Minecraft API

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

Minecraft World

Create connection to Minecraft ([address, port]) => Minecraft object

```
mc = Minecraft.create([address = "localhost", port = 4711])
```

Retrieve the block type at location x, y, z => int.

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

Retrieve a cuboid of blocks at x0, y0, z0, x1, y1, z1 => [id:int].

```
blocks = mc.getBlocks(x0,y0,z0,x1,y1,z1)
for block in blocks:
    print block
```

Retrieve the block at location x, y, z => BlockObj.

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

Set the block at location x, y, z

```
mc.setBlock(x,y,z, id, [data])
```

Set a cuboid of blocks (x0,y0,z0,x1,y1,z1,id,[data])

```
mc.setBlocks(x0, y0, z0, x1, y1, z1, blockType, [blockData])
```

Get the height of the world (x,z) => int

```
y = mc.getHeight(x,z)
```

Get the entity ids of the connected players => [id:int]

```
# Get the entity id's of the players connected to the game.
entityIds = mc.getPlayerEntityIds()
for entityId in entityIds:
      print entityId
```

Save a checkpoint that can be used for restoring the world

```
mc.saveCheckpoint()
```

Restore the world state to the checkpoint

```
mc.restoreCheckpoint()
```

Post a message to the game chat

```
mc.postToChat(“message”)
```

Set a world setting (key, True/False). keys: world_immutable, nametags_visible

```
mc.setting(setting, status)
```

Minecraft Block

The definition of a Block, used to describe a block type and (if applicable) its data.

```
# Create block of a specific type.
blockObj = block.Block(id)
# Create a block of a specific type and apply a data value.
blockObj = block.Block(id, data)
```

Minecraft Player / Entity

Gets the player's or entity's position as a Vec3 of floats (decimal numbers)

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

Moves the player or entity to a position by passing co-ordinates ([x,y,z])

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

Gets the position of the 'tile' the player or entity is currently on

```
x, y, z = mc.player.getTilePos()
x, y, z = mc.entity.getTilePos(entityId)
```

Move the player or entity to a tile position by passing co-ordinates ([x,y,z])

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

Set a player setting (setting, status). keys: autojump

```
mc.player.setting(setting, status)
```

Minecraft Camera

Set camera mode to normal or fixed Minecraft view ([entityId])

```
mc.camera.setNormal(entityId)
mc.camera.setFixed()
```

Set camera mode to follow an entity ([entityId])

```
mc.camera.setFollow(entityId)
```

Set camera entity position (x,y,z)

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

Minecraft Events

Block Hits (Triggered by sword) => [BlockEvent]

```
# Get block event that have occured since the last time
blockEvents = mc.events.pollBlockHits()
for blockEvent in blockEvents:
      print blockEvent
```

Clear all old events

```
mc.events.clearAll()
```

Type of block event; only one event is currently implemented: BlockEvent.HIT

```
blockEvent.type         # (0: BlockEvent.HIT)
```

The position of the block where the event occured, i.e. the block which was hit, returns the x,y,z co-ordinates

```
x, y, z = blockEvent.pos
```

The face of the block where the event occurred

```
face = blockEvent.face
```

EntityId of the player who caused the event, i.e. the player who hit the block

```
blockEvent.entityId
```