



## Einstieg in die Minecraft Pi Edition

Minecraft ist ein bekanntes sandbox Aufbau-Spiel. Auf dem Raspberry Pi befindet sich eine kostenlose Version von Minecraft, die zusätzlich noch mit einer Programmierschnittstelle ausgestattet ist. Das bedeutet, du kannst Befehle und Skripte in Python schreiben und damit z.B. im Spiel automatisch Sachen bauen. Auf diese Art und Weise kann man super Python lernen!



### 1 - Minecraft starten und steuern

Um die Minecraft Pi Edition zu starten, kannst du sie entweder über das Desktop-Menü oder über die Konsole aufrufen. Der Konsolenbefehl lautet

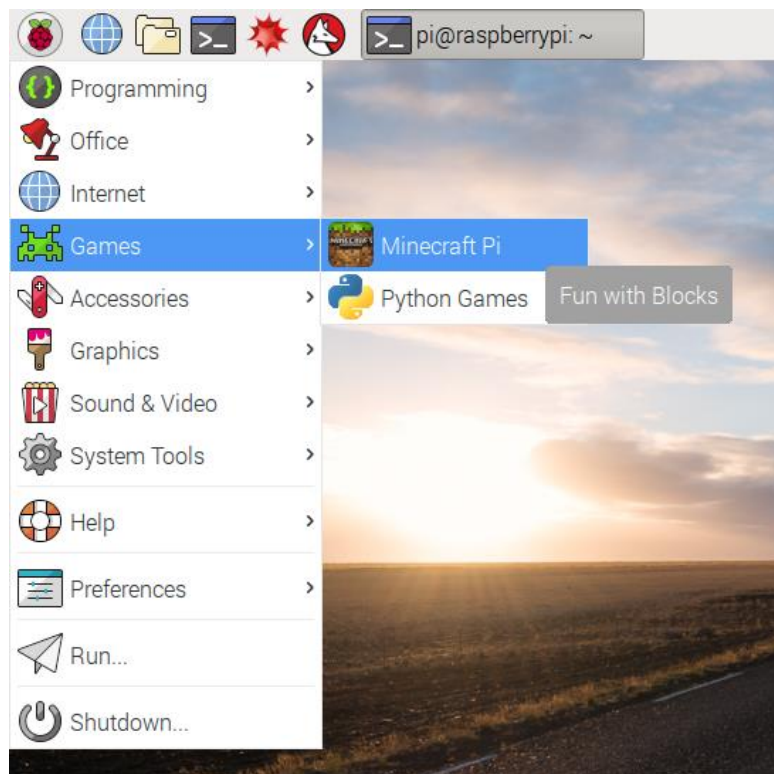
```
minecraft-pi
```

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

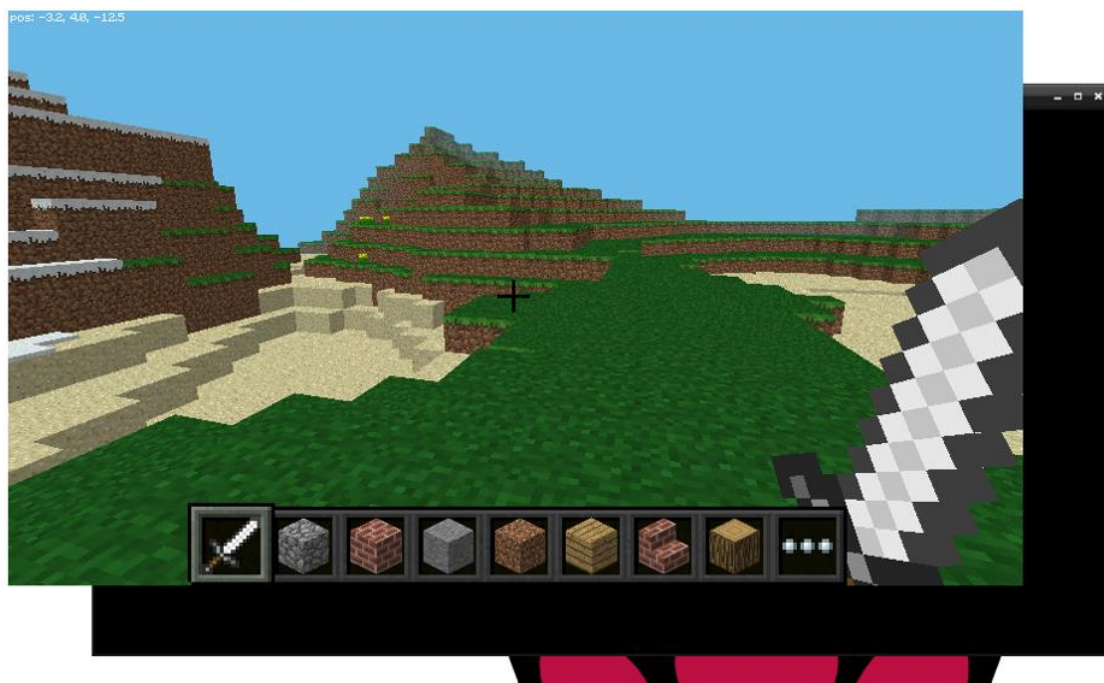
Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



Wenn Minecraft geladen ist, kannst du ein neues Spiel starten oder eine Welt laden.



Jetzt lauf herum, schlag auf Blöcke und bau etwas!

Mit der Maus kannst du dich umsehen, mit den folgenden Tasten bewegst du dich:

Taste	Aktion
-------	--------

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017

W	Vorwärts
A	Links
S	Rückwärts
D	Rechts
E	Inventar öffnen
Leertaste	Springen
2x Leertaste	Fliegen AN / AUS
Esc	Pause / Menü
Tab	Maus freigeben

Items kannst du mit dem Mausrad (oder den Nummerntasten) aus der Schnellzugriff-Leiste unten auswählen. Für eine größere Auswahl kannst du mit E im Inventar nachsehen.



Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



Während du das Schwert in der Hand hast, kannst du mit einem Klick Blöcke vor dir entfernen (z.B. um ein Loch zu graben). Hast du einen Block in der Hand, kannst du ihn mit der rechten Maustaste vor dir platzieren.

## 2 - Erste Schritte mit der Python-API

### Text ausgeben

Während Minecraft läuft und eine Welt geöffnet ist, kannst du mit der Python-Schnittstelle auf das Spiel zugreifen. Drücke dazu zuerst die Tab-Taste (TAB) um den Mauszeiger freizugeben und öffne Python3 aus dem Startmenü. Positioniere beide Fenster so, dass du sie nebeneinander sehen kannst, ohne dass sie sich überlappen.

Du kannst die folgenden Befehle entweder einzeln in das Python-Fenster eingeben, oder eine eigene Skript-Datei erstellen, sodass du deinen Code speichern und wiederverwenden kannst.

Um eine Datei zu erstellen, wähle Datei > Neues Fenster und dann Datei > Speichern. Speichere die Datei am Besten in deinem Home-Folder oder erstelle einen extra Ordner für deine Minecraft-Skripts.

Als ersten Schritt werden wir die Minecraft-Bibliothek einbinden, die uns erlaubt eine Verbindung zum Spiel herzustellen.

```
from mcpi.minecraft import Minecraft  
mc = Minecraft.create()
```

Testen werden wir diese Verbindung, indem wir "Hallo Spieler!" im Minecraft-Chat ausgeben.

```
mc.postToChat("Hallo Spieler!")
```

Wenn du die Befehle einzeln ins Python-Fenster eingibst, reicht es wenn du nach jeder Zeile die Eingabetaste drückst.

Schreibst du in eine Python-Datei, tippe alle Zeilen ein und speichere dann mit STRG+S und starte das Skript mit der F5 Taste.

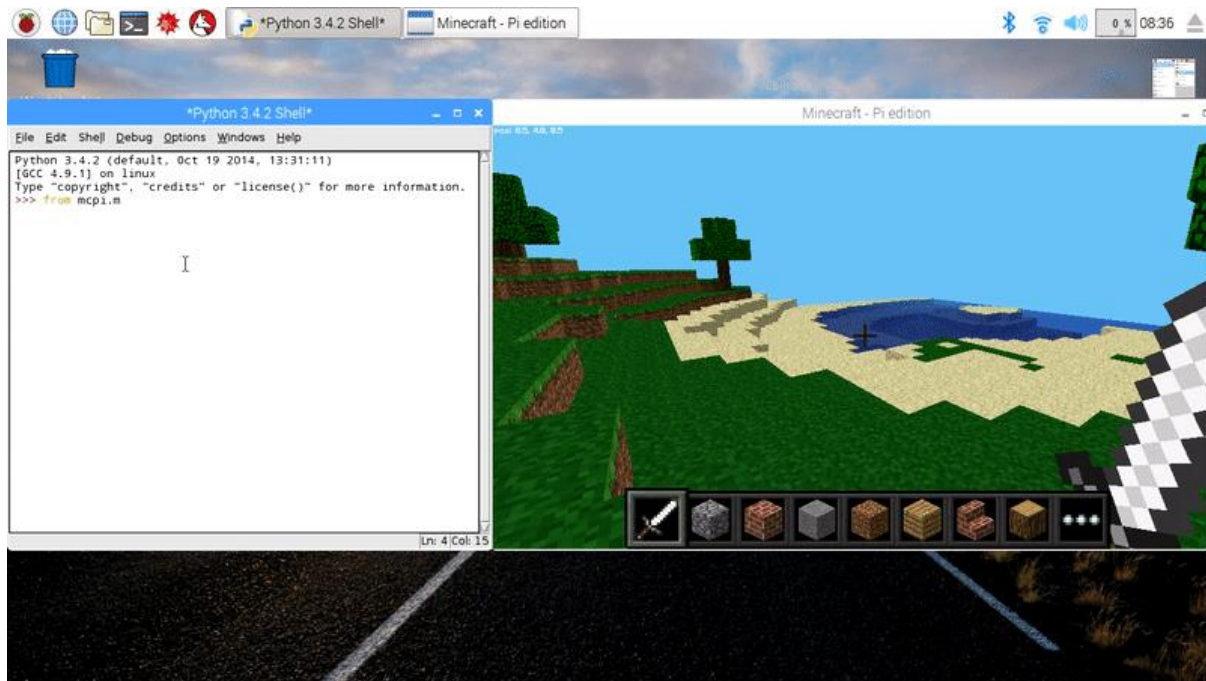
Dein Code sollte jetzt laufen und die Nachricht im Minecraft-Fenster erscheinen.

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



## Deine Position bestimmen

Um deine Position in der Minecraft-Welt zu bestimmen, tippe:

```
pos = mc.player.getPos()
```

Die Variable **pos** enthält jetzt deine Position, die sich aus drei Koordinaten zusammensetzt: pos.x, pos.y und pos.z.

Alternativ kannst du diese Koordinaten auch ganz komfortabel direkt mit Python "entpacken":

```
x, y, z = mc.player.getPos()
```

Jetzt enthalten die Variablen x, y und z die jeweiligen Teile deiner aktuellen Position. X und z sind die Richtungen in denen du dich bewegst (vor/zurück und links/rechts) und y ist oben/unten.

Denk daran, dass getPos() die Position zu genau diesem Zeitpunkt liefert, wo du es ausführst! Wenn du dich bewegst, musst du den Befehl erneut ausführen, um korrekte Werte zu haben.

## Teleportieren

Genauso, wie du deine Position auslesen kannst, kannst du sie auch selber bestimmen.

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



```
x, y, z = mc.player.getPos()  
mc.player.setPos(x, y+100, z)
```

Mit diesen Befehlen teleportierst du deine Spielfigur 100 Blöcke hoch in die Luft. Sehr wahrscheinlich wirst du also hoch oben in der Luft auskommen und dann recht schnell wieder Richtung Boden fallen.

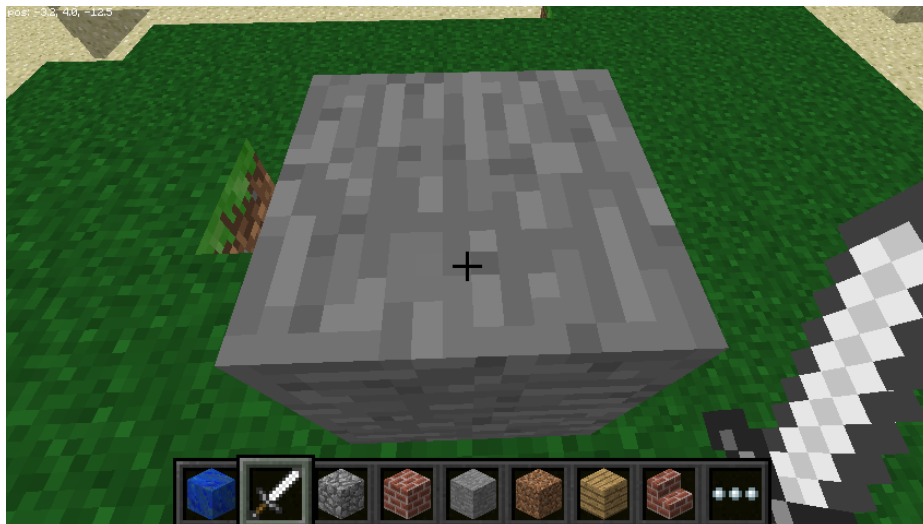
Versuch es selbst! Teleportiere dich woanders hin!

## Blöcke setzen

In Minecraft geht es ums Bauen. Kein Wunder also, dass du einzelne Blöcke an eine bestimmte Position setzen kannst. Der Befehl dafür lautet `mc.setBlock(x, y, z, id)`:

```
x, y, z = mc.player.getPos()  
mc.setBlock(x+1, y, z, 1)
```

Führst du diese beiden Zeilen aus, sollte jetzt ein Stein-Block vor deiner aktuellen Position erscheinen. Taucht er nicht direkt vor dir auf, ist er wahrscheinlich direkt neben oder hinter dir. Ist das der Fall, musst du mit der Maus in das Minecraft-Fenster zurückkehren und dich im Kreis drehen, bis du ihn vor dir siehst.



Die Werte, die wir dem Befehl übergeben sind `x,y,z` und `id`. (`x, y, z`) beschreiben dabei die Position in der Minecraft-Welt (wir haben diese auf einen Block von uns entfernt festgelegt indem wir `x+1` übergeben haben) und `id` beschreibt den Typ des Blocks, den wir gerne setzen wollen. 1 steht für Stein.

Andere Block-Typen, die du ausprobieren kannst, sind:

Luft: 0

Gras: 2

Erde: 3

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



Wenn der gesetzte Block in deinem Sichtfeld ist, ändere ihn in einen anderen Typ:

```
mc.setBlock(x+1, y, z, 2)
```

Du solltest jetzt sehen, dass sich der Stein-Block vor deinen Augen in einen Erde-Block verwandelt hat!



## Blöcke als Variablen

Wenn du die Typ-ID eines Blockes weißt, kann es manchmal Sinn machen, diese als Variable zu speichern und dann immer wieder zu benutzen:

```
erde = 3  
mc.setBlock(x, y, z, erde)
```

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



Eine Übersicht über alle in Minecraft verfügbaren Blöcke findet sich unter:

[https://minecraft.gamepedia.com/Java Edition data values/Block IDs](https://minecraft.gamepedia.com/Java_Edition_data_values/Block_IDs)

## Spezial-Blöcke

Es gibt auch Blöcke mit einer extra-Eigenschaft, so wie z.B. Wolle, bei der du die Farbe bestimmen kannst. Um diese Eigenschaft zu setzen, benutzen wir bei solchen Blöcken eine Extra-Wert wenn wir sie mit `setBlock` setzen:

```
wolle = 35
w_orange = 1
mc.setBlock(x, y, z, wolle, w_orange)
```

Hier übergeben wir eine extra Variable mit dem Wert 1 um die Wolle orange zu färben. Geben wir keinen vierten Wert an, so denkt Python wir wollen 0 also weiße Wolle.

Ein paar weitere Farben sind:

- 2: Magenta
- 3: Hell Blau
- 4: Gelb

Probier selber ein paar Zahlen und schau was sie bewirken!

## Mehrere Blöcke setzen

Anstatt nur einen beliebigen Block mit `setBlock()` zu setzen, kannst du auch eine Menge Blöcke mit `setBlocks()` setzen, die einen viereckigen Raum zwischen zwei Koordinaten ausfüllen:

```
stein = 1
x, y, z = mc.player.getPos()
mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, stein)
```

Mit diesen Befehlen füllen wir einen 10x10x10 großen Würfel mit Stein-Blöcken.

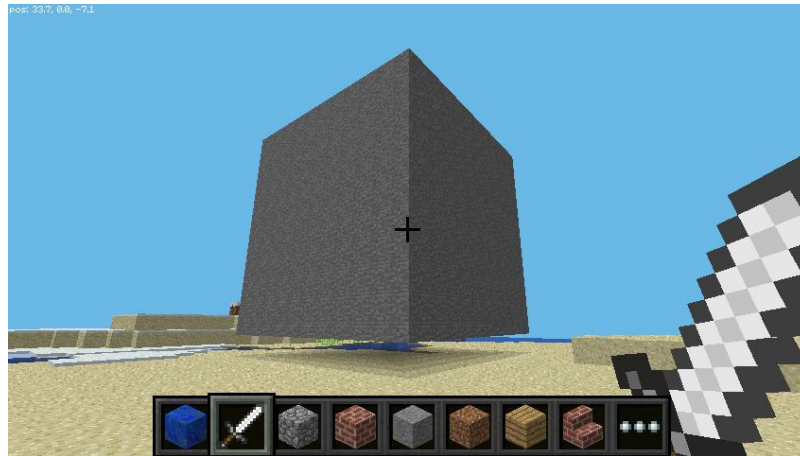
Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017





Denk dran: Je mehr Blöcke du mit `setBlocks()` setzt, desto länger kann es dauern bis sie vollständig auftauchen!

### 3 - Blöcke beim Laufen hinterlassen

Jetzt weißt du, wie man Blöcke setzt! Lass uns nun versuchen, Blöcke automatisch da zu hinterlassen, wo wir entlang laufen.

Das folgende Skript setzt Blumen hinter dir ab, während du dich bewegst:

```
from mcpi.minecraft import Minecraft
from time import sleep

mc = Minecraft.create()

blume = 38

while True:
    x, y, z = mc.player.getPos()
    mc.setBlock(x, y, z, blume)
    sleep(0.1)
```

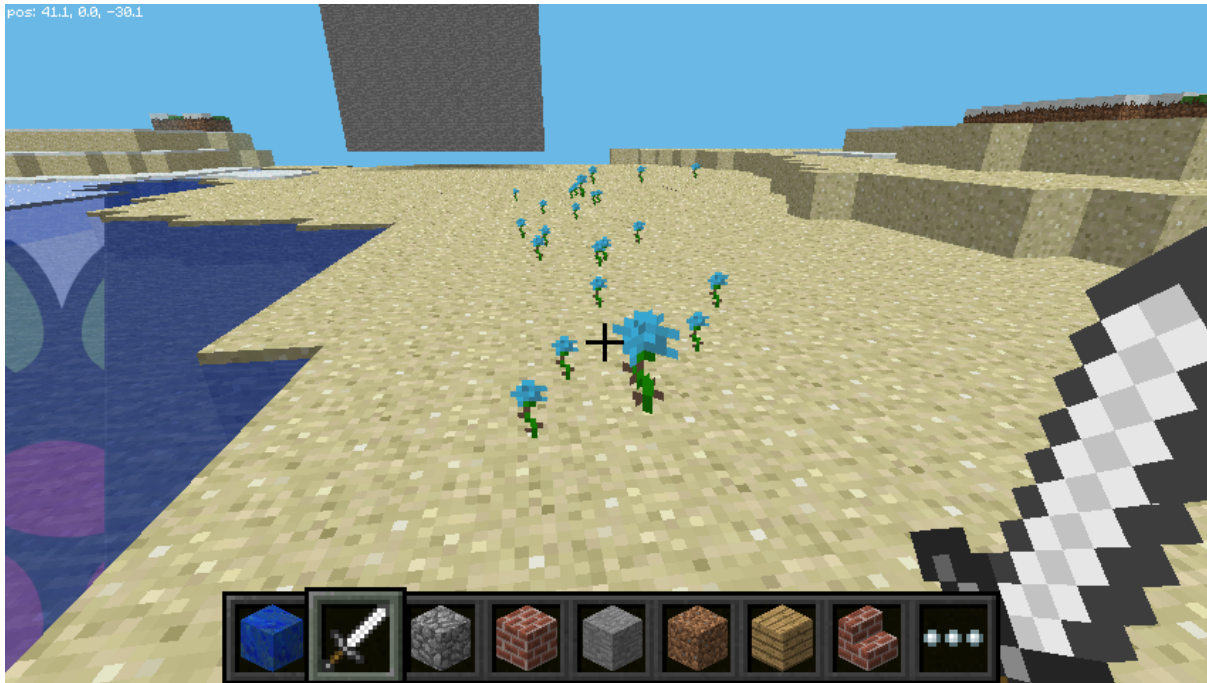
Führe es aus und laufe ein Stück vorwärts. Wenn du dich umdrehst, siehst du jetzt die Spur aus Blumen, die du hinterlässt.

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



Da wir **while True** benutzen, wird die Schleife ewig laufen. Hast du genug, drücke also **STRG + C** im Python-Fenster um das Skript manuell zu unterbrechen.

Da das Skript sehr schnell läuft, warten wir mit **sleep(0.1)** einen kurzen Moment, bevor wir die nächste Blume setzen.

Was wäre, wenn wir nur Blumen setzen wollten, wenn unsere Spielfigur auf Rasen läuft? Wir können `getBlock()` benutzen, um herauszufinden, auf welchem Typ Block wir uns befinden:

```
x, y, z = mc.player.getPos()
dieser_block = mc.getBlock(x, y, z)
print(dieser_block)
```

Dieses Skript ist noch fehlerhaft: Es liefert uns den Typ des Blocks IN DEM wir aktuell stehen (meistens 0 - Luft). Wir wollen aber wissen welchen Typ der Block UNTER UNS hat. Dafür ziehen wir 1 von unserer y-Position ab und erfragen den Typ des Blocks mit `getBlock()`:

```
x, y, z = mc.player.getPos()
block_unter_mir = mc.getBlock(x, y-1, z)
print(block_unter_mir)
```

Jetzt haben wir die ID des Blocks auf dem du stehst.

Du kannst diesen Code testen, indem du ihn in der while-Schleife von oben ausführst und dich bewegst:

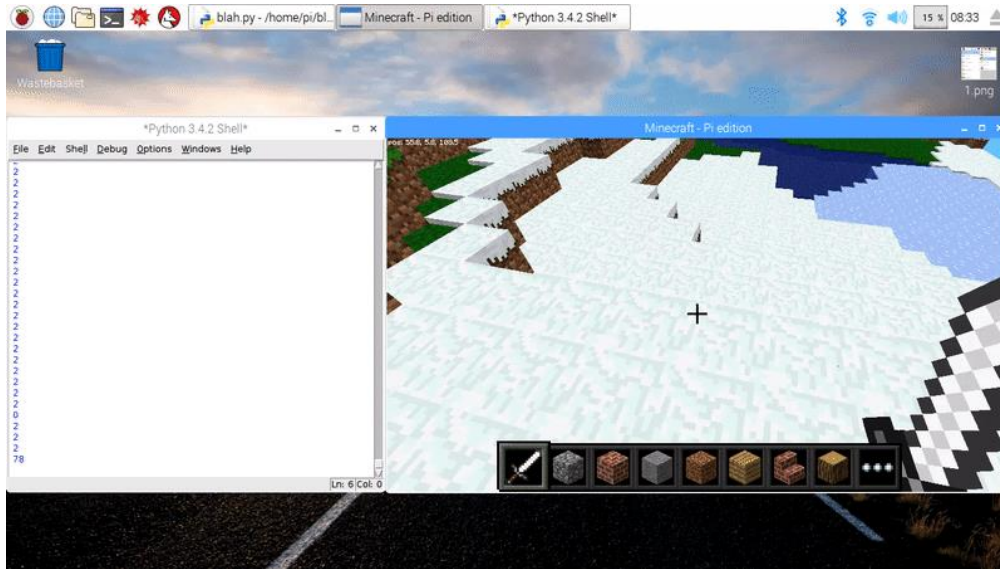
Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017

```
while True:
    x, y, z = mc.player.getPos()
    block_unter_mir = mc.getBlock(x, y-1, z)
    print(block_unter_mir)
```



Jetzt können wir mit einer if-Bedingung unterscheiden, ob wir eine Blume setzen wollen, oder nicht:

```
gras = 2
blume = 38

while True:
    x, y, z = mc.player.getPos()
    block_unter_mir = mc.getBlock(x, y-1, z)

    if block_unter_mir == gras:
        mc.setBlock(x, y, z, blume)
        sleep(0.1)
```

Zusätzlich können wir den Block unter uns in einen Gras-Block verwandeln wenn er noch keiner ist:

```
if block_unter_mir == gras:
    mc.setBlock(x, y, z, blume)
else:
    mc.setBlock(x, y-1, z, gras)
```

Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



Wenn du jetzt läufst und unter dir Gras ist, hinterlässt du eine Blume. Läufst du nicht auf Gras, so verwandelst du den Boden unter deinen Füßen in Gras, kannst dich umdrehen und durch erneutes überqueren jetzt eine Blume hinterlassen.



Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



## 4 - Das Spiel mit dem TNT

Ein weiterer interessanter Block ist TNT! Um einen normalen TNT-Block zu setzen, führe folgende Befehle aus:

```
tnt = 46  
mc.setBlock(x, y, z, tnt)
```



Diese Art von TNT ist jedoch relativ langweilig und ungefährlich. Setzen wir den Block jedoch mit der Extra-Eigenschaft 1, wird es interessanter:

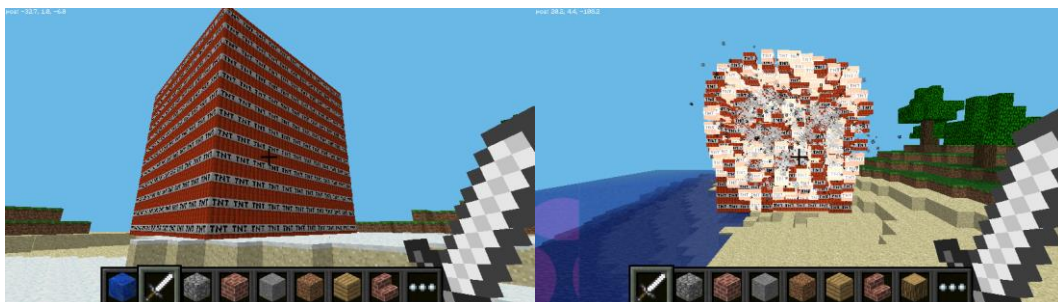
```
tnt = 46  
mc.setBlock(x, y, z, tnt, 1)
```

Wenn du jetzt mit deinem Schwert auf den Block schlägst, wird er aktiviert und explodiert nach wenigen Sekunden! Probier es aus!

Als nächstes erzeugen wir einen großen TNT-Würfel:

```
tnt = 46  
mc.setBlocks(x+1, y+1, z+1, x+5, y+5, z+5, tnt, 1)
```

Aktiviere einen der TNT-Blöcke mit deinem Schwert und bring dich schnell in Sicherheit. Aus sicherer Entfernung kannst du die Show genießen, wenn alles in die Luft fliegt! Minecraft kann sehr langsam werden, da es viel Rechenleistung fordert die Explosion darzustellen, hab also Geduld.



Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017



## 5 - Heißer Spaß mit Lava

Ein weiterer spaßiger Block ist der Lava-Block.

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
x, y, z = mc.player.getPos()
Lava = 10
mc.setBlock(x+3, y+3, z, Lava)
```

Finde den Lava-Block, den du gerade platziert hast und sieh dir an, wie die Lava vom Block auf den Boden fließt.

Das coole an Lava ist, dass sie zu Stein wird wenn sie abkühlt.

Probiere folgendes Skript an einer anderen Stelle in deiner Welt aus:

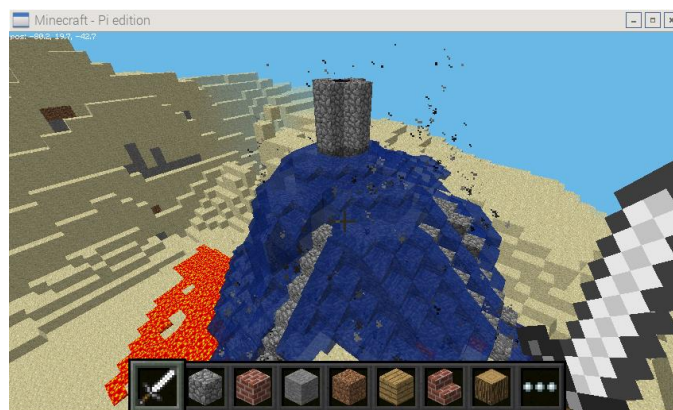
```
from mcpi.minecraft import Minecraft
from time import sleep
mc = Minecraft.create()

x, y, z = mc.player.getPos()

Lava = 10
wasser = 8
Luft = 0

mc.setBlock(x+3, y+3, z, Lava)
sleep(20)
mc.setBlock(x+3, y+5, z, wasser)
sleep(4)
mc.setBlock(x+3, y+5, z, Luft)
```

Indem du die Werte im sleep()-Befehl änderst, kannst du kontrollieren ob mehr oder weniger Lava fließt.



Quelle: <https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi>

Übersetzung: Martin Busch

Version: 1.2

Datum: 25.09.2017