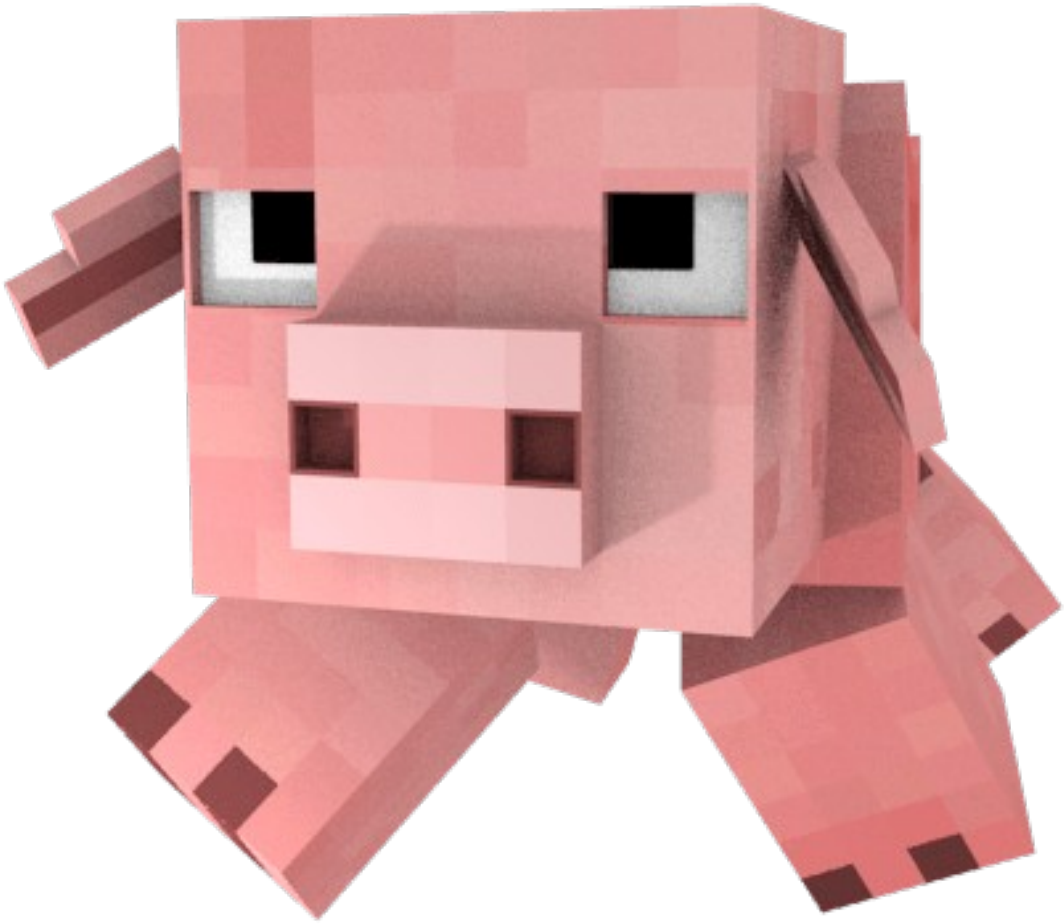
TOMMASO ANZIDEI

IL PICCOLO PROGRAMMATORE MINECRAFT



SOMMARIO

1 INTRODUZIONE

Questo documento è un piccolo manuale di programmazione Python in ambiente Minecraft per mentor coderdojo. L'obiettivo del manuale è insegnare il linguaggio di programmazione Python ad adolescenti che abbiano a disposizione un client Minecraft per Raspberry o PC.

Nel documento si presentano brevemente Minecraft, le varie modalità di integrazione con Python ed alcuni esercizi adatti ad un coderdojo.

Il materiale presentato è una traduzione libera di due tutorial disponibili in rete alle pagine:

https://www.raspberrypi.org/learning/getting-started-with-minecraft-pi/worksheet/

http://www.instructables.com/id/Python-coding-for-Minecraft/

E' importante qui ricordare che l'ambiente migliore col quale sperimentare la programmazione Minecraft con Python è senz'altro Raspberry PI: infatti il sistema operativo Raspbian è distribuito con una versione di Minecraft già integrata con Python. Chi non possiede un Raspberry può comunque emularlo sul proprio PC oppure modificare la propria installazione Minecraft attraverso un mod che implementa l'integrazione con Python (come vedremo nel prossimo paragrafo).

Nel corso degli esercizi si elencano via via **i concetti Python** che lo studente incontra e il mentor può approfondire. A volte i concetti vengono elencati prima che il mentor li spieghi.

2 INSTALLAZIONE

Ci sono quattro per predisporre un ambiente Minecraft/Python:

(1) Utilizzare Raspberry Pi: in questo caso non c'è niente da fare. Questa è l'installazione di riferimento in questa guida. Le altre modalità non sembrano altrettanto stabili e semplici come questa.

(2) Emulare Raspbian sul proprio PC. In questo caso è necessario installare Quemu, procurarsi un'immagine Raspbian ed avviarla. Detto così è semplice, in pratica alcune volte si devono superare varie difficoltà[1].

(3) Replicare l'integrazione nativa di Minecraft presente in Raspberry nella propria installzzazione Minecraft Windows/MacOSX/Linux. Seguire le istruzioni alla pagina http://www.instructables.com/id/Python-coding-for-Minecraft/[2]

(4) Utilizzare Minetest. Minetest è un clone Open Source di Minecraft. Lo si può scaricare all'indirizzo http://www.minetest.net/ e poi replicare l'integrazione nativa di Minecraft presente in Raspberry nella propria installzzazione Minetest seguendo le istruzioni presenti in https://forum.minetest.net/viewtopic.php?t=13316

A costo di essere noiosi ripetiamo che la versione Raspberry Pi di Minecraft è quella che dovrebbe essere di riferimento per chi vuole sperimentare Python con Minecraft. Le altre due modalità non sono altrettanto user-friendly. In ogni caso tutte e tre non utilizzano server Minecraft.

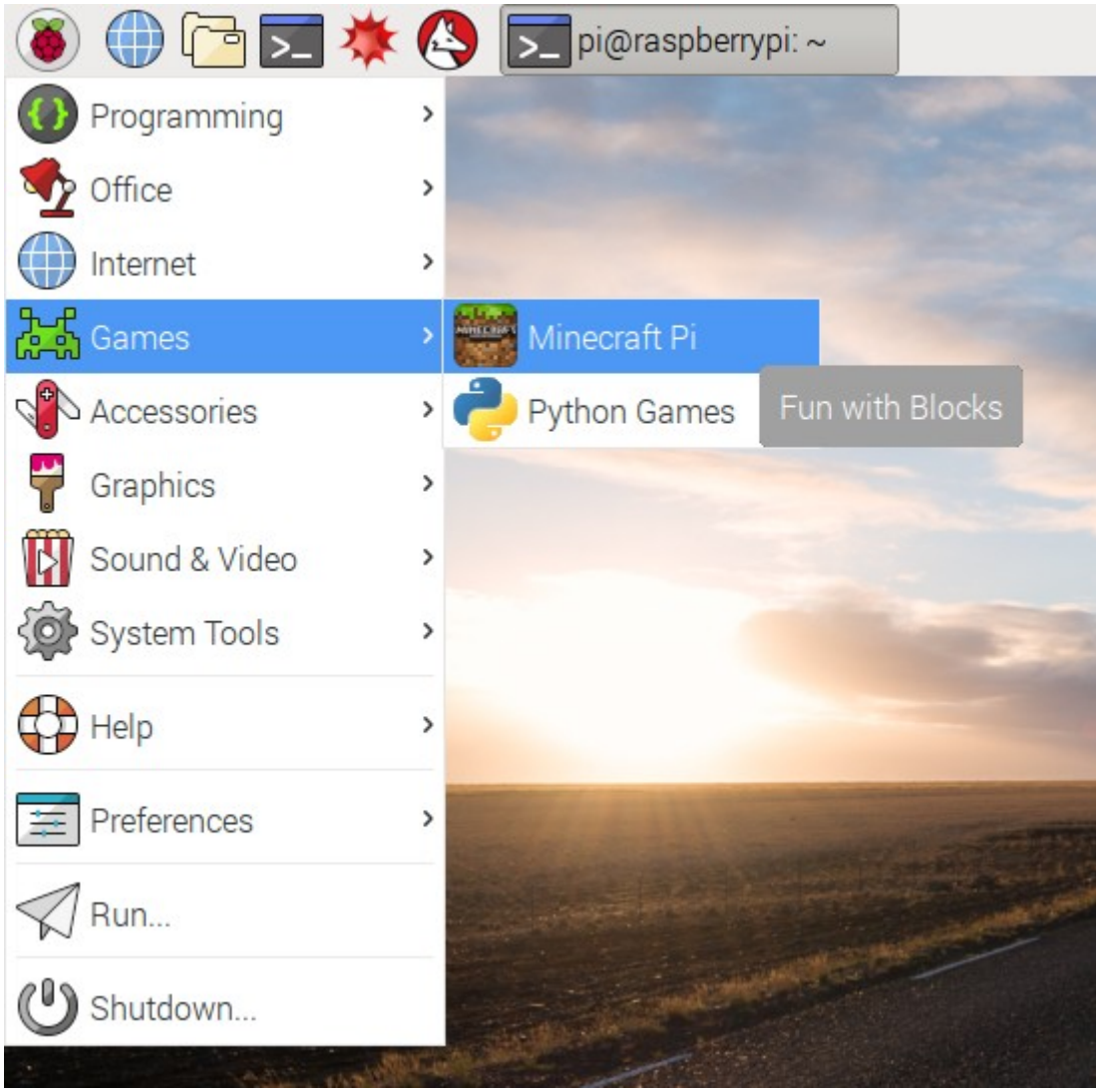
1 Si vedano i link: https://www.pcsteps.com/1199-raspberry-pi-emulation-for-windows-qemu/, http://embedonix.com/articles/linux/emulating-raspberry-pi-on-linux/, https://sourceforge.net/projects/rpiqemuwindows/ e https://sourceforge.net/projects/raspberrypiemulator/

2 L'autore ha scoperto che le cose funzionano meglio se, al momento dell'estrazione del file mods.zip, si corregge il nome dei folder in modo che rispecchi esattamente la versione Minecraft. Per esempio il file mods.zip contiene una cartella chiamata 1112. Questa cartella, dopo l'estrazione, dovrebbe essere rinominata 1.11.2

3 COME SI GIOCA A MINECRAFT

A beneficio dei mentor che non conoscono Minecraft accenniamo qui a come si gioca. Facciamo riferimento a Raspberry PI: Windows, Linux e MacOSX presentano modalità di gioco del tutto analoghe.

Minecraft è un gioco di tipo sandbox nel quale è possibile costruire il proprio mondo. Per eseguire Minecraft in Raspberry PI è sufficiente cliccare l'icona nel menu oppure invocare il comando minecraft-pi nel terminale:



Quando compare il menu principale di Minecraft si scelga il comando **Start Game** seguito da **Create New**. Comparirà un nuovo mondo Minecraft!

Si usi il mouse per orientarsi e i seguenti comandi da tastiere per eseguire azioni:

TASTO	AZIONE
W	Avanti
A	Sinistra
S	Indietro
D	Destra
E	Inventario
SPAZIO	Salto
DOPPIO SPAZIO	Volo/Caduta
ESC	Pausa/Menu del gioco
TAB	Rilascio del cursore del mouse

Per eseguire azioni si fa uso degli oggetti nell'inventario che sono direttamente raggiungibili con il mouse se presenti nel pannello inferiore dello schermo oppure attraverso il comando **E**:

E' possibile volare premendo due volte il tasto **spazio**, fermare il volo rilasciando il medesimo tasto e tornare sul terreno premendo di nuovo due volte il tasto spazio:

Utilizzando la spada è possibile rimuovere blocchi davanti a noi oppure scavare. Se invece si utilizza un blocco con il tasto destro lo si pone davanti a noi, col sinistro lo si rimuove.

4 ESERCIZIO 1: PYTHON E MINECRAFT

E' possibile interagire con i mondi creati con Minecraft attraverso il linguaggio di programmazione Python. Vediamo come fare.

CIAO MONDO

Se si utilizza Raspberry, o la sua emulazione tramite Qemu, è sufficiente seguire i passi:

(1) Mentre Minecraft è in esecuzione si prema il tasto **TAB** (così facendo è possibile utilizzare il mouse fuori dalla finestra di Minecraft)

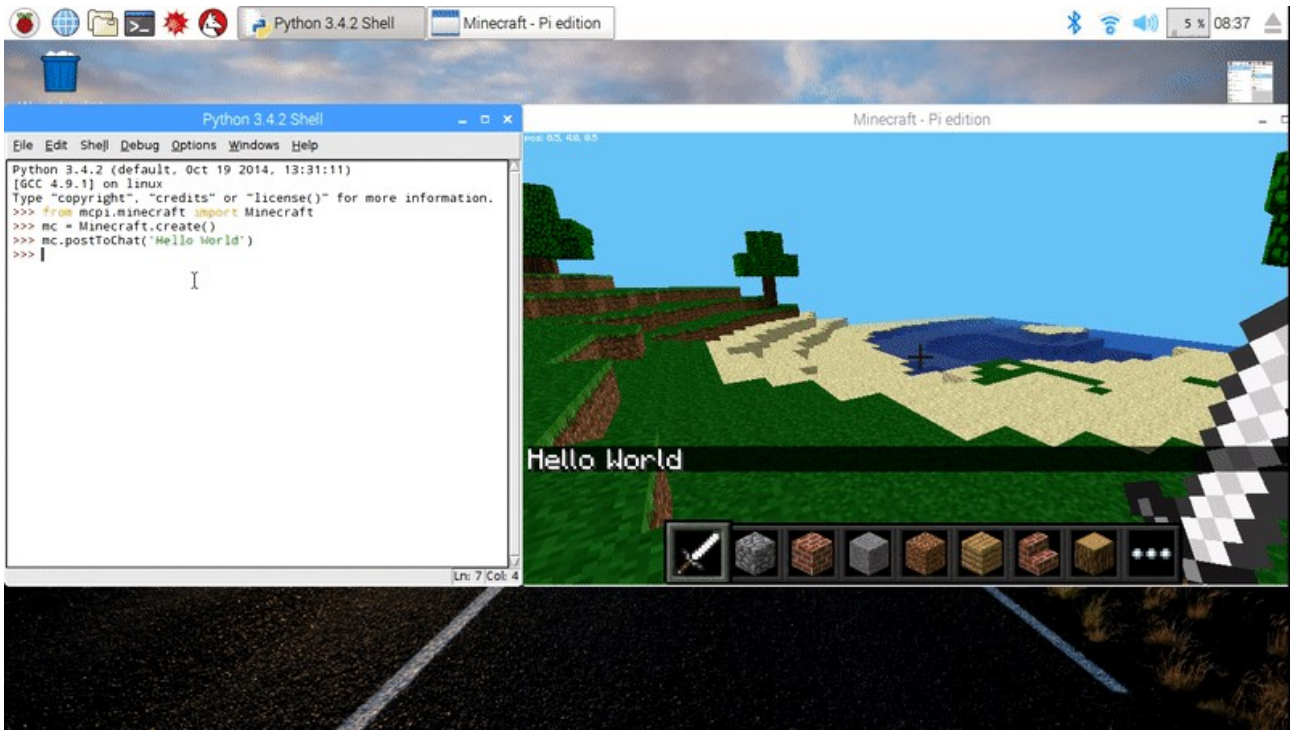
(2) Si apra Python3 dal menu di comandi di Raspberry e lo si affianchi alla finestra Minecraft

(3) Nell'interprete Python si scriva digitano **Enter** dopo ogni linea:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
mc.postToChat("Hello world")
```

(4) Si verifichi che nella finestra Minecraft compaia il messaggio “Hello world”

(5) In alternativa al passo 3 si può, dall'interprete Python, scrivere le tre righe di codice in un file (**File → New window** e **File → Save**) ed eseguirlo con il tasto **F5** dopo averlo salvato con **Ctrl + S**.



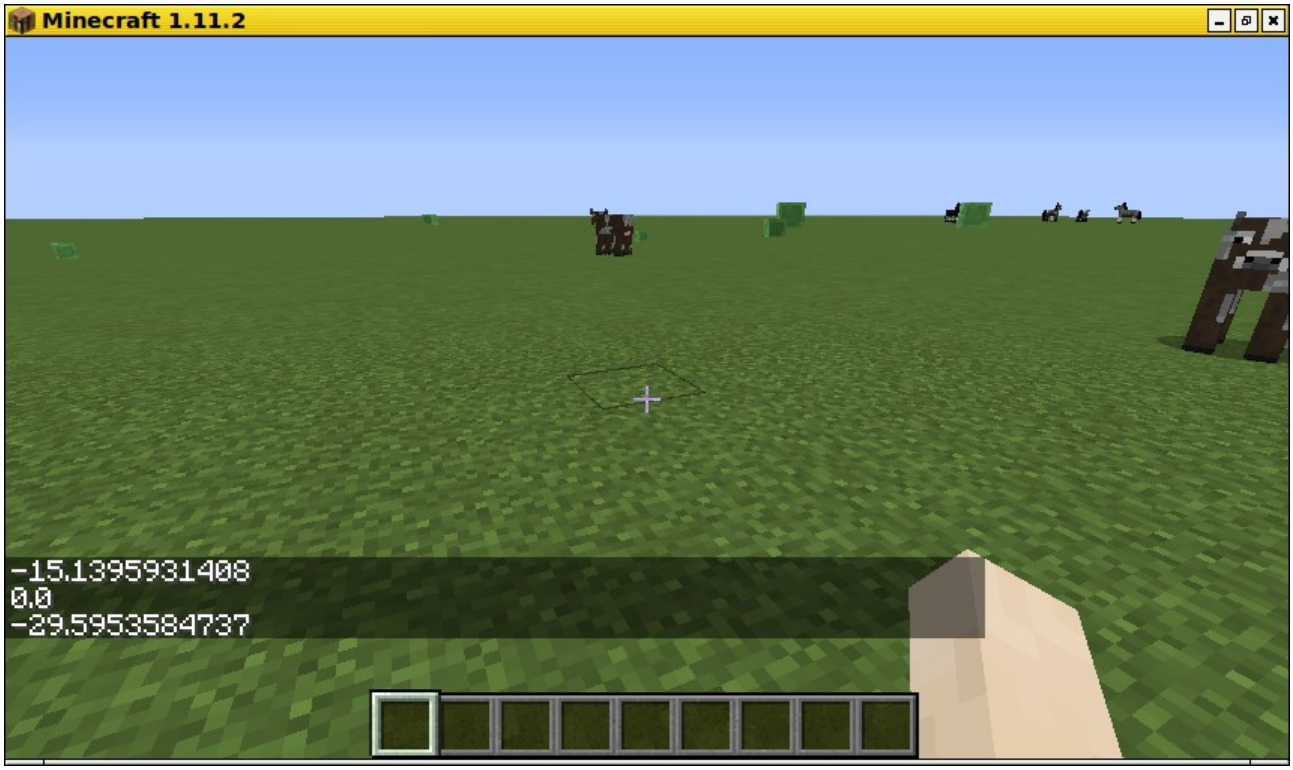
Se invece si utilizza Mincraft su PC o Minetest (entrambi con la mod indicata al capitolo 2) si dovrà, con un qualsiasi editor, creare un file di testo chiamato hello.py contente le tre righe precedentemente menzionate, salvare il file nella cartella mcpipy che si trova all'interno dell cartella Minecraft (dentro la cartella .minecraft della cartella appdata dell'utente corrente di Windows oppure dentrol cartella .minecraft della cartella $HOME di Linux). Si usi il tasto **T** (che apre la finextra dei comandi in basso) e si digiti **/python hello** (ovvero il nome del file hello.py senza estenzione) Il risultato sarà lo stesso.

CONCETTI PYTHON CHE HAI INCONTRATO: IMPORT DA UN PACKAGE, CLASSE, OGGETTO, METODO, PARAMETRO, STRINGA

LA MIA POSIZIONE

Vediamo adesso come conoscere la propria posizione (in coordinate spaziali) scrivendo ed eseguendo il file miaposizione.py:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
x, y, z = mc.player.getPos()
mc.postToChat(x)
mc.postToChat(y)
mc.postToChat(z)
```



CONCETTI PYTHON CHE HAI INCONTRATO: NUMERO DECIMALE

TELETRASPORTO

Prova adesso a scrivere ed eseguire il seguente file, teletrasporto.py:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
x, y, z = mc.player.getPos()
mc.player.setPos(x, y+100, z)
```

Il risultato sarà che volerai 100 blocchi in alto e poi tornerai a terra.

DEPOSITARE UN BLOCCO

Scrivi adesso questo file (chiamalo unblocco.py):

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

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

Il risultato sarà che avrai collocato vicino a te un blocco:

E' importante capire come funziona il metodo setBlock: i primi tre parametri corrispondono alle tre coordinate spaziali nelle quali depositare il blocco mentre la quarta è il tipo di materiale (in questo caso 1 che vuol dire pietra). In Minecraft puoi usare blocchi fatti da tantissimi tipi di materiale tra cui, per esempio:

Aria: 0

Erba: 2

Terreno: 3

BLOCCHI COME COSTANTI

Se conosci il nome dei blocchi puoi usarli al posto dei loro identificativi numerici. Per prima cosa dovrai aggiungere una import e successivamente potrai utilizzare il nome del blocco

```
from mcpi.minecraft import Minecraft
from mcpi import block
```

```
mc = Minecraft.create()
x, y, z = mc.player.getPos()
mc.setBlock(x+1, y, z, block.STONE.id)
```

Come vedi adesso l'istruzione è più semplice da leggere. Ecco altri nome blocco che puoi usare:

```
WOOD_PLANKS
WATER_STATIONARY
GOLD_ORE
GOLD_BLOCK
DIAMOND_BLOCK
NETHER_REACTOR_CORE
```

BLOCCHI COME VARIABILI

Puoi riscrivere gli esempi precedenti così:

```
from mcpi.minecraft import Minecraft
from mcpi import block

mc = Minecraft.create()
x, y, z = mc.player.getPos()
dirt = 3
mc.setBlock(x+1, y, z, dirt)
```

Come vedi hai dichiarato una variabile, dirt, e le hai assegnato il valore 3. L'uso delle variabili rende il tuo codice ordinato e leggibile.

CONCETTI PYTHON CHE HAI INCONTRATO: VARIABILE

BLOCCHI SPECIALI

Alcuni blocchi sono speciali nel senso che hanno proprietà aggintive. Per esempio la lana (codice 35) ha la proprietà aggiuntiva colore. In questo caso il metodo setBlock può essere invocato con un parametro aggiuntivo, il quarto, che per la lana è appunto il colore.

```
from mcpi.minecraft import Minecraft
from mcpi import block

mc = Minecraft.create()
lana = 35
x, y, z = mc.player.getPos()
mc.setBlock(x+1, y, z, lana, 1)
```

In questo caso abbiamo usato il colore 1 che è l'arancione. Altri colori della lana sono il banco (0), magenta (2), blu (3), giallo (4) etc.

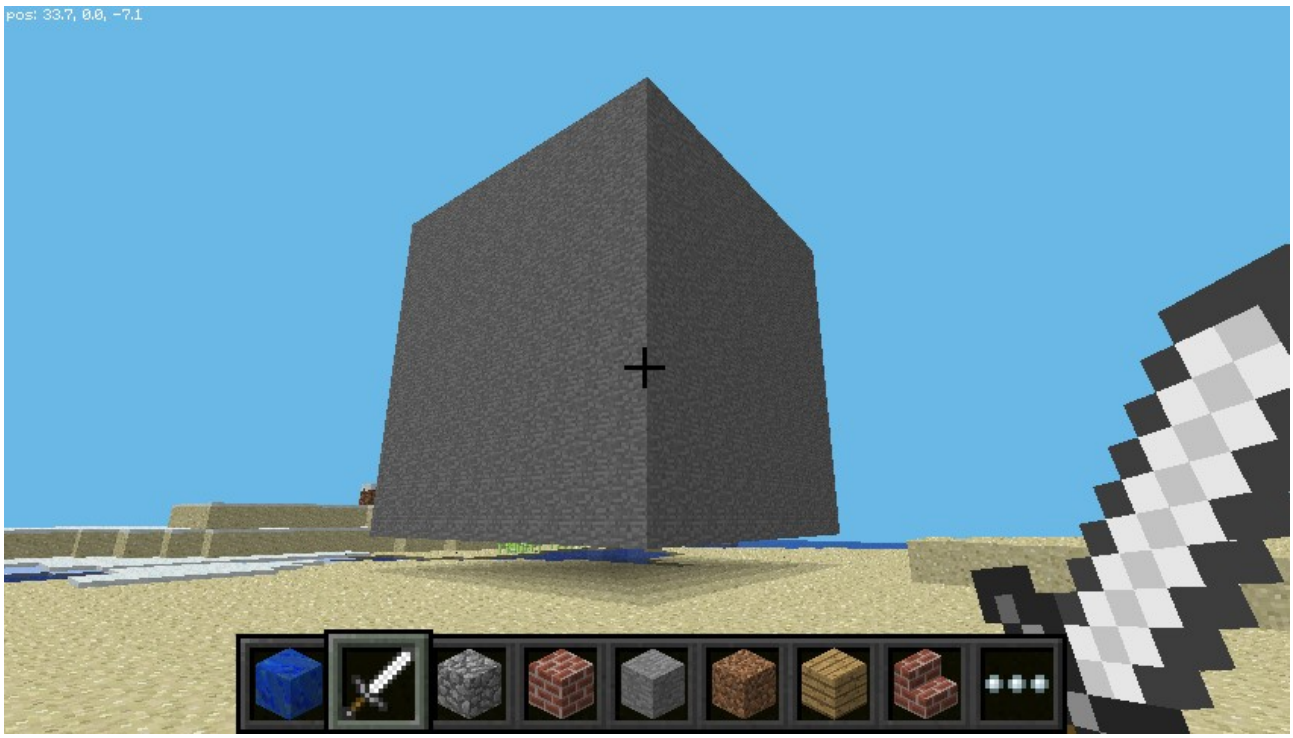
CONCETTI PYTHON CHE HAI INCONTRATO: POLIMORFISMO DELLE FUNZIONI

BLOCCHI MULTIPLI

Puoi anche depositare più blocchi in una volta col comando setBlocks (nota il plurale):

```
from mcpi.minecraft import Minecraft
from mcpi import block
mc = Minecraft.create()
stone = 1
x, y, z = mc.player.getPos()
mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, stone)
```

Come vedi setBlocks richiede le coordiante del blocco iniziale (i primi tre parametri) e quello finale (i successivi tre parametri) nonché il tipo di blocco (l'ultimo parametro).


pos: 33.7, 0.0, -7.1

5 ESERCIZIO 2: DI PIU' SU PYTHON

Python è un linguaggio di programmazione molto potente ma facile da usare. Adesso elenchiamo alcuni esempi che permettono di imparare Python all'interno di Minecraft.

STAMPARE

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
mc.postToChat("Hello world")
```

Questo lo avevamo già visto.

OPERAZIONI ARITMETICHE

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
a = 2
b = 2
mc.postToChat(a + b)
```

a e b sono due **variabili**.

Prova tutte le operazioni che ti vengono in mente.

STRINGHE

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
stringa = “Minecraft”
mc.postToChat(len (stringa))
```

Abbiamo calcolato la lunghezza di una parola.

CICLO WHILE

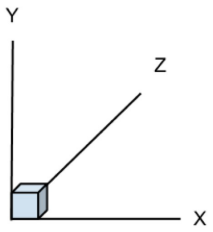
```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
a = 0
while a < 10:
      a = a + 1
      mc.postToChat(a)
```

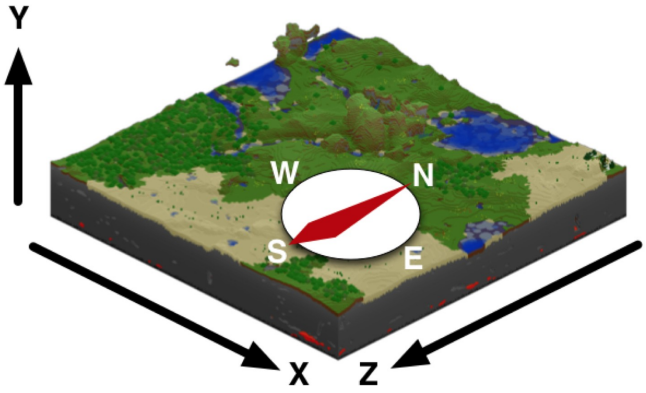
Abbiamo ripetuto 10 volte la stampa di un numero incrementandolo ogni volta.

DECISIONI

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
x, y, z = mc.player.getPos()
if y > 0:
      mc.postToChat(“Hey! Sei sopra il livello del mare!”)
else:
      mc.postToChat(“Hey! Sei sotto il livello del mare!”)
```

Abbiamo controllato l'altezza cui si trova il giocatore. Il sistema di coordinate di Minecraft è il seguente:



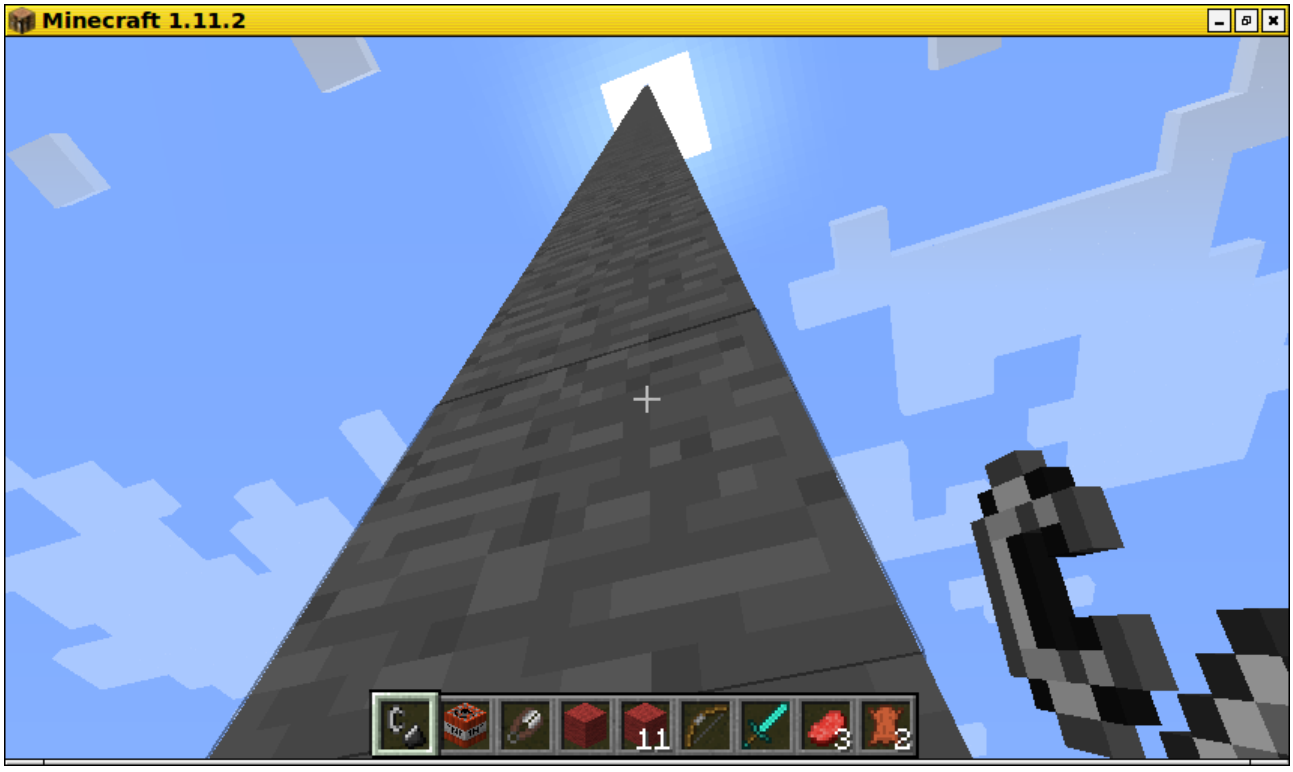


FUNZIONI

Le funzioni ti permettono di scrivere del codice da riutilizzare in futuro.

```
from mine import *
mc = Minecraft()
x, y, z = mc.player.getPos()
def torre (altezza) :
     a = 0
     while a <= altezza:
         mc.setBlock(x+1, y + a, z , 1)
torre(100)
mc.postToChat("Torre!")
```

Abbiamo scritto una funzione, torre, alla quale possiamo passare un parametro, altezza. Possiamo quindi chiamare la funzione torre con altezza diverse per ottenere torri diverse.


Minecraft 1.11.2
11
3
2

6 ESERCIZIO 3: BUTTA FIORI MENTRE CAMMINI!

Adesso prova a scrivere questo codice Python:

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

mc = Minecraft.create()

flower = 38

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

Il codice che hai scritto contiene un ciclo while durante il quale, ad intervalli di tempo molto brevi, un fiore viene collocato in prossimità della tua posizione col risultato di seguito mostrato:

Puoi lasciare i fiori anche in cielo mentre voli:

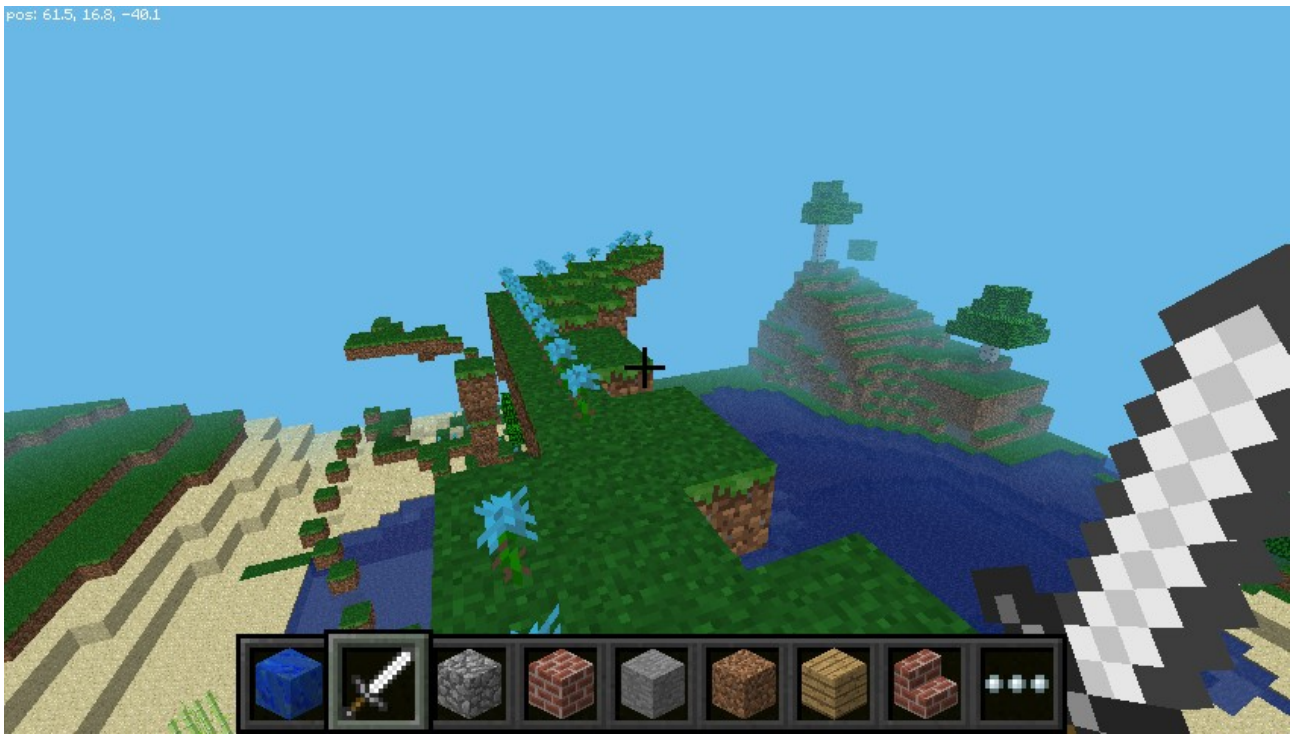
CONCETTI PYTHON CHE HAI INCONTRATO: CICLO WHILE, SLEEP

Se però vuoi lasciare fiori solo quando cammini sull'erba puoi implementare questo ragionamento: prima di tutto devi capire il tipo di blocco sul quale stai camminando con il metodo getBlock() e poi solo se il blocco sul quale ti trovi è un blocco di erba rilasci un fiore:

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

mc = Minecraft.create()
grass = 2
flower = 38

while True:
    x, y, z = mc.player.getPos()  # player position (x, y, z)
    block_beneath = mc.getBlock(x, y-1, z)  # block ID

    if block_beneath == grass:
        mc.setBlock(x, y, z, flower)
    sleep(0.1)
```

Come vedi con getBlock si recupera il tipo di blocco che si trova subito al di sotto di noi (ecco perchè usiamo y-1 e non y).

CONCETTI PYTHON CHE HAI INCONTRATO: CICLO WHILE, IF-THEN

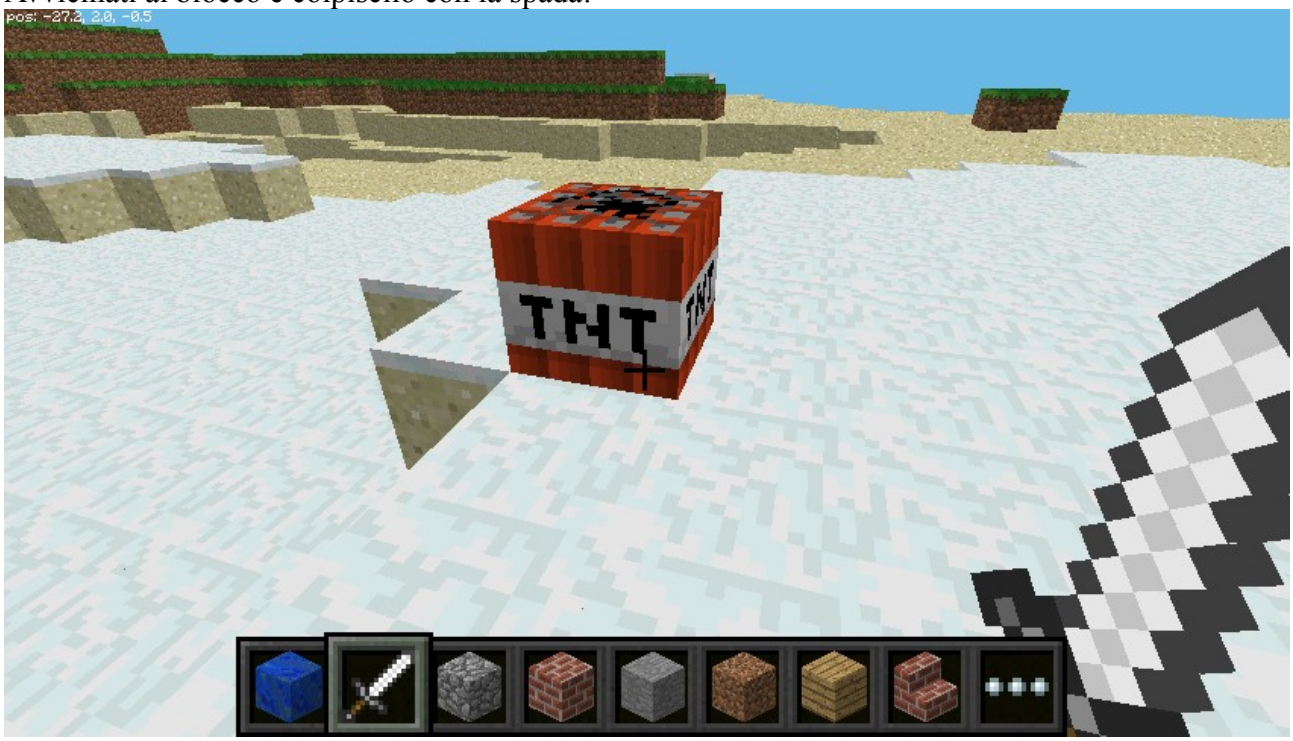

pos: 61.5, 16.8, -40.1

7 ESERCIZIO 4: GIOCHIAMO CON LA TNT

Adesso ci divertiamo con la TNT. Scrivi questo codice:

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

mc = Minecraft.create()

x, y, z = mc.player.getPos()  # player position (x, y, z)
tnt = 46
mc.setBlock(x, y, z, tnt, 1)
```

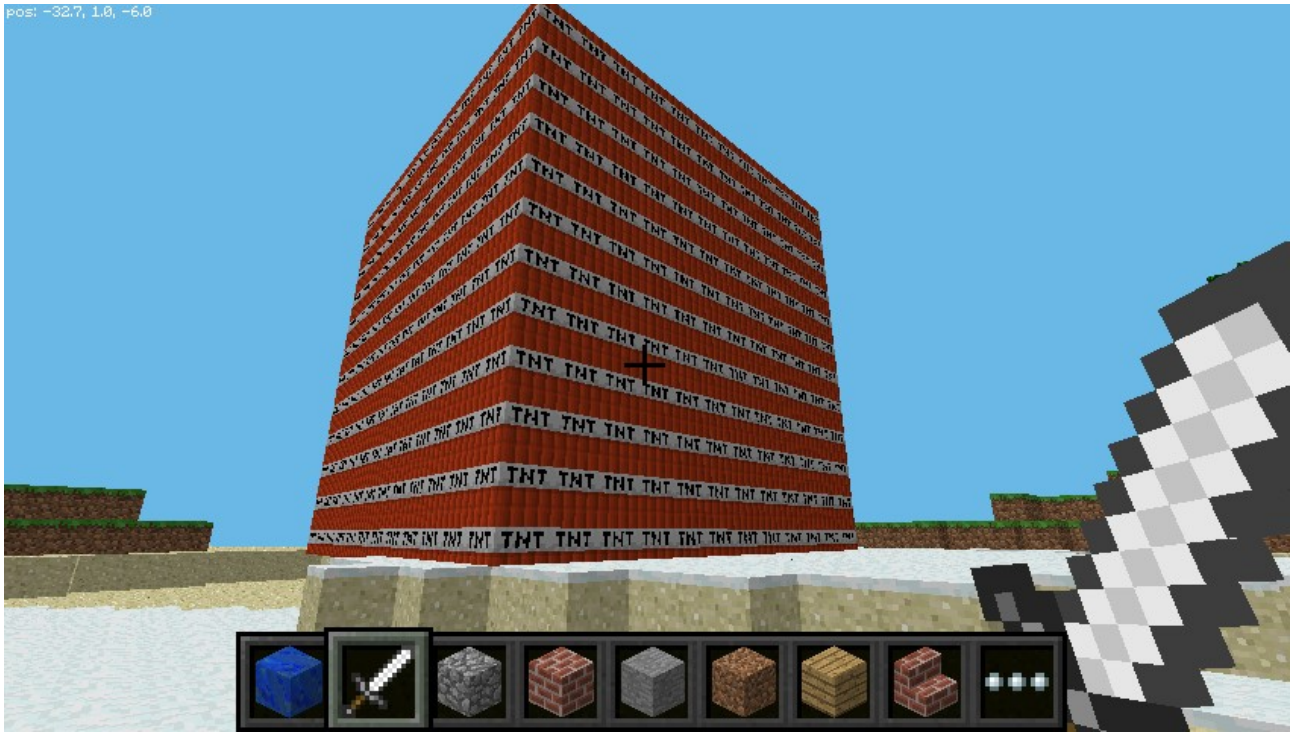
Avvicinati al blocco e colpiscilo con la spada!



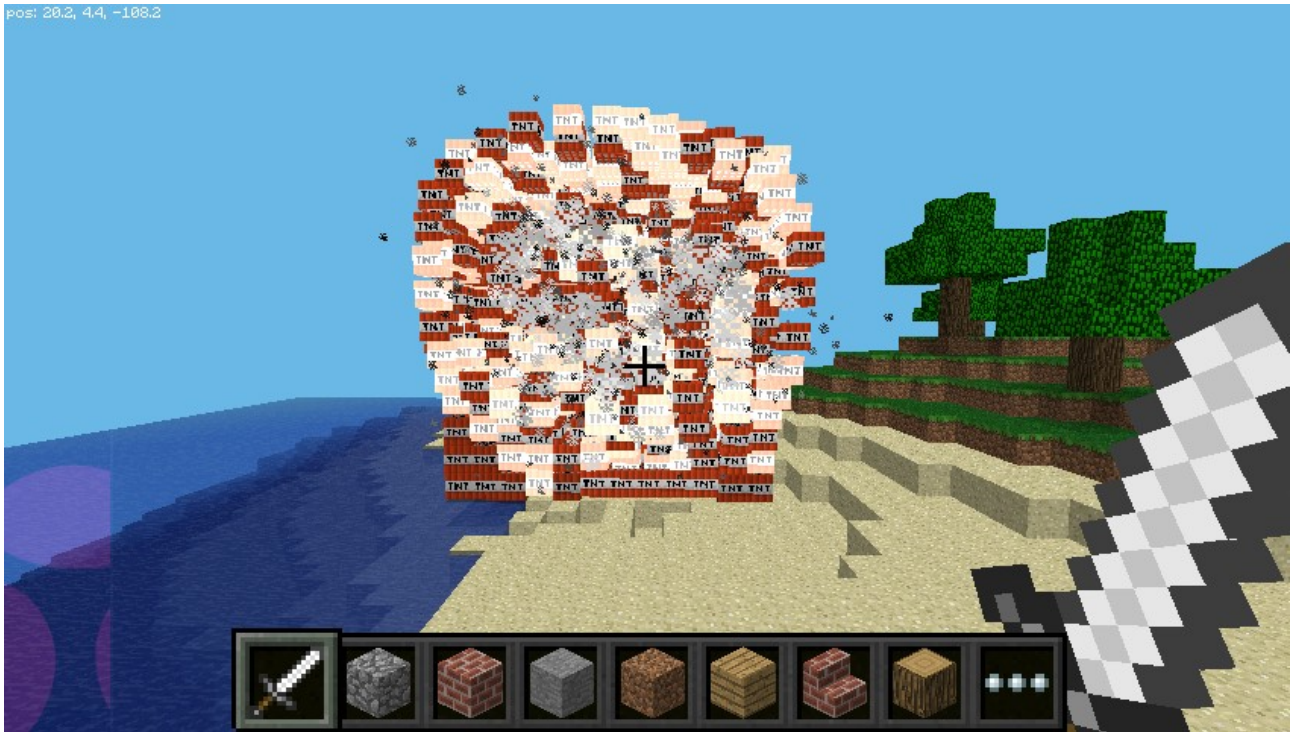
Adesso però divertiamoci davvero! Facciamo un blocco moooolto più grande:

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

mc = Minecraft.create()

x, y, z = mc.player.getPos()  # player position (x, y, z)
tnt = 46
mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, tnt, 1)
```

Stavolta il blocco fa molta più impressione!



Se lo colpisci (e ti allontani molto velocemente!):



8 ESERCIZIO 5: DIVERTIAMOCI CON LA LAVA

Un blocco divertente è la lava che scorre. Prova a scrivere:

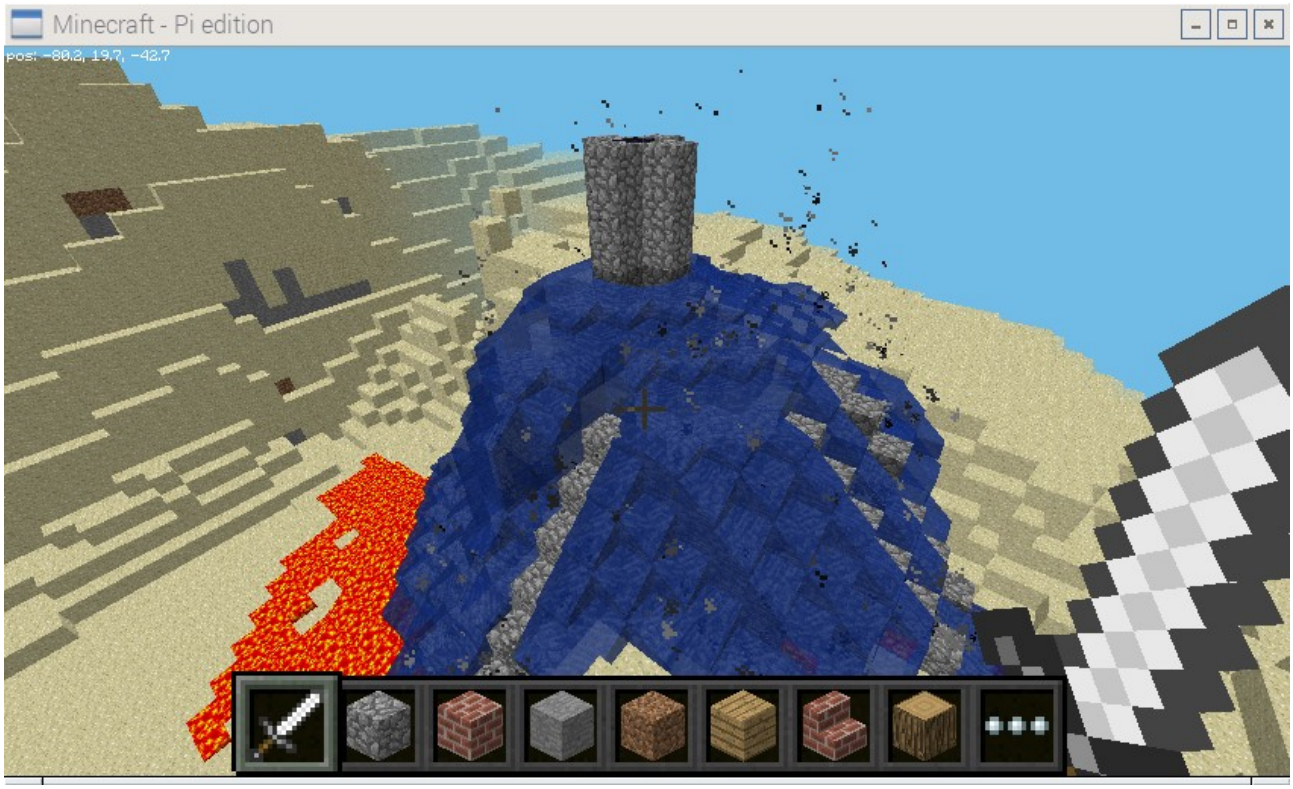
```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()

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

lava = 10

mc.setBlock(x+3, y+3, z, lava)
```

Vedrai veramente la lava scorrere dal blocco che hai depositato. La lava che scorre dopo un po diventa roccia.

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

mc = Minecraft.create()

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

lava = 10
water = 8
air = 0

mc.setBlock(x+3, y+3, z, lava)
sleep(20)
mc.setBlock(x+3,y+5, z, water)
sleep(4)
mc.setBlock(x+3, y+5, z, air)
```

CONCETTI PYTHON CHE HAI INCONTRATO: SLEEP

Nella prossima immagine il risultato.



9 ESERCIZIO 7: IL DRAGONE

Ora guardiamo cosa può fare Phyton. Dalla versione PC di Minecraft digita **T** e poi:

```
/python render dragon
```

Ecco qua!

10 Esercizio 6: Geometria della tartaruga

In preparazione.

11 Esercizio 7: Ancora tartarughe

In preparazione.

12 Esercizio 8: Frattali L-system

In preparazione.

13 Esercizio 9: Geometria cartesiana

In preparazione.

14 Esercizio 10: Superfici parametriche

In preparazione.

15 Esercizio 11: Nodi

In preparazione.