The MagPi ESSENTIALS

[CHAPTER FOURTEEN]

MINECRAFT: PI EDITION API REFERENCE

Learn all the API functions of Minecraft: Pi from the mcpi Python library…

MINECRAFT

This is the main class and is how you create a connection to Minecraft and access the API.

.create(address = "localhost", port = 4711)

Create a connection to Minecraft => connection : Minecraft().

```
# use default address and port
mc = minecraft.Minecraft.create()
# specify ip address and port
mc = minecraft.Minecraft.create("192.168.1.1", 4711)
```

.getBlock(x,y,z)

Get a block => block id : int.

```
# retrieves the block type for the block at 0,0,0
blockType = mc.getBlock(0,0,0)
```

.getBlockWithData(x,y,z)

Get a block with data => block data : Block().

```
# retrieves a block object for the block at 0,0,0
blockObj = mc.getBlockWithData(0,0,0)
```

.setBlock(x,y,z)

Set a block.

```
# sets a block at an x,y,z coordinate to a particular type
mc.setBlock(0,0,0,block.DIRT.id)
# sets a block to a particular type and 'subtype'
mc.setblock(0,0,0,block.WOOL.id, 1)
```

.setBlocks(x0, y0, z0, x1, y1, z1, blockType, blockData)

Set a cuboid of blocks.

```
# sets many blocks at a time, filling the gap between 2
sets of x,y,z coordinates
mc.setBlocks(-1, -1, -1, 1, 1, 1, block.STONE.id)
```

.getHeight(x,z)

Get the height of the world => int.

```
# find the y (vertical) of an x,z coordinate which
represents the 'highest' (non-air) block
y = mc.getHeight(0,0)
```

.getPlayerEntityIds()

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

```
entityIds = mc.getPlayerEntityIds()
for entityId in entityIds:
    print entityId
```

.saveCheckpoint()

Save a checkpoint that can be used for restoring the world.

```
mc.saveCheckpoint()
```

.restoreCheckpoint()

Restore the world state to the checkpoint.

```
mc.restoreCheckpoint()
```

.postToChat(message)

Post a message to the game chat.

```
# write 'Hello Minecraft World' to the chat window
mc.postToChat("Hello Minecraft World")
```

.setting(setting, status)

Set a world setting.

```
# change world immutable to True
mc.setting("world_immutable", True)
# change name tags visible setting to False
mc.setting("nametags_visible", False)
```

MINECRAFT.PLAYER

The player class allows you to interact with the main player in the game. In a multiplayer game, this is the player who is hosting the game.

.getPos()

Gets the player's position in the world as an x, y, z of floats (decimal numbers) i.e. if the player is in the middle of a block, x.5 is returned.

```
# get player's position as floats
playerPos = mc.player.getPos()
```

.setPos(x,y,z)

Moves the player to a position in the world by passing x, y, z coordinates.

```
# set the player's position as floats
mc.player.setPos(0.0,0.0,0.0)
```

.getTilePos()

Gets the position of the 'tile' the player is currently on as an x, y, z of integers (whole numbers).

```
# get the position of the tile the player is on
playerTile = mc.player.getTilePos()
```

.setTilePos(x,y,z)

Move the player to a tile position in the world by passing x, y, z coordinates.

```
# set the position of the tile the player is on
mc.player.setTilePos(0,0,0)
```

.setting(setting, status)

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

```
# change the autojump setting to True
mc.player.setting("autojump", True)
```

MINECRAFT.ENTITY

The entity class allows you to interact with other players in the game, not just the main player, and is useful when creating programs for multiplayer games.

The entity functions should be used in conjunction with the **Minecraft.getPlayerEntityIds()** function.

```
entityIds = mc.getPlayerEntityIds()

player1EntityId = entityIds[0]

player2EntityId = entityIds[1]
```

.getPos(entityId)

Gets an entity's position in the world as an x, y, z of floats (decimal numbers) i.e. if the entity is in the middle of a block, x.5 is returned.

```
# get first entity position as floats
entityPos = mc.entity.getPos(entityId)
```

.setPos(entityId,x,y,z)

Moves the entity to a position in the world by passing x, y, z coordinates.

```
# set the entity's position as floats
mc.player.setPos(entityId,0.0,0.0,0.0)
```

.getTilePos(entityId)

Gets the position of the 'tile' the entity is currently on.

```
# get the position of the tile the entity is on
entityTile = mc.entity.getTilePos(entityId)
```

.setTilePos(entityId, x,y,z)

Move the entity to a tile position in the world by passing x, y, z coordinates.

```
# set the position of the tile the entity is on
mc.player.setTilePos(entityId,0,0,0)
```

MINECRAFT. CAMERA

The camera class allows you to modify the view that the player sees.

.setNormal(entityId)

Set camera mode to normal Minecraft view.

```
# set camera mode to normal for a specific player
mc.camera.setNormal(entityId)
```

.setFixed()

Set camera mode to fixed view.

```
# set camera mode to fixed
mc.camera.setFixed()
```

.setFollow(entityId)

Set camera mode to follow a player.

```
# set camera mode to follow a specific player
mc.camera.setFollow(entityId)
```

.setPos(x,y,z)

Set the camera position which will point down.

```
# set camera position to a specific position of x, y, z
mc.camera.setPos(0,0,0)
```

MINECRAFT.EVENTS

.pollBlockHits()

Gets block hits, triggered by right-clicking with a sword, since the last time the function was run => BlockEvent().

```
# get block event hits
blockEvents = mc.events.pollBlockHits()
for blockEvent in blockEvents:
    print blockEvent
```

.clearAll()

Clear all old events.

```
# clear all events that have happened since the last
.pollBlockHits call
mc.events.clearAll()
```

BLOCKEVENT

The BlockEvent class is how you get information about block hit events which have been returned by the Minecraft **events.pollBlockHits()** function.

```
blockEventType = blockEvent.type
```

.type

Type of block event. There is only 1 event currently implemented – BlockEvent.HIT = 0.

```
blockEventType = blockEvent.type
```

.pos

The position of the block which was hit as x, y, z coordinates.

```
blockEventPos = BlockEvent.pos
```

.face

The face of the block which was hit, as a number 0 – 6.

```
blockEventFace = BlockEvent.face
```

.entityId

The entity ID of the player who hit the block.

```
blockEventPlayer - BlockEvent.entityId
```

BLOCK

The block module provides constants which let you use blocks by their names rather than their IDs. Many blocks also have data types which let you change a block – e.g. the wool block has the ID 35 and the constant WOOL; a data ID between 0 – 16 changes the colour.

.id	Constant	.data	Sub-type
0	AIR	-	-
1	STONE	-	-
2	GRASS	-	-
3	DIRT	-	-
4	COBBLESTONE	-	-
5	WOOD_PLANKS	0	Oak
		1	Spruce
		2	Birch
		3	Jungle
6	SAPLING	0	Oak
		1	Spruce
		2	Birch
		3	Jungle
7	BEDROCK	-	-
8	WATER_FLOWING I WATER	-	-
9	WATER_STATIONARY	0	High
		7	Low
10	LAVA_FLOWING I LAVA	-	-
11	LAVA_STATIONARY	0	High
		7	Low
12	SAND	-	-
13	GRAVEL	-	-
14	GOLD_ORE	-	-
15	IRON_ORE	-	-
16	COAL_ORE	-	-
17	WOOD	0	Oak (up/down)

.id	Constant	.data	Sub-type
		1	Spruce (up/down)
		2	Birch (up/down)
		3	Jungle (up/down)
		4	Oak (east/west)
		5	Spruce (east/west)
		6	Birch (east/west)
		7	Jungle (east/west)
		8	Oak (north/south)
		9	Spruce (north/south)
		10	Birch (north/south)
		11	Jungle (north/south)
		12	Oak (only bark)
		13	Spruce (only bark)
		14	Birch (only bark)
		15	Jungle (only bark)
18	LEAVES	1	Oak leaves
		2	Spruce leaves
		3	Birch leaves
20	GLASS	-	-
21	LAPIS_LAZULI_ORE	-	-
22	LAPIS_LAZULI_BLOCK	-	-
24	SANDSTONE	0	Sandstone
		1	Chiselled Sandstone
		2	Smooth Sandstone
26	BED	-	-
30	COBWEB	-	-
31	GRASS_TALL	0	Shrub
		1	Grass
		2	Fern
		3	Grass (colour by biome)
35	WOOL	0	White
		1	Orange
		2	Magenta
		3	Light Blue
		4	Yellow

.id	Constant	.data	Sub-type
		5	Lime
		6	Pink
		7	Grey
		8	Light Grey
		9	Cyan
		10	Purple
		11	Blue
		12	Brown
		13	Green
		14	Red
		15	Black
37	FLOWER_YELLOW	-	-
38	FLOWER_CYAN	-	-
39	MUSHROOM_BROWN	-	-
40	MUSHROOM_RED	-	-
41	GOLD_BLOCK	-	-
42	IRON_BLOCK	-	-
43	STONE_SLAB_DOUBLE	**0**	Stone
		1	Sandstone
		2	Wooden
		3	Cobblestone
		4	Brick
		5	Stone Brick
		6	Nether Brick
		7	Quartz
44	STONE_SLAB	**0**	Stone
		1	Sandstone
		2	Wooden
		3	Cobblestone
		4	Brick
		5	Stone Brick
		6	Nether Brick
		7	Quartz
45	BRICK_BLOCK	-	-
46	TNT	**0**	Inactive

.id	Constant	.data	Sub-type
		1	Ready to explode
47	BOOKSHELF	-	-
48	MOSS_STONE	-	-
49	OBSIDIAN	-	-
50	TORCH	**0**	Standing on the floor
		1	Pointing east
		2	Pointing west
		3	Pointing south
		4	Pointing north
51	FIRE	-	-
53	STAIRS_WOOD	**0**	Ascending east
		1	Ascending west
		2	Ascending south
		3	Ascending north
		4	Ascending east (upside down)
		5	Ascending west (upside down)
		6	Ascending south (upside down)
		7	Ascending north (upside down)
54	CHEST	**2**	Facing north
		3	Facing south
		4	Facing west
		5	Facing east
56	DIAMOND_ORE	-	-
57	DIAMOND_BLOCK	-	-
58	CRAFTING_TABLE	-	-
60	FARMLAND	-	-
61	FURNACE_INACTIVE	**2**	Facing north
		3	Facing south
		4	Facing west
		5	Facing east
62	FURNACE_ACTIVE	**2**	Facing north
		3	Facing south
		4	Facing west
		5	Facing east
64	DOOR_WOOD	-	-

.id	Constant	.data	Sub-type
65	LADDER	**2**	Facing north
		3	Facing south
		4	Facing west
		5	Facing east
67	STAIRS_COBBLESTONE	**0**	Ascending east
		1	Ascending west
		2	Ascending south
		3	Ascending north
		4	Ascending east (upside down)
		5	Ascending west (upside down)
		6	Ascending south (upside down)
		7	Ascending north (upside down)
71	DOOR_IRON	-	-
73	REDSTONE_ORE	-	-
78	SNOW	**0**	Lowest
		7	Highest
79	ICE	-	-
80	SNOW_BLOCK	-	-
81	CACTUS	-	-
82	CLAY	-	-
83	SUGAR_CANE	-	-
85	FENCE	-	-
89	GLOWSTONE_BLOCK	-	-
95	BEDROCK_INVISIBLE	-	-
98	STONE_BRICK	**0**	Stone brick
		1	Mossy stone brick
		2	Cracked stone brick
		3	Chiseled stone brick
102	GLASS_PANE	-	-
103	MELON	-	-
107	FENCE_GATE	-	-
246	GLOWING_OBSIDIAN	-	-
247	NETHER_REACTOR	**0**	Unused
		1	Active
		2	Stopped / used up