navigation zu übergeordnetem Ordner mit ../

## Kurzinput: Daten einlesen

- Schritt 3: Um zu prüfen, ob die Daten korrekt eingelesen wurden, kann die Methode `head()` angewandt werden, um die ersten 5 Datensätze anzuzeigen:

### Daten einlesen

Nun müssen wir erst einmal die Daten hier in das Jupyter Notebook hineinbekommen, um diese dann später analysieren und interpretieren zu können. Dazu lesen wir die Daten als sogenannte **csv-Datei** (csv = comma-separated values, zu Deutsch: durch Komma getrennte Werte) in ein **Dataframe** ein, welches wir `df` nennen. Ein Dataframe kann man sich vorstellen wie eine Tabelle, in der die Zeilen jeweils den aufgenommenen Datensätzen entsprechen und die Spalten die einzelnen Attribute/Variablen - wie z.B. "Datum" oder "gemessener Temperaturwert" darstellen.

```
In [2]: df = pd.read_csv('data_complete_hPa.csv', sep =';')
```

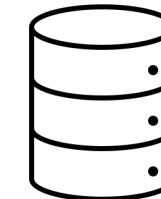
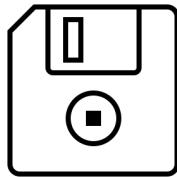
Um zu überprüfen, ob das Einlesen der Daten funktioniert hat, können wir uns mithilfe der Methode `head()` die ersten 5 Zeilen - also Datensätze - des Dataframes anzeigen lassen:

```
In [3]: df.head()
```

Out [3]:

|   | timestamp                | value_co2 | value_temp | value_hum | value_airpressure | value_light |
|---|--------------------------|-----------|------------|-----------|-------------------|-------------|
| 0 | 2022-02-08T09:31:36.000Z | 1008      | 24.26      | 34.57     | 1010.9109         | 0           |
| 1 | 2022-02-08T09:32:06.000Z | 1014      | 24.35      | 34.16     | 1010.8366         | 48          |
| 2 | 2022-02-08T09:32:36.000Z | 1023      | 24.52      | 33.95     | 1010.8469         | 48          |
| 3 | 2022-02-08T09:33:06.000Z | 1028      | 24.54      | 33.75     | 1010.8881         | 48          |
| 4 | 2022-02-08T09:33:37.000Z | 1047      | 24.42      | 33.92     | 1010.8738         | 49          |

# Workshop-Phase 2: Computational Essay – Daten einlesen



# Datenprojekte mit System

Ziel: Durchführen eines Datenanalyseprojekts und Präsentation

Epistemisch programmiert: Neue Erkenntnisse

Erstellung von Computational Essays  
Entlang Worked Examples

(C) Konklusion

Problem

Generierung von statistischen Fragestellungen  
Was sind gute stat. Fragestellungen?

Generieren von statistischen Fragestellungen rund um Luftqualität im Klassenzimmer

Datenanalyse mit Jupyter Notebooks, Exploration der Fragestellung

Erstellen von aussagekräftigen Graphiken

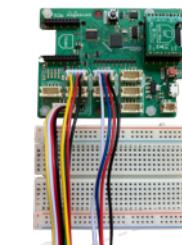
Analyse

Plan

Statistische Auswertungsmethoden:  
Diagramme  
Kennzahlen  
Filter

Daten

Datenerhebung planen und durchführen



Import der Umwelt-Daten in Jupyter Notebook, Bereinigung der Daten

Datenanalyse mit Jupyter Notebooks und Python

## Workshop-Phase 2: Daten einlesen

Link zum Worked Example: [tinyurl.com/prodabi-fobi-m4](http://tinyurl.com/prodabi-fobi-m4)

Alternativlink:

<https://mybinder.org/v2/gh/svenhuesing/Fortbildung-M4.git/main?urlpath=tree>

(Unterordner JupyterNotebooks → Unterrichtsreihe → WorkedExamples → ComputationalEssay\_CO2.ipynb)

Macht Euch mit der Jupyter-Umgebung vertraut und arbeitet Worked Example bis zur Code-Zelle vor der >>Datenanalyse<< durch.

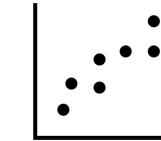
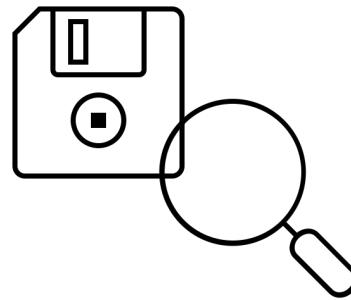
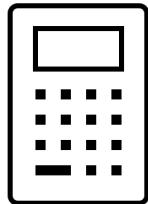
Nutzt die Datei `Eigenes_Computational_Essay.ipynb` im Ordner JupyterNotebooks → Unterrichtsreihe, um dort die hier im Raum erhobenen Daten einzulesen. Übertragt zuvor auch Eure Forschungsfrage(n) in Euer Computational Essay.

Die Daten im Unterordner res und haben den Dateinamen `daten_fobi.csv`

Zeit bis ca. 11:45

# Workshop-Phase 3:

## Computational Essay – Daten analysieren

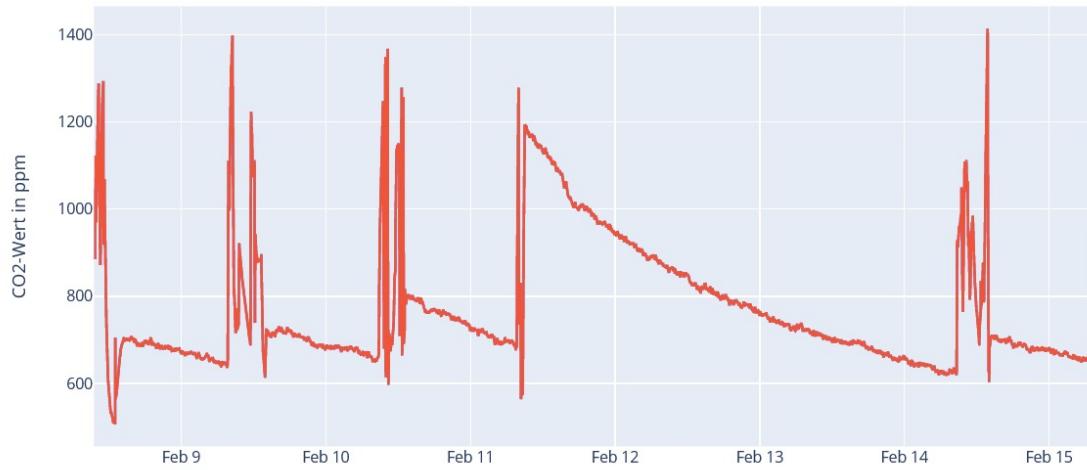


## Daten analysieren - Zeitreihen

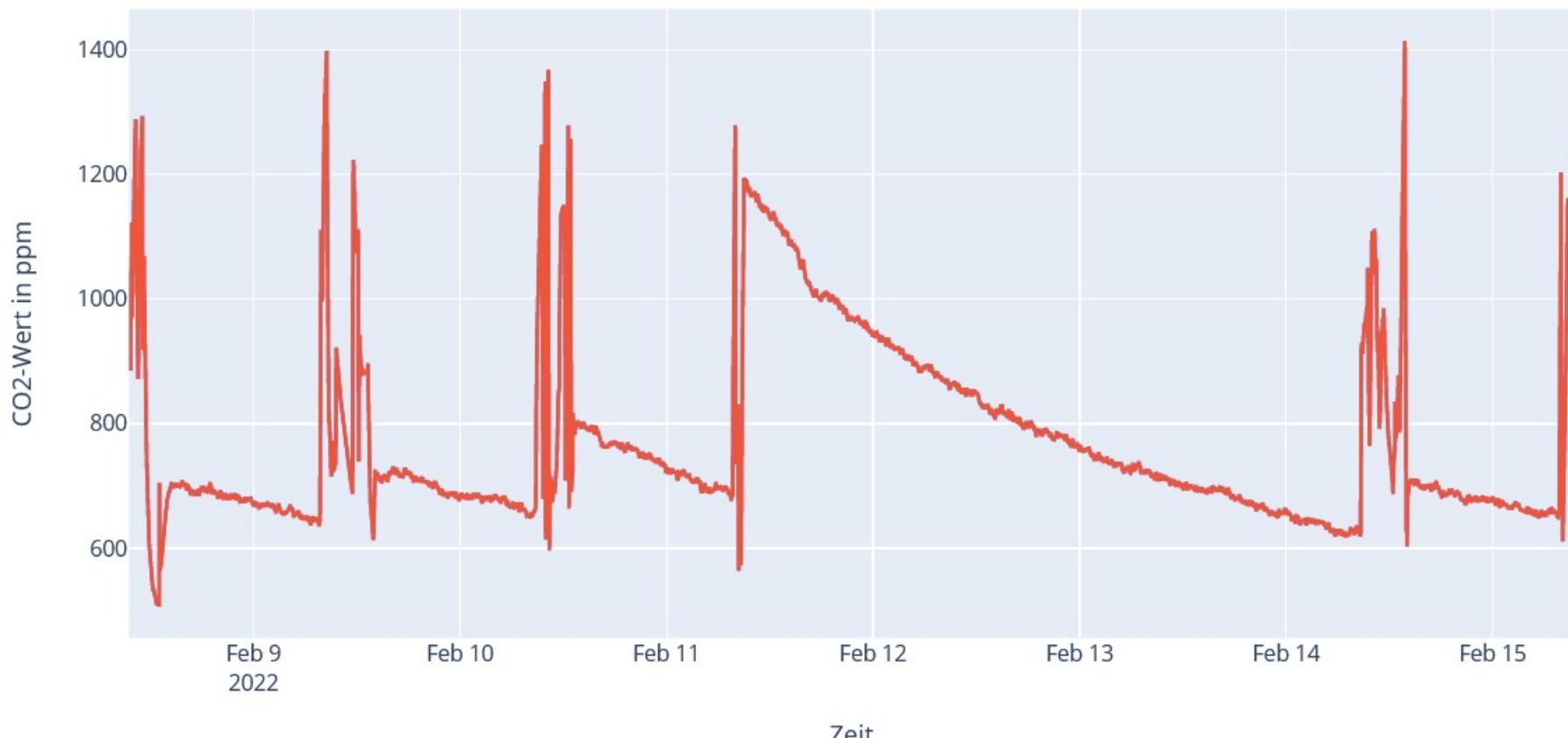
Mögliche Strategien

- Suchen nach Ausreißern und Sprüngen
- Eingrenzen auf bestimmte Zeiträume (Filtern)
- Suchen nach Werten ober- oder unterhalb von Grenzwerten
- Suchen nach Mustern oder zyklischen Änderungen

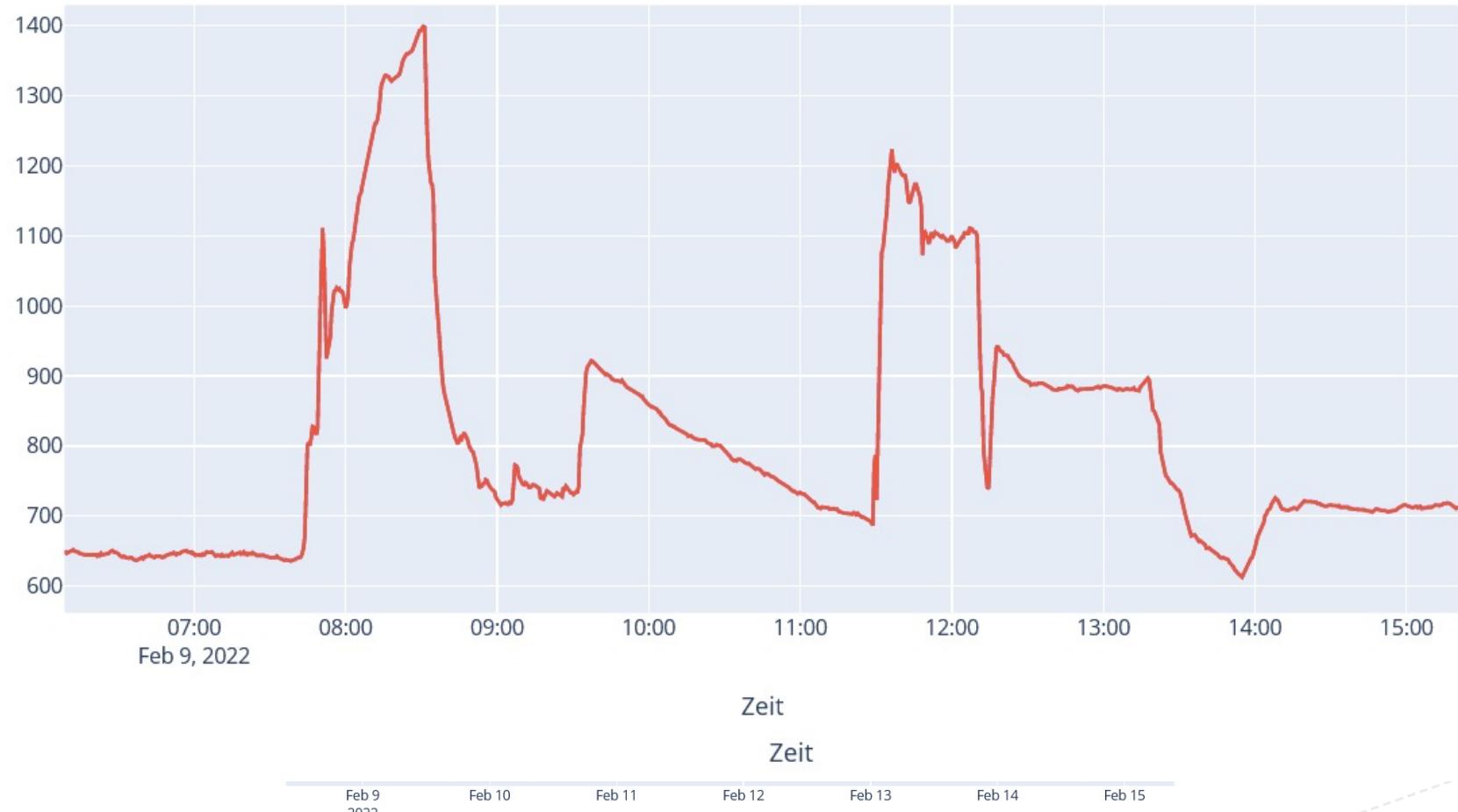
CO2-Werte in unserem Klassenraum



## CO2-Werte in unserem Klassenraum



## CO2-Werte in unserem Klassenraum



## Workshop-Phase 3: Daten analysieren

In der folgenden Workshop-Phase wollen wir eine eigene Datenanalyse auf Grundlage des Worked Examples erstellen. Die verwendeten Methoden und Visualisierungen sind dabei eng mit der eigenen Forschungsfrage verknüpft. Konkret wollen wir uns die folgenden Aspekte einer Datenanalyse angucken:

- Erstellen von Visualisierungen
  - Streudiagramme
  - Balkendiagramme
  - Boxplots
  - ...
- Daten filtern
- Einzeichnen mehrerer Graphen in ein Koordinatensystem
- Hinzufügen zusätzlicher Trennlinien
- Gruppieren von Daten
- Visualisierungen zum Vergleich mehrerer Variablen
- Korrelation

# Datenprojekte mit System

Ziel: Durchführen eines Datenanalyseprojekts und Präsentation

Epistemisch programmiert: Neue Erkenntnisse

Erstellung von Computational Essays  
Entlang Worked Examples

Generierung von statistischen Fragestellungen  
Was sind gute stat. Fragestellungen?

Generieren von statistischen Fragestellungen rund um Luftqualität im Klassenzimmer

(C) Konklusion

Problem

Plan

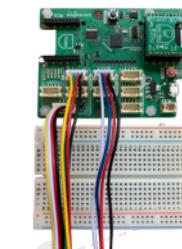
Daten

Analyse

Datenanalyse mit Jupyter Notebooks, Exploration der Fragestellung  
Erstellen von aussagekräftigen Graphiken

Statistische Auswertungsmethoden:  
Diagramme  
Kennzahlen  
Filter

Datenerhebung planen und durchführen



Import der Umwelt-Daten in Jupyter Notebook, Bereinigung der Daten

Datenanalyse mit Jupyter Notebooks und Python

Sven Hüsing, Harald Neubeck, Denise Pallerberg, Susanne Podworny

## Workshop-Phase 3: Daten analysieren

Link zum Worked Example: [tinyurl.com/prodabi-fobi-m4](http://tinyurl.com/prodabi-fobi-m4)

Alternativlink:

<https://mybinder.org/v2/gh/svenhuesing/Fortbildung-M4.git/main?urlpath=tree>

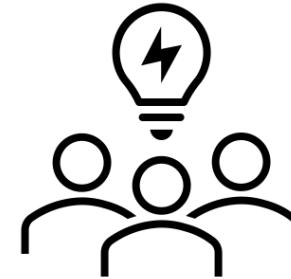
(Unterordner JupyterNotebooks → Unterrichtsreihe → WorkedExamples → ComputationalEssay\_CO2.ipynb)

Macht Euch mit der Jupyter-Umgebung vertraut und arbeitet das Worked Example bis zum Ende durch.

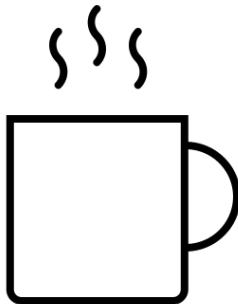
Analysiert die Daten im Hinblick auf Eure Forschungsfrage in Eurem eigenen Computational Essay und probiert dabei die verschiedenen Visualisierungen und Auswertungsmöglichkeiten aus.

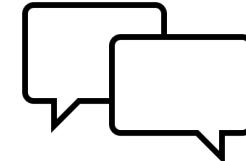
Haltet am Ende Eures Computational Essays mindestens zwei Erkenntnisse fest, die Ihr durch die Analyse der Daten gewinnen konntet. Formuliert ebenso Handlungsempfehlungen, um das Raumklima zu verbessern.

Zeit bis ca. 14:45 Uhr

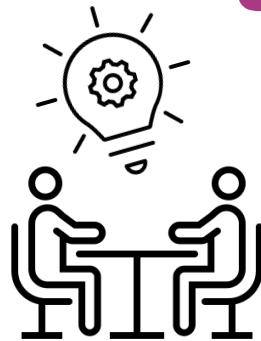


# (Kaffee-)Pause (eine Ebene höher)





# Workshop-Phase 4: Ergebnispräsentation und Diskussion



# Datenprojekte mit System

Ziel: Durchführen eines Datenanalyseprojekts und Präsentation

Erstellung von Computational Essays  
Entlang Worked Examples

Epistemisch programmiert: Neue Erkenntnisse

(C) Konklusion

Datenanalyse mit Jupyter Notebooks, Exploration der Fragestellung  
Erstellen von aussagekräftigen Graphiken

Statistische Auswertungsmethoden:  
Diagramme  
Kennzahlen  
Filter

Datenanalyse mit Jupyter Notebooks und Python

Import der Umwelt-Daten in Jupyter Notebook, Bereinigung der Daten

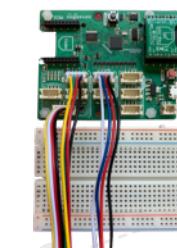
Generierung von statistischen Fragestellungen  
Was sind gute stat. Fragestellungen?

Problem

Generieren von statistischen Fragestellungen rund um Luftqualität im Klassenzimmer

Plan

Datenerhebung planen und durchführen



## Die Phase „Konklusion“

### Ergebnisse formulieren

Was haben wir herausgefunden?

Wie können wir unser Verhalten ändern? (Handlungsempfehlungen)

### Präsentieren der Ergebnisse

Jupyter Notebook als Computational Essay

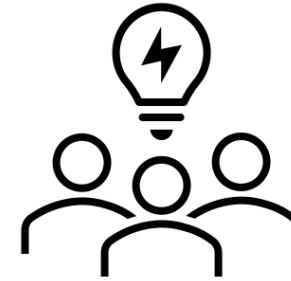
Die **Checkliste** kann helfen!

Link zur Checkliste: [tinyurl.com/prodabi-fobi-m4](https://tinyurl.com/prodabi-fobi-m4)

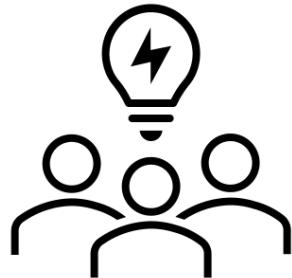
Alternativlink:

<https://mybinder.org/v2/gh/svenhuesing/Fortbildung-M4.git/main?urlpath=tree>

Unterordner Material

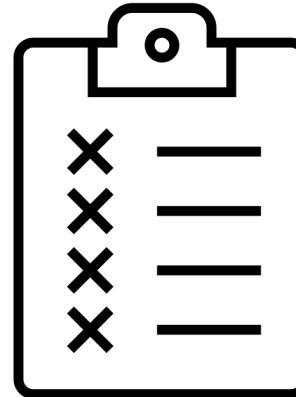


# Abschlussdiskussion



## Abschlussbefragung

Wir sind Euch sehr dankbar, wenn Ihr zur Evaluation der Fortbildung den Fragebogen hinter folgendem Link ausfüllt: [go.upb.de/m4-eval](http://go.upb.de/m4-eval)

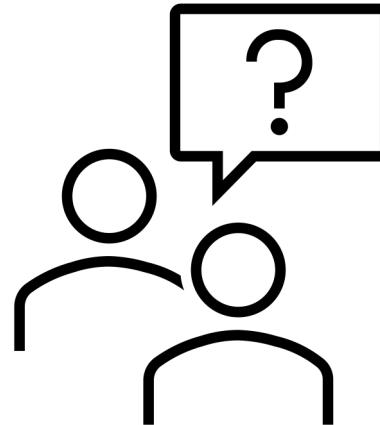


Seid gerne ehrlich – es gibt kein „richtig“ und „falsch“ ☺

Vielen Dank – Ihr helft uns damit sehr bei der Überarbeitung der Fortbildungsreihe für zukünftige Durchführungen.

## Abschluss

Noch Fragen?



Wir sehen uns wieder am 16.03.2022 (14:30-16:00 Uhr) – online

Weitere Infos folgen per Mail (inkl. Link).

Vielen Dank, dass Ihr heute dabei wart ☺ Habt eine gute Zeit

## Kontakt

- Bei weiteren Fragen, schreibt uns gerne eine Mail an

[prodabi@campus.upb.de](mailto:prodabi@campus.upb.de)

- ...oder besucht uns im Internet unter

[www.prodabi.de](http://www.prodabi.de)

- ...oder tragt Euch in unsere Mailingliste ein für weitere Informationen zu Veröffentlichungen von Material, Lehrkräftefortbildungen, Workshops etc.:

<https://lists.uni-paderborn.de/mailman/listinfo/prodabi-schule>