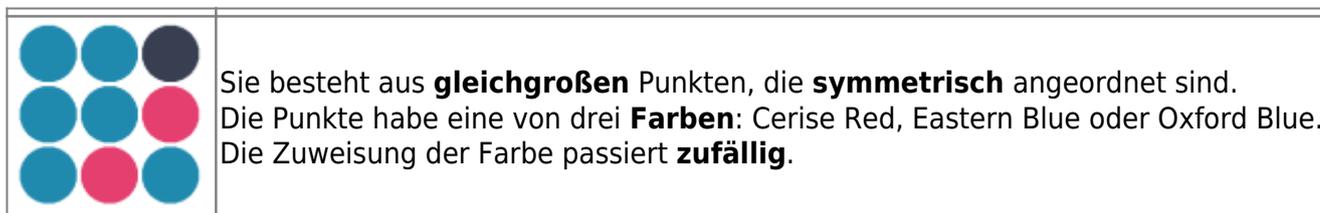


Alternative

In dieser Lektion beschäftigen wir uns mit Zufallszahlen und Alternativen. Dabei programmieren wir auf den Pfaden von [Vera Molnar](#), einer Pionierin der Computerkunst.

Das Ziel dieser Lektion ist die Programmierung dieser Grafik *à la Molnar*:



Probiere es aus; klicke auf **Run!**

```
<html>
```

```
<iframe
```

```
src=„https://trinket.io/embed/python/94ee1b191b?outputOnly=true&runOption=run&start=result“  
width=„40%“ height=„150“ frameborder=„0“ marginwidth=„0“ marginheight=„0“  
allowfullscreen></iframe>
```

```
</html>
```

Arbeitsauftrag 1

- Schreibe ein Programm, das drei Punkte nebeneinander zeichnet, also der ersten Zeile der oben angeführten Grafik entspricht.
- Die drei Punkte sollen die Farbe Eastern Blue haben -> #208aae
- Mit `turtle.dot(size, color)` zeichnet die Turtle Punkte. Beachte: Farbbezeichnung in Anführungszeichen setzen!
- Teste dein Programm im Editor.
- Notiere deine Lösung handschriftlich auf Papier oder digital!

```
<html>
```

```
<iframe src=„https://trinket.io/embed/python/4060457aff“ width=„100%“ height=„300“  
frameborder=„0“ marginwidth=„0“ marginheight=„0“ allowfullscreen></iframe>
```

```
</html>
```

Zufallszahlen

Eine natürliche Zufallszahl gibt uns folgende Anweisung zurück:

```
random.randint(m, n)
```

Der Aufruf `random.randint(1, 10)` liefert also eine natürliche Zahl zwischen 1 und 10, der Aufruf `random.randint(5, 8)` eine natürliche Zahl zwischen 5 und 8.

Wir können diese Zufallszahl einer Variable zuweisen und mit `print()` in der Konsole ausgeben lassen.

Kopiere den Quellcode und füge unten in Editor ein. Dann klicke immer wieder auf Run! Beobachte die Ausgabe (Result) auf der rechten Seite!

```
import random
zahl = random.randint(1, 100)
print(zahl)
```

```
<html>
```

```
<iframe src=„https://trinket.io/embed/python/bc384dac6a“ width=„100%“ height=„200“
frameborder=„0“ marginwidth=„0“ marginheight=„0“ allowfullscreen></iframe>
```

```
</html>
```

Alternative / Zweiseitige Auswahlstruktur - Allgemeine Syntax in Python

Die Alternative besteht aus einer Bedingung, deren Wahrheitswert überprüft wird. Je nach Ergebnis dieser Prüfung wird einer von zwei Anweisungsblöcken ausgeführt.

```
if bedingung:
    anweisung(en)
else:
    anweisung(en)
```

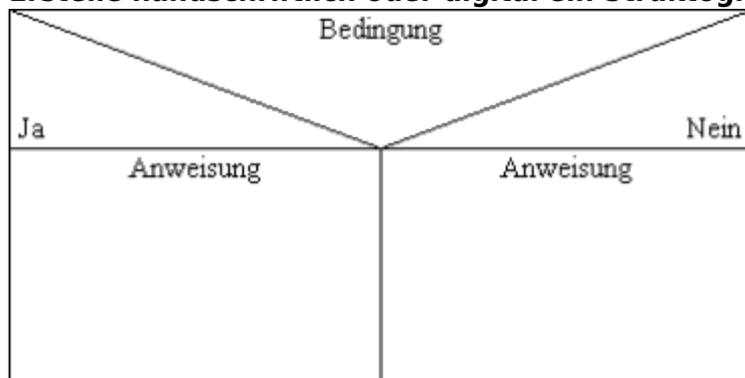
Beispiel

```
if x > y:
    turtle.forward(5)
else:
    turtle.forward(20)
```

Arbeitsauftrag 2

Wir möchten diese Zufallszahl-Funktion nun nutzen. Wenn die Zufallszahl $\{1,100\}$ größer als 95 ist, dann soll die Farbe Oxford Blue ($\#373f51$) für den Punkt gewählt werden. Ansonsten die Farbe Eastern Blue ($\#208aae$).

Erstelle handschriftlich oder digital ein Struktogramm zu dieser Problemstellung.



Arbeitsauftrag 3

Übersetze nun dein Struktogramm vom vorherigen Arbeitsauftrag in Python-Syntax.

- Orientiere dich an der allgemeinen Syntax und dem Beispiel für eine Verzweigung in Python.
- **Notiere deine Lösung handschriftlich oder digital!**

Zur Erinnerung:

Wenn die Zufallszahl $\{1,100\}$ größer als 95 ist, dann soll ein Punkt mit der Farbe Oxford Blue ($\#373f51$) gezeichnet werden. Ansonsten soll ein Punkt mit der Farbe Eastern Blue ($\#208aae$) gezeichnet werden. `<html>`

```
<iframe src=„https://trinket.io/embed/python/f1be80d8f8“ width=„100%“ height=„300“
frameborder=„0“ marginwidth=„0“ marginheight=„0“ allowfullscreen></iframe>
```

```
</html>
```

Arbeitsauftrag 4

Erweitere deinen Quellcode so, dass drei Punkte nacheinander gezeichnet werden. Die Farbe der Punkte soll weiterhin zufällig bestimmt werden.

- Teste dein Quellcode oben im Editor!
- **Notiere deine Lösung handschriftlich oder digital!**

[[informatik](#), [computerkunst](#), [lernpfad](#), [python](#)]

Last update: 2020/05/25 16:20 informatik:computerkunst:alternative <https://herr-pfeiffer.de/unterrichtswiki/informatik:computerkunst:alternative?rev=1590416452>

From:
<https://herr-pfeiffer.de/unterrichtswiki/> - **Unterrichtswiki - Herr Pfeiffer**

Permanent link:
<https://herr-pfeiffer.de/unterrichtswiki/informatik:computerkunst:alternative?rev=1590416452>

Last update: **2020/05/25 16:20**

