

# Anleitung: *Let's code!*

*Let's Code!* ist eine Einführung in die Grundlagen der digitalen Poesie. Dabei wirst du nicht nur ein Gedicht schreiben, sondern auch die Programmiersprache Python kennenlernen. Dafür brauchst du:

- diese Anleitung
- Programmiercode: Let's Code! (als txt.-Dokument)
- Bedienungsanleitung für den Code
- FAQ: Let's code! (PDF)
- einen internetfähigen Laptop oder PC

Wir haben einen Code, also einen Text in der Programmiersprache Python, geschrieben. Zuerst haben wir dafür die Substantive, Verben und Adjektive aus dem Gedicht "Der Knabe im Moor" von Annette von Droste-Hülshoff sortiert und in Listen geschrieben. Der Code gibt nun den Befehl, einzelne Begriffe aus den Listen zu nehmen und sie wieder neu zusammenzusetzen. Auf diese Weise entsteht ein neues Gedicht. Das ist schon ganz cool, aber es geht bestimmt noch besser! Also geben wir den Code jetzt an dich ab. Du kannst ihn verändern wie du möchtest und so ein neues Gedicht programmieren. Klingt kompliziert? Keine Sorge, diese Anleitung hilft dir! Du kannst sie Schritt für Schritt befolgen, bis du dich sicher genug fühlst, um den Code alleine zu bearbeiten.

## Anleitung:

### **Vorbereitung**

1. Lade dir alle drei Dokumente herunter. Das sind: Diese Anleitung, die txt-Datei mit Code, die Bedienungsanleitung für den Code, das FAQ.
2. Rufe den Compiler auf. Du findest ihn hier: <https://www.programiz.com/python-programming/online-compiler/>
3. Lösche alles (!) aus dem linken Textfeld
4. Kopiere den Code aus dem Dokument Programmiercode: Lets code! und füge ihn in das linke Textfeld im Compiler ein.

### **Den Code ausführen und verstehen**

5. Drücke "run" und schau was passiert.
6. Versuche den vorgegebenen Code zu verstehen. Dabei hilft dir die Bedienungsanleitung für den Code.

## **Den Code verändern**

7. Verändere den Code, indem du neue Befehle überlegst und programmierst. Auch dabei hilft dir die Bedienungsanleitung.

## **Reflexion (optional)**

8. Wenn ihr in einer Gruppe seid, könnt ihr euch am Ende die Ergebnisse präsentieren und darüber sprechen. Schaut euch zum Beispiel an, was Neues durch die Veränderung des Codes entstanden ist. Außerdem könnt ihr euch folgende Fragen stellen:
  - A. Was ist das Gedicht? Der Code oder nur das Gedicht, das durch den Code entsteht?
  - B. Wer ist der\*die Autor\*in von den Gedichten?
  - C. Wie sinnhaft ist das entstandene Gedicht und wie kann es interpretiert werden?
  - D. Muss für eine Interpretation mitgedacht werden, dass das entstandene Gedicht programmiert wurde oder kann es auch für sich stehen?

## Bedienungsanleitung für den Code

```
# Knabe im Moor von Annette von Droste-Hülshoff dekonstruiert und  
# durcheinandergewürfelt mit pythoncode. Code angelehnt an A House of  
# Dust copyright (c)  
# 2014 Nick Montfort <nickm@nickm.com>  
# Original by Alison Knowles & James Tenney, 1967  
# Bearbeitet von Kollektiv Bindestrich, 2023
```

Alles, was hinter einem Hashtag (#) steht, wird Kommentar genannt. Kommentare verändern nichts daran, was der Code ausgibt. Sie werden von Programmierer\*innen zum Beispiel dazu verwendet, Stellen zu markieren, die noch überarbeitet werden müssen oder um den Code zu erklären.



Was du hier verändern kannst:

Du kannst Kommentare löschen oder Neue hinzufügen. Damit veränderst du nur den Code, aber nicht die Ausgabe, also das Ergebnis des ausgegebenen Textes. Kommentare kannst du nicht nur hier, sondern an unterschiedlichen Stellen im Code platzieren.

```
import time  
from random import choice
```

In diesen zwei Zeilen werden Befehle geladen, die später dafür genutzt werden, dass zufällig Wörter aus den untenstehenden Listen ausgewählt werden. Verändere hier nichts, außer du kennst dich sehr gut mit python-codes aus und weißt genau was du tust.

```
substantive = ['Moor', 'Heiderauche', 'Phantome', 'Dünste', 'Ranke', 'Strauche',  
, 'Tritte', 'Quellchen', 'Spalte', 'Moor', 'Röhricht', 'Hauche', 'Fibel', 'Kind',  
'Fläche', 'Wind', 'Hage', 'Gräberknecht', 'Meister', 'Torfe', 'Rind', 'Knäblein',  
'Ufer', 'Gestumpf', 'Föhre', 'Knabe', 'Ohr', 'Riesenhalm', 'Speer', 'Spinnerin',  
'Spinnlenor', 'Haspel', 'Geröhre', 'Lauf', 'Fuße', 'Sohle', 'Melodei', 'Geiger-  
mann', 'Knauf', 'Hochzeitheller', 'Moor', 'Seufzer', 'Höhle', 'Margreth', 'Seele',  
, 'Knabe', 'Reh', 'Schutzengel', 'Knöchelchen', 'Grab', 'Moorgeschwele', 'Boden',  
, 'Weide', 'Lampe', 'Knabe', 'Scheide', 'Moor', 'Blick', 'Geröhre', 'Heide']
```

Das ist eine Liste der Substantive aus dem Gedicht „Der Knabe im Moor“ von Annette von Droste Hülshoff. Um ausgegeben zu werden, müssen sie zwischen Apostrophen stehen und durch ein Komma voneinander getrennt sein.



Was du hier verändern kannst:

1. Begriffe Löschen
2. Neue Begriffe hinzufügen
3. Begriffe austauschen
4. Auslassungen hinzufügen, indem du den Bereich zwischen den Apostrophen freilässt. So: ' ',

```
verben = ['geht', 'wimmelt', 'dreht', 'häkelt', 'springt', 'zischt', 'singt',  
'geht', 'knistert', 'hält', 'rennt', 'jagt', 'saust', 'raschelt', 'verzecht',  
'bricht', 'duckt', 'starrt', 'nickt', 'rennt', 'rieselt', 'knittert', 'dreht',  
'holt', 'brodeln', 'pfeift', 'stiehlt', 'ruft', 'gründet', 'flimmert', 'steht',  
'atmet', 'wirft', 'ist', 'ist']
```

Das ist eine Liste der Verben aus dem Gedicht „Der Knabe im Moor“ von Droste. Du kannst hier das gleiche verändern, wie bei den Substantiven.

```
adjektive = ['schaurig', 'schaurig', 'fest', 'zitternd', 'hohl', 'gespenstig', 'am  
besten', 'irre', 'zage', 'unheimlich', 'gespannt', 'unselig', 'gebannt', 'gespen-  
tig', 'ungetreu', 'diebisch', 'klaffend', 'verdammte', 'arm', 'wund', 'bleichend',  
'heimatlich', 'scheu', 'fürchterlich', 'schaurig']
```

Das ist eine Liste der Adjektiven aus dem Gedicht. Auch hier Du kannst hier das gleiche verändern, wie bei den Substantiven.

```
for i in range(0,10):
```

Hier wird festgelegt, dass 10 Zeilen ausgegeben werden. Du kannst die Anzahl der Ausführung ändern. Wenn du anstatt einer 10 eine 1 schreibst, wird nur eine Zeile ausgeführt. Der Befehl würde so aussehen:(0,1)

```
print(choice(substantive) + ' ' + choice(verben) + ' ' +  
choice(adjektive))
```

Dies ist der Print-Befehl. Ohne ihn läuft gar nichts! Denn das, was hinter Print steht, ist das was der Code ausgibt. Den Printbefehl (print) solltest du nicht verändern, aber das was dahinter steht.



Was du hier verändern kannst:

1. Du kannst die Reihenfolge ändern, in der die Wörter ausgegeben werden. Dafür musst du die Abfolge der Befehle ändern. Beispiel:  
`print(choice(verben) + ' ' + choice(adjektive) + ' ' + (choice(substantive)))`
2. Du kannst Befehle löschen und hinzufügen und z.B. zwei Adjektive ausgeben lassen. Beispiel:  
`print(choice(adjektive) + ' ' + choice(adjektive))`
3. Du kannst Wörter oder Sätze hinzufügen, die immer ausgegeben werden. Beispiel  
`print('Droste sagt: ' + choice(substantive) + ' ' + choice(verben) + ' ' + choice(adjektive))`

```
time.sleep(1.75)
```

In der letzten Zeile wird festgelegt, dass nach 1.75 Sekunden eine neue Zeile ausgegeben wird.



Was du hier verändern kannst:

1. Du kannst die Zeitangabe in den Klammern (1.75) verändern und die Ausgabeabstände so verlängern oder verkürzen.
2. Du kannst die Befehle import time und time.sleep(1.75) löschen. Dann werden alle Zeilen gleichzeitig ausgegeben.

## Let's code! FAQ:

### **1. Ich habe noch nie etwas programmiert. Kann ich das Format trotzdem machen?**

Es ist überhaupt kein Problem, wenn du noch keine Erfahrungen damit hast, zu programmieren. Alles was du wissen musst, findest du in der Bedienungsanleitung für den Code. Nimm dir Zeit den Code genau anzuschauen und zu verstehen. Lies die Bedienungsanleitung durch und verändere erstmal Beispiele, die dort gegeben werden. Es kann auch helfen, sodass du auf deinen alten Arbeitsstand zurückgreifen kannst, wenn etwas nicht funktioniert.

### **2. Was ist ein Code?**

Ein Code ist ein Text, der in einer Programmiersprache geschrieben wurde. Er gibt einem Computerprogramm Anweisungen, um etwas Bestimmtes auszuführen. Unseren Code, findest du im Dokument: *Programmiercode: Let's code!*

### **3. Was ist ein Compiler?**

Ein Compiler ist ein Programm, welches Programme erzeugt. Er sorgt dafür, dass der geschriebene (für uns verständliche) Code in ein Programm (also eine für den Computer lesbare Form) umgewandelt wird, damit dieser den Code ausführen kann. In dem Online-Compiler wird der Code auf der linken Seite eingefügt und auf der rechten Seite schreibt das kompilierte Programm die entstandenen Ausgaben.

### **4. Ich verstehe den Code nicht. Was kann ich tun?**

Keine Panik! Wenn du noch keine Erfahrungen hast, sieht der Code ziemlich kompliziert aus. Nimm dir Zeit, die Bedienungsanleitung genau durchzulesen und mit dem Code im Compiler zu vergleichen. Versuche zu verstehen, an welcher Stelle was passiert. Wenn du gar nicht weiter kommst: Frage jemanden in deiner Umgebung und denkt gemeinsam darüber nach.

### **5. Wie kann ich den Code verändern?**

Anregungen dafür, was du im Code verändern kannst, findest du in der Bedienungsanleitung. Schau sie dir an und trau dich zu experimentieren.

### **6. Ich habe schon Erfahrungen mit Coding. Kann ich das Format trotzdem machen?**

Na klar! Schau dir den Code an und experimentiere damit. Du musst dich nicht an die Anleitungen halten. Wenn du schon Erfahrungen hast, fällt dir selbst bestimmt viel ein, wie du den Code verändern und ein digitales Gedichte schreiben kannst!

### **7. Wie kann ich den Code und das ausgegebene Gedicht speichern?**

Im Compiler selbst kannst du leider nicht speichern. Aber du kannst den Code im linken Textfeld kopieren und in ein separates Dokument (z.B. in die txt.-Datei oder bei Word) einfügen. So kannst du immer wieder darauf zugreifen. Das Gedicht kannst du auf die gleiche Weise speichern. Kopiere dafür das Gedicht aus dem rechten Textfeld in ein Dokument.

### **8. Ich kann nicht in den Code klicken, um ihn zu bearbeiten oder zu kopieren. Woran liegt das?**

Wahrscheinlich hast du auf "run" geklickt und die Ausgabe ist noch nicht fertig. Um den Code zu bearbeiten, musst du warten, bis ganz unten an deinem Gedicht dieses Zeichen steht: >

### **9. Ich habe etwas falsch eingegeben. Was kann ich tun?**

Leider kannst du im compiler nicht auf “zurück” klicken, um deine Eingabe rückgängig zu machen. Du musst die entsprechende Stelle also noch einmal richtig eingeben. Wenn du nicht weißt, was du falsch gemacht hast, schau mal bei Punkt 8. Dort findest du eine Liste der häufigsten Fehler im Code.

**10. Ich habe auf “run” geklickt, jetzt kommt eine Fehlermeldung. Woran liegt es?**

Das passiert häufig, meistens ist das Problem aber schnell gefunden. Wahrscheinlich ist es einer dieser Fehler:

- Die letzten beiden Zeilen sind nicht eingerückt.  
Vor `print(choice(substantive) + ' ' + choice(verben) + ' ' + choice(adjektive))`  
und `time.sleep(1.75)` müssen jeweils vier Leerzeichen stehen.
- Ein Satzzeichen ist falsch. Es kann zum Beispiel eine Klammer, ein Komma oder ein Leerzeichen zu viel oder zu wenig sein und schon läuft der Code nicht mehr.
- Du hast den falschen Apostroph (') verwendet. Dieser ist der Richtige: ' . Du findest ihn auf der Tastatur am rechten Rand. Er ist auf der gleichen Taste wie der Hashtag (#). Um ihn einzugeben, musst du auf einer Windows-Tastatur die strg-Taste gedrückt halten, bei einer Apple-Tastatur ist das die shift-Taste.

Der Fehler muss an der letzten Stelle sein, die du bearbeitet hast. Lies sie dir aufmerksam durch, um den Fehler zu finden. Vergleiche deinen Code dabei am besten mit dem Original oder einer ähnlichen Stelle in deiner Bearbeitung. Du weißt immer noch nicht weiter? Hol dir Hilfe! Wenn ihr zu zweit über den Code schaut, findet ihr den Fehler bestimmt sofort.

Ganz zur Not, kannst du jederzeit die Seite aktualisieren und den vorgegebenen Code nochmal ganz von vorne einfügen und verändern.