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]

```
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
```

Minecraft Blocks

Block Type	Id	Block Type	Id
AIR	0	BRICK_BLOCK	45
STONE	1	TNT (*)	46
GRASS	2	BOOKSHELF	47
DIRT	3	MOSS_STONE	48
COBBLESTONE	4	OBSIDIAN	49
WOOD_PLANKS	5	TORCH (*)	50
SAPLING (*)	6	FIRE	51
BEDROCK	7	STAIRS_WOOD (*)	53
WATER / WATER_FLOWING	8	CHEST (*)	54
WATER_STATIONARY	9	DIAMOND_ORE	56
LAVA / LAVA_FLOWING	10	DIAMOND_BLOCK	57
LAVA_STATIONARY	11	CRAFTING_TABLE	58
SAND	12	FARMLAND	60
GRAVEL	13	FURNACE_INACTIVE (*)	61
GOLD_ORE	14	FURNACE_ACTIVE (*)	62
IRON_ORE	15	DOOR_WOOD	64
COAL_ORE	16	LADDER (*)	65
WOOD (*)	17	STAIRS_COBBLESTONE (*)	67
LEAVES (*)	18	DOOR_IRON	71
GLASS	20	REDSTONE_ORE	73
LAPIS_LAZULI_ORE	21	SNOW	78
LAPIS_LAZULI_BLOCK	22	ICE	79
SANDSTONE (*)	24	SNOW_BLOCK	80
BED (*)	26	CACTUS	81
COBWEB	30	CLAY	82
GRASS_TALL (*)	31	SUGAR_CANE	83
WOOL (*)	35	FENCE	85
FLOWER_YELLOW	37	GLOWSTONE_BLOCK	89
FLOWER_CYAN	38	BEDROCK_INVISIBLE	95
MUSHROOM_BROWN	39	STONE_BRICK (*)	98
MUSHROOM_RED	40	GLASS_PANE	102
GOLD_BLOCK	41	MELON	103
IRON_BLOCK	42	FENCE_GATE	107
STONE_SLAB_DOUBLE (*)	43	GLOWING_OBSIDIAN	246
STONE_SLAB (*)	44	NETHER_REACTOR_CORE (*)	247

WOOL	Data
White	0
Orange	1
Magenta	2
Light Blue	3
Yellow	4
Lime	5
Pink	6
Grey	7
Light Grey	8
Cyan	9
Purple	10
Blue	11
Brown	12
Green	13
Red	14
Black	15

WOOD / SAPLING / LEAVES	Data
Oak	0
Spruce	1
Birch	2

GRASS	
Shrub	0
Tall Grass	1
Fern	2

TORCH	
Standing on Floor	0
Pointing East	1
Pointing West	2
Pointing North	3
Pointing South	4

STONE_BRICK	
Stone Brick	0
Mossy Stone Brick	1
Cracked Stone Brick	2
Chiselled Stone Brick	3

STONE_SLAB / STONE_SLAB_DOUBLE	
Stone	0
Sandstone	1
Wooden	2
Cobblestone	3
Brick	4
Stone Brick	5

TNT	
Inactive	0
Ready to Explode	1

STAIRS_[COBBLESTONE, WOOD]	
Ascending East	0
Ascending West	1
Ascending North	2
Ascending South	3
Ascending East (Upside Down)	4
Ascending West (Upside Down)	5
Ascending North (Upside Down)	6
Ascending South (Upside Down)	7

SANDSTONE	
Sandstone	0
Chiselled Sandstone	1
Smooth Sandstone	2

NETHER_REACTOR_CORE	
Unused	0
Active	1
Stopped / Used Up	2

LADDERS, CHESTS, FURNACES	
Facing North	2
Facing South	3
Facing West	4
Facing East	5

Bed			
Head Facing South	0	Head Facing East	3
Head Facing West	1	Bed Occupied	+4
Head Facing North	2	Head of Bed	+8