

Data Club @yourSchool Python

Inhaltsverzeichnis

<u>Genereller Ablauf</u>	2
<u>Hinweise zum Konzept</u>	2
<u>Lernziele</u>	2
<u>Technologie / Repl.it</u>	2
<u>Konzept</u>	3
<u>Begrüßung und generelle Motivation (~15 Min) (Plenum; Hauptinspirer)</u>	3
<u>Vorstellungsrunde (Kleingruppe)</u>	3
<u>Programmieren (~ 3h + 30 min Pause) (in der Kleingruppe)</u>	3
<u>Interne Abschlussrunde (Kleingruppe, ~5 Minuten)</u>	8
<u>Abschlussrunde (Plenum, ~20 min)</u>	8

Genereller Ablauf

- Der Kurs beginnt mit der Begrüßung im Plenum. Ein erfahrener Inspirer übernimmt die Einführung in das Thema und versucht sicherzustellen, dass die Technik bei den Meisten funktioniert. Danach werden die Kinder in kleine Teams eingeteilt. Jeder Inspirer ist während dem Kurs für etwa 4 bis 5 Kids verantwortlich.
- Die eingeteilten Teams werden in Breakout-Rooms (bei Zoom) eingeteilt. Wenn ihr in den Breakout Rooms Fragen habt, ruft gerne über Zoom um Hilfe. Der Haupt-Inspirer wird dann zu euch in den Breakout-Room kommen.
- Vorstellungsrunde in den Gruppen. Stellt euch einmal vor und erklärt gerne, wie ihr zu der IT gekommen seid. Lasst die Kids sich vorstellen, inklusive
- Programmiererfahrung, Erwartungen und Motivation.
- Jetzt geht's ans Eingemachte: Programmieren! Mehr Informationen zu den Einheiten findet ihr unten.
- Versucht nach etwa der Halbzeit eine Pause von 30 Minuten einzulegen.
- Der Abschluss findet wieder im Plenum mit der gesamten Klasse statt. Hier stellt sich jeder Inspirer einmal kurz vor und erklärt, warum er*sie für IT brennt.

P.S.: Vergesst nicht, eure Pausen zu machen :)

Hinweise zum Konzept

Lernziele

- Es geht darum, die Kids zu begeistern. Unser Ziel ist es, dass sie sich nach den 3 Stunden Kurs noch 12 Stunden hinsetzen und weiter recherchieren und programmieren. Wie das bei euch konkret aussieht, ist euch überlassen. Ihr merkt, die Kids haben voll Lust mit euch eine halbe Stunde über Roboter zu reden? Dann macht das gerne :)
- Versucht eher, den Kids grundlegende Konzepte beizubringen ("Wie debugge ich ein Programm?"), als ihnen diese eine Eigenheit von Python lang und breit zu erklären. Zeigt/erklärt ihnen auch gerne, wo das, was sie gerade lernen, im echten Leben zum Einsatz kommt.
- Lasst die Kids so viele Schritte wie möglich selbst gehen. Sie werden so am Meisten lernen, und sind hoffentlich auch in der Lage, an das Gelernte nach Ende des Kurses anknüpfen zu können.
- Lasst sie dabei auch scheitern. Im Optimalfall versucht ihr dann mit ihnen gemeinsam zu ergründen, wieso ihr Ansatz nicht funktioniert hat - dadurch müssen sie ihre Gedanken kommunizieren, und lernen so mehr.
- Dieses Konzept ist unser Vorschlag, wie man Kinder und Jugendliche begeistern könnte. Seht es also mehr als groben Leitfaden und weniger als festen Lehrplan.

Technologie / Repl.it

- Wir arbeiten während dem Kurs mit dem Online-Editor Repl.it -> <https://repl.it> • Wie genau die Kids sich einloggen und mit welchen Accounts sie arbeiten, erklärt der hauptverantwortliche Inspirer am Anfang im Plenum. Das ist also nicht eure Aufgabe, darum kümmern wir uns.
-

• Wir arbeiten während dem Kurs mit dem Online Editor Replit: <https://replit> - Wie genau die Kurs Seite einloggen und mit welchen Accounts sie arbeiten, erklärt der hauptverantwortliche Inspirer am Anfang im Plenum. Das ist also nicht eure Aufgabe, darum kümmern wir uns.

Konzept

Begrüßung und generelle Motivation (~15 Min) (Plenum; Hauptinspirer)

- Wer sind wir und was wünschen wir uns? (an die Kinder gerichtet)
 - Wir wollen, dass ihr alle etwas mitnehmen könnt. Uns ist es lieber, dass ihr Zusammenhänge verstanden habt, aber dafür weniger. (Wird trotzdem vorkommen, dass euch Dinge überfordern, aber das ist normal. Behaltet das trotzdem im Hinterkopf)
 - Vordergründig bringen wir euch heute Programmieren bei, aber eigentlich geht es uns darum, euch mit Menschen zusammenzubringen, die das richtige Mindset haben. Damit ihr von Ihnen lernen könnt. Wenn ihr inhaltlich überhaupt nichts verstehtet, aber dafür die Fehlerkultur mitnehmt, sind wir happy.
 - A propos Fehlerkultur: Wenn ihr etwas nicht versteht - fragt nach. Das hier soll alles werden, bloß kein Frontalunterricht, in dem ihr Dinge schlucken, verdauen, und nachher wieder auskotzen sollt. Uns geht es darum, dass ihr Spaß habt, und vor allem: etwas lernt. Und gerade dem stehen so ein paar Dinge im Weg, wie: Schon beim ersten kleinen Fitzelchen Syntax zu scheitern.
 - Also: Wenn ihr jetzt gleich bei euren Inspirern seid, und euer Programm nicht funktioniert, was macht ihr dann? Richtig. Fragt nach.
 - grober inhaltlicher Überblick
 - zeitlicher Überblick
 - Einmal kurz die Technologie durchgehen (repl.it)
-

Vorstellungsrunde (Kleingruppe)

- Stellt euch selber vor. (2-3 Sätze)
 - Was habt ihr gemacht, als ihr so alt wart wie die Teilnehmenden?
 - Lasst die Teilnehmenden sich vorstellen.
 - Wie heißt du?
 - Wieso bist du hier?
 - Hast du schonmal programmiert?
 - Was erhoffst du dir von diesem Kurs?
 - Wichtig: achtet darauf, dass die Kinder sich nur mit Vornamen vorstellen (Datenschutz :))
-

Programmieren (~ 3h + 30 min Pause) (in der Kleingruppe)

Hallo Welt!

- Teilt euren Bildschirm und programmiert Hallo Welt.
 - Erklärt gerne Zeichen für Zeichen
 - Warum kommen da jetzt Klammern hin? - Und welche?
 - Was bedeuten die "" dort?
 - Was ist eine Konsole?
 - Wie führe ich das Programm aus? (Ggf. auch auf mobilen Endgeräten)
 - Lasst die Kinder, nachdem ihr fertig seid, selber ausprobieren eigene Texte auf der Konsole auszugeben.
-

Was sie lernen

- print()
 - Strings
 - indirekt Funktionsaufrufe (muss aber noch nicht erklärt werden, darauf könnt ihr aber später zurückkommen)
-

- indirekt Funktionsaufrufe (muss aber noch nicht erklärt werden, darauf könnt ihr aber später zurückkommen)
- Was die Konsole macht/ist
- Wie ich ein Programm starte

Code

```
print("Hallo Welt!")
```

Hallo <Name>!

- Teilt euren Bildschirm und fügt den Code unten zu dem bereits bestehendem Hallo Welt Programm hinzu.
- Lasst die Kinder, nachdem ihr fertig seid, selber ausprobieren eigene kreative Fragen und angepasste Antworten auf der Konsole auszugeben.

Was sie lernen

- Variablendeklaration mit `<Variablenname> = <Wert>`
- mit `input("")` eine Frage stellen
- Wieder auf Variablen zugreifen

Code

```
name = input("Wie heißt du? ")
print("Hallo " + name)
```

If und Else

- Teilt euren Bildschirm und ersetzt das print() aus "Hallo, <Name>!" mit dem Code unten.

Was sie lernen

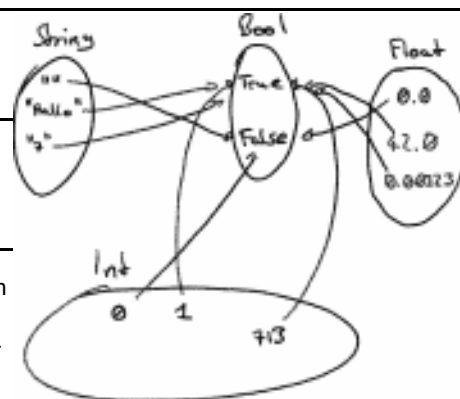
- Fallunterscheidungen mit `if <condition>: <then> else: <else>`
- was eine Condition (Fallunterscheidung) ist
- Das jeweils nur <then> (exklusives) oder <else> ausgeführt wird.

Code

```
if name == "Inspirer":
    print("Oh, dich habe ich doch schon einmal gesehen!")
else:
    print("Hallo, " + name)
```

Datentypen und Funktionen

- Jetzt wird es ein bisschen komplizierter, und die Kids müssen ihre Intuition testen.
- Optional: "Was steht da eigentlich in Wirklichkeit hinter dem if?" -> lasst sie andere Ausdrücke ausprobieren, die sie aus der Mathematik kennen, und/oder fragt sie, ob sie schon einmal etwas von logischen Operatoren gehört haben.
- Um dann schließlich das Ganze mit `True`, `False` und Datentypen (String, Integer, Float, Boolean) aufzulösen, und ihnen eine Funktionsdarstellung mit Mengendiagrammen aufzumalen. (über Bildschirm teilen und Paint/aggie.io/Zoom drawing Tool)



- Optional: Zeigt ihnen gerne auch, wie weit die Umwandlung in Python geht. Auch `1`, `0` und `""` haben Wahrheitswerte (`if 1: if "Hallo": if 0:`)
- Was ist eigentlich eine Funktion? (Sollte aus dem Matheunterricht bekannt sein, also könnt ihr die Kinder auch gerne fragen, was sie sich unter einer Funktion vorstellen) => Ein Ding, das, wenn man es fragt, einem etwas zurückgibt. Konkrete Repräsentation bei uns: Ein "Stück" Code, der anhand von Parametern einen Wert "berechnet" und zurückgibt. In unserem Fall: eine Zahl x * 2. Beziehungsweise einen Bool: True oder False.
- Die Funktion `malZwei()` wird in der Kleingruppe live gecodet.
- Lasst die Kids danach ihre erste eigene Funktion schreiben: `betrag(int zahl)` soll den Betrag der angegebenen Zahl zurückgeben. (optional: signum bzw. istPositiv)

Was sie lernen

Was sie lernen

- Wie man Funktionen definiert: `def <name>(*params): <body>`
- was ein Datentyp ist; welche Datentypen es u.a. gibt (Int, Float, String, Bool) • Funktionen sind mehr als nur Graphen in Koordinatensystemen (mehr als das, was sie bisher in Mathe hatten).

Code

```
# Diese Funktion wird live gecoded
```

```
def malZwei(x):
```

```
    return 2*x
```

```
# Könnt ihr bei interessierten Gruppen zusätzlich live coden (+erklären) def to_bool(x):
```

```
    if x == "":
```

```
        return False
```

```
    if x == 0:
```

```
        return False
```

```
    if x == None:
```

```
        return False
```

```
    else:
```

```
        return True
```

```
# malZwei(2) => 4
```

```
# malZwei(4) => 8
```

```
# Das sollten die Teilnehmenden am Ende dieser Einheit selber schreiben def betrag(x):
```

```
    if x < 0:
```

```
        return -x
```

```
    if x >= 0:
```

```
        return x
```

Wenn ihr hier bei in etwa der Hälfte der Zeit seid (und noch keine gemacht habt), würde sich hier eine größere Pause von 30 Minuten anbieten. Dann hätte man eine klare Einteilung zwischen fachlichem Input und Projektarbeit.

Input zu CO₂/Klimawandel

- Eine kleine Fragerunde zum Thema Klima/Klimawandel - was wissen die Kids schon, was fehlt ihnen?
 - Was stößt alles CO₂ aus, und wie viel?
 - Was ist ansonsten relevant?
 - Was ist mit den Kühen? -> gerne ein kleines Gespräch darüber beginnen :)
- kurzer Input zum Thema Treibhausgase:
 - Reflexion von Strahlung an den GreenHouse Gases in der Atmosphäre ○ mehr GHG in der Atmosphäre, mehr Reflexion -> die Atmosphäre wird wärmer
- Achtung! -> um klimaneutral zu werden, muss die gesamte Wirtschaft dekarbonisiert werden. Eine individuelle Reduktion des CO₂-Fußabdrucks ist zwar hilfreich, reicht aber keinesfalls aus.

Den Rest des Kurses werden wir mit [100 Punkte hat ein guter Tag](#) arbeiten. Die Idee ist, dass wenn wir als Menschheit klimaneutral leben wollen, wir alle im Schnitt nicht das Budget von 100 Punkten am Tag überschreiten sollten. Schaut euch die Seite gerne im Vorhinein schon einmal an und zeigt sie auch jetzt den Kindern.

Code lesen (in der Kleingruppe)

- Die erste "große" Funktion, die die Kids lesen und analysieren dürfen.
- Geht sie am besten im Gespräch mit den Kids Stück für Stück durch, lasst hauptsächlich sie reden, diskutieren, ausprobieren, und lenkt sie dann, wenn es stark in die falsche Richtung geht (inhaltlich oder sozial)

Was sie lernen

- Wie man komplexere Funktionen liest und versteht

Was sie lernen

- Wie man komplexere Funktionen liest und versteht
- Dass Funktionen eigentlich nahezu beliebige, wiederverwendbare Codeblöcke sind, die man bloß nicht so oft tippen möchte.
- Dass Funktionen optimalerweise eine Sinneinheit bilden
- Was Kommentare sind und wie sie funktionieren

Code

```
def CO2_berechnen(mult, weg):  
    punkte = mult*weg  
    return punkte
```

```
def zugfahrt(multiplikator):  
    strecke = input("Wie weit fährst du mit dem Zug? (in km)")  
    return CO2_berechnen(multiplikator, int(strecke))
```

```
punkte_pro_km = 1.4 # Punkte vom CO2 Budget  
points = zugfahrt(punkte_pro_km)  
print("Durch deine Fahrt mit dem Zug hast du " + str(points) + " Punkte verbraucht.")
```

Projektarbeit

- Von hier an ist alles offen. Die Kids dürfen gerne ihre eigenen Funktionen bauen, zusätzliche Verkehrsmittel hinzufügen, Pflanzen einbauen oder eine längere Fragerunde daraus machen.
- Wahrscheinlich seid ihr aber dann auch hier am Ende angekommen, vor allem, wenn ihr von den Kids genug ausgefragt wurdet.

(Größeres) Beispielprojekt

- Dieses Projekt ist nur ein Beispielprojekt, lasst die Kids gerne ihre eigenen Ideen ausprobieren.
- Dieses Projekt sollte man auch nicht als einen Erwartungshorizont verstehen, wenn die Kids nicht "so weit" kommen, ist das auch vollkommen okay.
- Andere Ideen, über die die Teilnehmenden noch nachdenken könnten, wären:
 - Wie kriege ich verschiedene Transportmittel in mein Programm?
 - Welche Freizeitaktivitäten haben welchen CO₂-Ausstoß?
 - Für Schnelle: CO₂-Budget von einem Urlaub, also Stück für Stück über Transportmittel, Länge der Reise, Aktivitäten, Essen, etc.
 - Wie sieht es mit Pflanzen aus? Gibt es auch einen positiven CO₂ Abdruck? ◦ Es gibt hier viel Code, der sich wiederholt. Geht das besser? Wenn ja, wie?

```
def CO2_berechnen(mult, weg):  
    punkte = 0  
    punkte = mult*weg  
    return punkte
```

```
def zugfahrt(multiplikator):  
    strecke = input("Wie weit fährst du mit dem Zug? (in km)")  
    return CO2_berechnen(multiplikator, int(strecke))
```

```
def autofahrt(multiplikator):  
    strecke = input("Wie weit fährst du mit dem Auto? (in km)")  
    return CO2_berechnen(multiplikator, int(strecke))
```

```
def fahrradfahrt(multiplikator):  
    strecke = input("Wie weit fährst du mit dem Fahrrad (in km)?")  
    return CO2_berechnen(multiplikator, int(strecke))
```

```
vkmittel = input("Mit was fährst du? Auto, Zug oder Rad? ")  
if vkmittel == "Zug":
```

```

vkmittel = input("Mit was fährst du? Auto, Zug oder Rad? ")
if vkmittel == "Zug":
    punkte_pro_km_zug = 1.4 # Punkte vom CO2 Budget
    points = zugfahrt(punkte_pro_km_zug)
    print("Durch deine Fahrt mit dem Zug hast du " + str(points) + " Punkte verbraucht.")

elif vkmittel == "Auto":
    punkte_pro_km_auto = 2 # Punkte vom CO2 Budget
    points = autofahrt(punkte_pro_km_auto)
    print("Durch deine Fahrt mit dem Auto hast du " + str(points) + " Punkte verbraucht.")

elif vkmittel == "Rad":
    punkte_pro_km_rad = 0 # Punkte vom CO2 Budget
    points = fahrradfahrt(punkte_pro_km_rad)
    print("Durch deine Fahrt mit dem Rad hast du " + str(points) + " Punkte verbraucht.")

else:
    print("Dieses Verkehrsmittel kenne ich leider nicht :)")

```

Interne Abschlussrunde (Kleingruppe, ~5 Minuten)

- Wenn ihr eurer Gruppe am Ende noch Feedback geben wollt, dann nehmt euch gerne jetzt Zeit dafür.
- Ihr solltet eine Person bestimmen, die etwas von eurer Arbeit vorstellt.
- Es gibt aber nach dieser Abschlussrunde noch eine im großen Plenum.

Abschlussrunde (Plenum, ~20 min)

- Die Kids stellen ihre Projekte vor (einer pro Gruppe), und erzählen ein bisschen, was sie so gemacht haben.
- Die Inspirer reden über sich, wie sie zur IT gekommen sind, was ihnen daran Spaß macht, wieso sie es weiterhin machen wollen, und was sie den Jugendlichen gerne mitgeben möchten.
- Data Club Lüneburg & Feedbackbögen
- Der*Die Lehrer*in bekommt kurz das Wort.
- Der Hauptinspirer redet nochmal kurz darüber, welche Möglichkeiten es gibt sich noch mehr mit Programmieren auseinander zu setzen. Wir verteilen den Feedbackbogen-Link des Data Clubs Lüneburg an die Teilnehmer*innen und den*die Lehrer*in. Es werden Hinweise auf die Angebote des Data Clubs gegeben.