

CoderDojo Minetest: Dein Spickzettel

Mit diesem Spickzettel lernst du, wie du deine eigene Minetest-Welt mit Code verändern kannst!

Als erstes kurz einmal die wichtigsten Tasten, die du in Luanti brauchen wirst:

- Bewegen: W, A, S, D (vorwärts, links, rückwärts, rechts)
- Flugmodus An/Aus: K
- Inventar öffnen: I
- Pause-Menü: ESC
- Befehle: Mit `/`, wichtige sind zum Beispiel `/time 6000` um die Zeit auf Tag zu setzen
`/sethome` um das Zuhause zu setzen, `/home` um sich zum Zuhause zu teleportieren

Die Grundlagen von Lua

- **Befehle:** Jeder Befehl, den du Minetest gibst, beginnt mit `mod.`. Zum Beispiel `mod.chat("Hallo")`.
- **Variablen:** Stell dir eine Variable wie eine Kiste vor, in der du etwas aufbewahren kannst, zum Beispiel einen Text oder eine Zahl.

```
-- In die Kiste 'gruss' legen wir den Text "Hallo Welt"
local gruss = "Hallo Welt!"
-- Jetzt können wir die Kiste benutzen, um den Gruss in den Chat zu schicken
mod.chat(gruss)
```

- **Positionen:** Um Minetest zu sagen, wo etwas passieren soll, brauchen wir Koordinaten (x, y, z).
 - **x:** nach links oder rechts (Ost/West)
 - **y:** nach oben oder unten
 - **z:** nach vorne oder hinten (Nord/Süd)
Wir schreiben das so: `mod.position(10, 5, 20)`
- **Entscheidungen mit `if`:** Manchmal soll dein Code nur etwas tun, wenn eine bestimmte Bedingung wahr ist. Wie eine Entscheidung an einer Weggabelung.

```
-- Prüfe, ob der Spieler sehr hoch in der Luft ist
if mod.spieler_pos().y > 50 then
    mod.chat("Achtung, du fällst tief!")
else
    mod.chat("Alles sicher hier unten.")
end
```

- **Wiederholungen mit `for`:** Eine `for`-Schleife ist super, um eine Aktion mehrmals zu wiederholen, ohne den Code immer wieder neu schreiben zu müssen.
 - **Von-Bis zählen:** Sage dem Computer, wie oft er etwas tun soll.

```
-- Baut eine kleine Treppe aus 5 Stein-Stufen nach oben
for i = 1, 5 do
    mod.set_block("default:stone", mod.position(i, i, 0))
end
```

- **Eine Liste durchgehen:** Manchmal hast du eine Liste von Dingen und möchtest für jedes Ding in der Liste dasselbe tun.

```
-- Eine Liste mit verschiedenen Blöcken
local bloecke = {"default:dirt", "default:stone", "default:sand", "default:glass"}
-- Gehe die Liste durch und schreibe jeden Block-Namen in den Chat
for block in each(bloecke) do
    mod.chat("Ich habe in meiner Liste gefunden: " .. block)
end
```

- **Zufall:** Manchmal soll der Computer etwas Unvorhersehbares tun. Dafür gibt es `zufall`!

- **Zufallszahl:** Bekomme eine zufällige Zahl zwischen zwei Werten.

```
-- Teleportiere dich an eine zufällige Höhe zwischen 10 und 50
local zufalls_hoehe = zufall(10, 50)
mod.teleportiere_spieler(mod.position(0, zufalls_hoehe, 0))
```

- **Zufall aus einer Liste:** Gib dem Computer eine Liste und er sucht sich ein zufälliges Teil daraus aus.

```
-- Eine Liste mit bunten Blöcken
local bunte_bloecke = {"default:goldblock", "wool:red", "wool:blue", "wool:green"}
-- Wähle einen zufälligen Block aus der Liste aus
local zufalls_block = zufall(bunte_bloecke)
-- Platziere den zufälligen Block direkt vor dem Spieler
mod.setze_block(zufalls_block, mod.spieler_pos())
```

- **Warten:** Um etwas erst nach einer gewissen Zeit auszuführen, kannst du `warte` verwenden.

```
mod.chat("Das passiert sofort!")
warte(3, function()
    mod.chat("Das passiert nach 3 Sekunden!")
end)
```

- **Kommentare:** Wenn du eine Zeile mit `--` beginnst, ignoriert der Computer sie. Das ist super, um dir selbst Notizen in den Code zu schreiben!

```
-- Das hier ist eine Notiz und macht gar nichts.
```

Alle Befehle im Überblick

Chat & Chat-Befehle

`mod.chat(nachricht)`

- **Was es tut:** Schreibt eine Nachricht in den Chat, die alle sehen können.
- **Was du brauchst:**
 - `nachricht`: Der Text, den du senden willst (in Anführungszeichen!).
- **Beispiel:**

```
mod.chat("Willkommen in meiner Welt!")
```

`mod.neuer_befehl(name, funktion)`

- **Was es tut:** Erfindet einen neuen Chat-Befehl, den du mit `/` benutzen kannst.
- **Was du brauchst:**
 - `name`: Der Name für deinen Befehl (z.B. "hallo").
 - `funktion`: Der Code, der ausgeführt wird, wenn du den Befehl eingibst.
- **Beispiel:**

```
-- Erstellt den Befehl /zauber
mod.neuer_befehl("zauber", function()
    mod.chat("Simsalabim!")
end)
```

Welt bearbeiten

`mod.setze_block(block_name, position)`

- **Was es tut:** Setzt einen einzelnen Block an eine bestimmte Stelle.
- **Was du brauchst:**
 - `block_name`: Der Name des Blocks (z.B. `"default:stone"` oder `"default:glass"`).
 - `position`: Der Ort, wo der Block hin soll.
- **Beispiel:**

```
-- Setzt einen Glasblock an die Position x=10, y=5, z=20
mod.setze_block("default:glass", mod.position(10, 5, 20))
```

`mod.wuerfel(block_name, mitte, groesse)`

- **Was es tut:** Baut einen Würfel aus einem Material deiner Wahl.
- **Was du brauchst:**
 - `block_name`: Woraus der Würfel sein soll.
 - `mitte`: Die Position, wo die Mitte des Würfels sein soll.
 - `groesse`: Wie groß der Würfel sein soll (z.B. 5 für 5x5x5 Blöcke).
- **Beispiel:**

```
-- Baut einen Gold-Würfel der Größe 5 an der Spielerposition
mod.wuerfel("default:goldblock", mod.spieler_pos(), 5)
```

`mod.kugel(block_name, mitte, radius)`

- **Was es tut:** Baut eine Kugel.
- **Was du brauchst:**
 - `block_name`: Das Material der Kugel.
 - `mitte`: Die Position der Kugelmitte.
 - `radius`: Wie groß die Kugel sein soll.
- **Beispiel:**

```
-- Baut eine riesige Glaskugel mit Radius 15 um den Spieler
mod.kugel("default:glass", mod.spieler_pos(), 15)
```

`mod.setze_bereich(block_name, position1, position2)`

- **Was es tut:** Füllt einen ganzen Bereich zwischen zwei Ecken mit einem Block.
- **Was du brauchst:**
 - `block_name`: Das Material zum Füllen.
 - `position1`: Die erste Ecke des Bereichs.
 - `position2`: Die zweite, gegenüberliegende Ecke.
- **Beispiel:**

```
-- Erstellt ein Schwimmbecken aus Wasser
local pos1 = mod.position(0, -1, 0)
local pos2 = mod.position(10, -1, 10)
mod.setze_bereich("default:water_source", pos1, pos2)
```

`mod.entferne_block(position)`

- **Was es tut:** Macht einen Block weg (ersetzt ihn durch Luft).
- **Was du brauchst:**
 - `position`: Der Ort des Blocks, der weg soll.
- **Beispiel:**

```
mod.entferne_block(mod.position(10, 5, 20))
```

`mod.baum(position, typ)`

- **Was es tut:** Lässt an einer Stelle einen Baum wachsen.
- **Was du brauchst:**
 - `position`: Wo der Baum wachsen soll.
 - `typ`: Welche Art von Baum (es gibt `baum`, `apfel`, `dschungel`, `urwaldriese`, `tanne`, `schneetanne`, `akazie`, `espe`, `busch`, `blaubeerbusch` und `riesenkaktus`).
- **Beispiel:**

```
-- Lässt eine Tanne an der Spielerposition wachsen  
mod.baum(mod.spieler_pos(), "tanne")
```

Dinge finden & untersuchen

`mod.lese_block(position)`

- **Was es tut:** Schaut nach, welcher Block an einer bestimmten Stelle ist.
- **Was du brauchst:**
 - `position`: Der Ort, an dem du nachschauen willst.
- **Beispiel:**

```
-- Finde heraus, worauf der Spieler gerade steht  
local spieler_pos = mod.spieler_pos()  
local boden_pos = mod.position(spieler_pos.x, spieler_pos.y - 1, spieler_pos.z)  
local block_name = mod.lese_block(boden_pos).name  
mod.chat("Du stehst auf: " .. block_name)
```

`mod.finde_block(startposition, entfernung, block_name)`

- **Was es tut:** Sucht den nächstgelegenen Block einer bestimmten Art in deiner Nähe.
- **Was du brauchst:**
 - `startposition`: Wo die Suche starten soll (z.B. `mod.spieler_pos()`).
 - `entfernung`: Wie weit weg gesucht werden soll (eine Zahl).
 - `block_name`: Welcher Block gesucht wird (z.B. `"default:diamond_ore"`).
- **Beispiel:**

```
-- Finde einen Diamanten in der Nähe (Umkreis von 10 Blöcken)  
local diamant_pos = mod.finde_block(mod.spieler_pos(), 10, "default:diamond_ore")  
if diamant_pos then  
    mod.chat("Ich habe einen Diamanten bei " .. dump(diamant_pos) .. " gefunden!")  
end
```

`mod.finde_bloেকে(startposition, entfernung, block_name)`

- **Was es tut:** Sucht *alle* Blöcke einer bestimmten Art in deiner Nähe.
- **Was du brauchst:**
 - Die gleichen wie bei `mod.finde_block`.
- **Beispiel:**

```
-- Verwandle alle Kohleblöcke in der Nähe in Diamanten
local alle_kohleblöcke = mod.finde_bloেকে(mod.spieler_pos(), 5, "default:coal_ore")
-- Gehe die Liste mit allen gefundenen Positionen durch
for kohle_pos in each(alle_kohleblöcke) do
    mod.setze_block("default:diamond_ore", kohle_pos)
end
```

Eigene Blöcke & Items erfinden

Das ist der spannendste Teil! Hier kannst du deine eigenen Blöcke und Gegenstände mit besonderen Fähigkeiten erfinden.

`mod.neuer_block(name, bild, eigenschaften)`

- **Was es tut:** Erfindet einen komplett neuen Block, den du dann im Inventar findest und in der Welt platzieren kannst.
- **Was du brauchst:**
 - `name`: Der Name für deinen Block (z.B. "Party Block").
 - `bild`: Der Name der Bild-Datei für das Aussehen (z.B. "default_gold_block.png").
 - `eigenschaften`: Eine Liste von Dingen, die der Block tun soll.
- **Mögliche Eigenschaften - Nicht alle müssen angegeben werden!:**
 - `rechtsklick = function(position, gegenstand_in_hand)`
 - **Wann?** Wenn ein Spieler mit der rechten Maustaste auf den Block klickt.
 - **Du bekommst:** Die `position` des Blocks.
 - `linksklick = function(position)`
 - **Wann?** Wenn ein Spieler mit der linken Maustaste auf den Block schlägt.
 - **Du bekommst:** Die `position` des Blocks.
 - `abbauen = function(position)`
 - **Wann?** Genau in dem Moment, wenn der Block fertig abgebaut ist, aber bevor er verschwindet.
 - **Du bekommst:** Die `position` des Blocks.
- **Beispiel:** Ein Teleporter-Block, der dich an einen hohen Ort bringt.

```
mod.neuer_block("Teleporter", "default_obsidian.png", {
    rechtsklick = function(position)
        mod.chat("Teleportiere!")
        -- Teleportiert den Spieler 100 Blöcke über den Block
        local ziel_pos = mod.position(position.x, position.y + 100, position.z)
        mod.teleportiere_spieler(ziel_pos)
    end,
    linksklick = function(position)
        mod.chat("Das ist ein Teleporter! Klicke mit Rechts, um ihn zu benutzen.")
    end,
    abbauen = function(position)
        mod.chat("Der Teleporter wurde zerstört!")
    end
})
```

`mod.neues_item(name, bild, eigenschaften)`

- **Was es tut:** Erfindet einen neuen Gegenstand (Item), den du in der Hand halten und benutzen kannst. Du findest ihn danach im Inventar.
- **Was du brauchst:**
 - `name`: Der Name für dein Item (z.B. "Baum-Zauberstab").
 - `bild`: Der Name der Bild-Datei für das Item (z.B. "default_stick.png").
 - `eigenschaften`: Eine Liste von Dingen, die das Item tun soll.
- **Mögliche Eigenschaften - Nicht alle müssen angegeben werden!:**
 - `platzieren = function(position, spieler)`
 - **Wann?** Wenn du versuchst, das Item wie einen Block auf den Boden oder an eine Wand zu setzen.
 - **Du bekommst:** Die `position`, wo das Item platziert wird.
 - `linksklick = function(position_vom_block, spieler)`
 - **Wann?** Wenn du mit dem Item in der Hand mit der linken Maustaste auf einen Block in der Welt klickst.
 - **Du bekommst:** Die `position_vom_block`, auf den du geklickt hast.
 - `rechtsklick = function(spieler)`
 - **Wann?** Wenn du mit dem Item in der Hand einen Rechtsklick machst (also in die Luft, nicht auf einen Block).
- **Beispiel:** Ein Zauberstab, der Bäume wachsen lässt und Blöcke in Gold verwandelt.

```
mod.neues_item('Zauberstab der Natur', 'default_stick.png', {  
  -- Mit Rechtsklick einen Baum wachsen lassen  
  platzieren = function(position)  
    mod.baum(position, 'apfel')  
  end,  
  -- Mit Linksklick einen Block in Gold verwandeln  
  linksklick = function(position_vom_block)  
    mod.setze_block('default:goldblock', position_vom_block)  
  end,  
  -- Mit Rechtsklick in die Luft einen Partikeleffekt machen  
  rechtsklick = function()  
    mod.partikel(mod.spieler_pos(), 'heart.png', 200, 10)  
  end,  
})
```

Spieler & Physik

`mod.spieler_pos()`

- **Was es tut:** Gibt dir die aktuelle Position des Spielers. Sehr nützlich!
- **Beispiel:**

```
-- Setzt einen Fackelblock genau unter den Spieler
mod.setze_block("default:torch", mod.spieler_pos())
```

`mod.teleportiere_spieler(position)`

- **Was es tut:** Teleportiert den Spieler an einen anderen Ort.
- **Was du brauchst:**
 - `position`: Wohin du reisen möchtest.
- **Beispiel:**

```
-- Teleportiert den Spieler 100 Blöcke in die Luft
mod.teleportiere_spieler(mod.position(0, 100, 0))
```

`mod.setze_schwerkraft(staerke)`

- **Was es tut:** Ändert die Schwerkraft.
- **Was du brauchst:**
 - `staerke`: Eine Zahl. `1` ist normal. Weniger als 1 (z.B. `0.1`) ist wie auf dem Mond, mehr als 1 (z.B. `3`) zieht dich stark nach unten.
- **Beispiel:**

```
-- Fast keine Schwerkraft!
mod.setze_schwerkraft(0.1)
```

`mod.setze_sprungkraft(staerke)`

- **Was es tut:** Lässt dich höher springen.
- **Was du brauchst:**
 - `staerke`: Eine Zahl. `1` ist normal. `3` lässt dich super hoch springen!
- **Beispiel:**

```
-- Super-Sprung an!
mod.setze_sprungkraft(3)
```

`mod.setze_geschwindigkeit(tempo)`

- **Was es tut:** Macht dich schneller oder langsamer.
- **Was du brauchst:**
 - `tempo`: Eine Zahl. `1` ist normal. `5` ist super schnell!
- **Beispiel:**

```
-- RENN!
mod.setze_geschwindigkeit(5)
```

Besondere Effekte

`mod.partikel(position, bild, anzahl, reichweite)`

- **Was es tut:** Erzeugt coole Partikel-Effekte (wie Funken oder Rauch).
- **Was du brauchst:**
 - `position`: Wo die Partikel erscheinen sollen.
 - `bild`: Das Aussehen der Partikel (z.B. `"fire_basic.png"`).
 - `anzahl`: Wie viele Partikel es sein sollen.
 - `reichweite`: Wie weit die Partikel sich verteilen.
- **Beispiel:**

```
-- Erzeugt 100 Feuer-Partikel um den Spieler
mod.partikel(mod.spieler_pos(), "fire_basic.png", 100, 3)
```

`mod.pfeil(funktion)`

- **Was es tut:** Führt einen Code aus, immer wenn ein Pfeil einen Block trifft.
- **Was du brauchst:**
 - `funktion`: Der Code, der ausgeführt wird. Er bekommt die Position, wo der Pfeil eingeschlagen ist.
- **Beispiel:**

```
-- Verwandelt jeden Block, den ein Pfeil trifft, in einen Goldblock
mod.pfeil(function(treffer_position)
  mod.setze_block("default:goldblock", treffer_position)
end)
```

`mod.schiesse_projektil(partikel_bild, funktion, verzoeigerung, reichweite)`

- **Was es tut:** Schießt ein magisches Projektil in Blickrichtung.
- **Was du brauchst:**
 - `partikel_bild`: Das Aussehen der Flugbahn des Projektils.
 - `funktion`: Was passieren soll, wenn das Projektil auftrifft.
 - `verzoeigerung`: Eine kleine Zahl für die Geschwindigkeit (z.B. `0.1`).
 - `reichweite`: Wie weit das Projektil fliegt (z.B. `100`).
- **Beispiel:**

```
-- Ein Zauber, der beim Aufprall eine kleine Kugel aus Glas erschafft
mod.neuer_befehl("glaskugel", function()
  mod.schiesse_projektil("bubble.png", function(treffer_position)
    mod.kugel("default:glass", treffer_position, 3)
  end, 0.05, 50)
end)
```

`mod.wiederhole_alle(sekunden, funktion)`

- **Was es tut:** Führt einen Code immer wieder aus, nach einer bestimmten Zeit.
- **Was du brauchst:**
 - `sekunden`: Der Abstand in Sekunden zwischen den Wiederholungen.
 - `funktion`: Der Code, der wiederholt werden soll.
- **Beispiel:**

```
-- Schreibt alle 10 Sekunden "Hallo" in den Chat
mod.wiederhole_alle(10, function()
  mod.chat("Schon wieder 10 Sekunden vorbei!")
end)
```