

## Minecraft: Pi Edition Python Cheat Sheet

---

Source: <https://www.stuffaboutcode.com/p/minecraft-api-reference.html>

### STARTER CODE

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

mc = Minecraft.create()
```

### BLOCK FUNCTIONS

---

`mc.getBlock(x, y, z)`

Returns the numeric ID of the block at the specified coordinates.

Example: `mc.getBlock(1, 0, -1)`

---

`mc.getBlockWithData(x, y, z)`

Returns a Python object for the block at the specified coordinates.

Example: `mc.getBlockWithData(1, 0, -1)`

---

`mc.setBlock(x, y, z, BlockID, [metadata])`

Sets a block of the given ID and optional metadata to the specified coordinates.

Example: `mc.setBlock(1, 0, -1, block.DIRT.id)`

Example: `mc.setBlock(1, 0, -1, block.WOOD.id, 1)`

---

`mc.setBlocks(x0, y0, z0, x1, y1, z1, BlockID, [metadata])`

Sets a cuboid of blocks of the given ID and optional metadata to the specified coordinates.

Example: `mc.setBlocks(0, 0, 0, 3, 3, 3, block.DIRT.id)`

Example: `mc.setBlocks(0, 0, 0, 3, 3, 3, block.WOOD.id, 1)`

## MULTIPLAYER-CENTRIC PLAYER FUNCTIONS

---

### `mc.getPlayerEntityIds()`

Returns a list of player IDs. Each ID can be accessed by index, starting at 0.

**Starter code:**

```
players = mc.getPlayerEntityIds()
playerOne = players[0]

# Uncomment if there is a second player
#playerTwo = players[1]
```

---

### `mc.getPos(PlayerID)`

Returns the coordinates of the entity with the given ID.

Example: `coordinates = mc.entity.getPos(25)`

Example: `coordinates = mc.entity.getPos(playerOne)`

---

### `mc.setPos(PlayerID, x, y, z)`

Sets the position of the given player ID to the specified coordinates. **Does not work in multiplayer** 🚫.

Example: `mc.entity.setPos(playerOne, 0, 500, 0)`

---

### `mc.getTilePos(PlayerID)`

Returns the coordinates of the block that the player with the given ID is standing on.

Example: `mc.entity.getTilePos(playerOne)`

---

### `mc.setTilePos(PlayerID, x, y, z)`

Sets the position of the given player ID to be standing on the specified coordinates.

Example: `mc.entity.setTilePos(playerOne, 0, 10, 0)`

## SINGLEPLAYER-CENTRIC PLAYER FUNCTIONS

---

```
mc.player.getPos()
```

Returns a the coordinates of the player.

---

```
mc.player.setPos(x, y, z)
```

Sets the position of the player to the specified coordinates.

Example: `mc.player.setPos(0, 10, 0)`

---

```
mc.player.getTilePos()
```

Returns the coordinates of the block that the player is standing on.

---

```
mc.player.setTilePos(x, y, z)
```

Sets the position of the player to be standing on the specified coordinates.

Example: `mc.player.setTilePos(0, 10, 0)`

---

```
mc.player.setting(SettingName, True/False)
```

Modifies a setting for the player. Available settings: "autojump".

Example: `mc.player.setting("autojump", False)`

## CAMERA FUNCTIONS

---

```
mc.camera.setNormal(PlayerID)
```

Sets the camera for the player back to normal.

Example: `mc.camera.setNormal(playerOne)`

---

```
mc.camera.setFollow(playerID)
```

Sets the camera to be third-person for a player with the given ID.

Example: `mc.camera.setFollow(playerOne)`

---

```
mc.camera.setPos(x, y, z)
```

Causes the camera to look at the specified coordinates.

Example: `mc.camera.setPos(0, 10, 0)`

## WORLD FUNCTIONS

---

`mc.saveCheckpoint()`

Saves the world, which can be reverted to using `restoreCheckpoint()`.

---

`mc.restoreCheckpoint()`

Restores the world to the last checkpoint.

---

`mc.postToChat(message)`

Posts a message to the game chat.

Example: `mc.postToChat("Hello There")`

---

`mc.setting(SettingName, True/False)`

Modifies a setting for the world. Available settings: "world\_immutable", "nametags\_visible"

Example: `mc.setting("world_immutable", True)`

---

`mc.getHeight(x, z)`

Returns the Y coordinate of the highest non-air block at the specified x and z coordinates.

Example: `mc.getHeight(0, 0)`

## EVENT FUNCTIONS

---

`mc.events.pollBlockHits()`

Returns a list of new events where a block was clicked on with a sword:

```
blockEvents = mc.events.pollBlockHits()
for blockEvent in blockEvents:
    print blockEvent
```

---

`mc.events.clearAll()`

Clears all old events.