Minecraft Programming Cheat Sheet

This is for the Bluefruit taster sessions, learning programming via procedural architecture in Minecraft using the Raspberry Pi. This is a quick cheat sheet of all the functions and everything needed for the workshop.

basics

from bluefruit_minecraft import *

You need to put this at the top of your script to tell Python where to find all the Bluefruit and Minecraft commands we are using.

debug()

example: `debug("hello")`

Sends "hello" to the Minecraft chat. Useful for debugging your programs.

bulldoze()

Flattens a large area in the middle of the world for you to work on. A good idea to call this all the time from the top of your program, so the world is cleared before you build things.

points

Everything in Minecraft is three dimensional, so we need to use 3 numbers to specify locations and sizes of things. For positioning, it's useful to look at the X,Y,Z coordinates at the top left of the Minecraft window.

`mypoint = point(10,0,2)` : makes a new point

The values of the x y and z can be retrieved with `mypoint.x` , `mypoint.y` or `mypoint.z` .

Points can be added or subtracted, for example: `newpoint = mypoint + point(1,2,3)`

Will result in newpoint containing x=11 y=2 z=5

```
distance(point_a,point_b)
```

Returns the distance between two points.

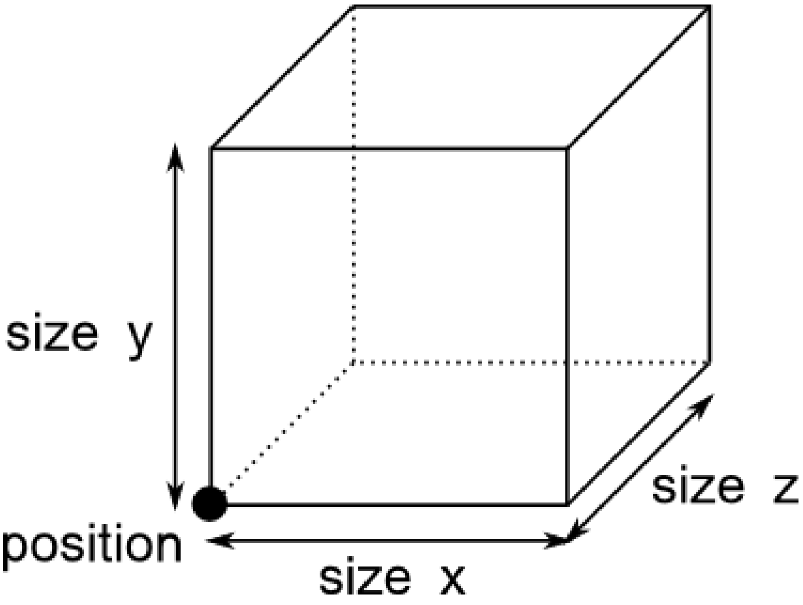
primitives

Primitives are simple shapes you can use to create more complex ones. They all take a block type, if this is set to AIR then the shape will 'eat into' other shapes made previously.

box(blocktype, position_point, size_point)

example: `box(CLAY, point(0,0,0), point(10,10,10))`

Will create a 10x10x10 block of clay in the middle of the world (0,0,0)

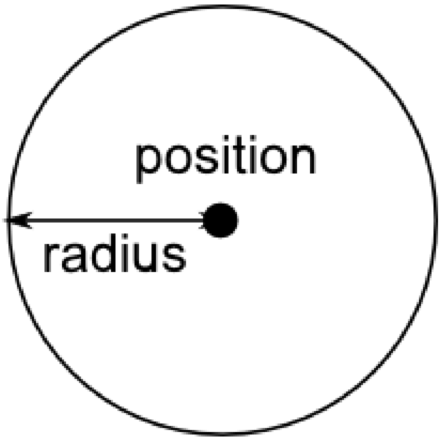


A box image

sphere(blocktype, centre_point, radius)

example: `sphere(MELON, point(0,10,0), 10)`

Will create a sphere of melon slightly above the centre of the world with a radius of 10 blocks.

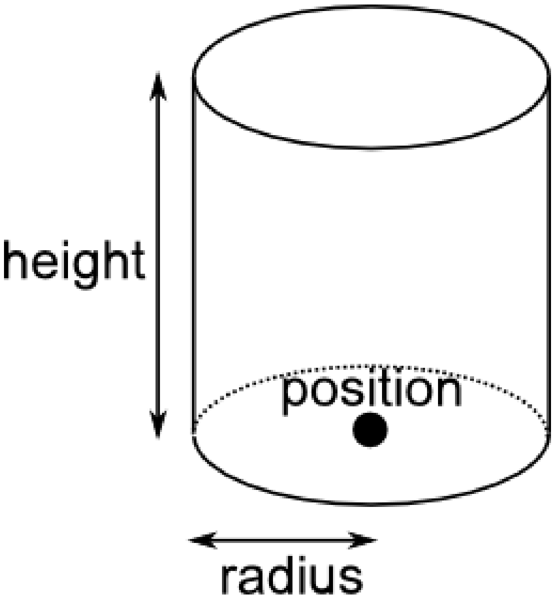


A sphere image

cylinder(blocktype, position_point, radius, height):

example: `cylinder(STONE_BRICK,point(0,0,0),6,20)`

Builds a cylinder of stone brick in the centre of the world radius 6, height 20.

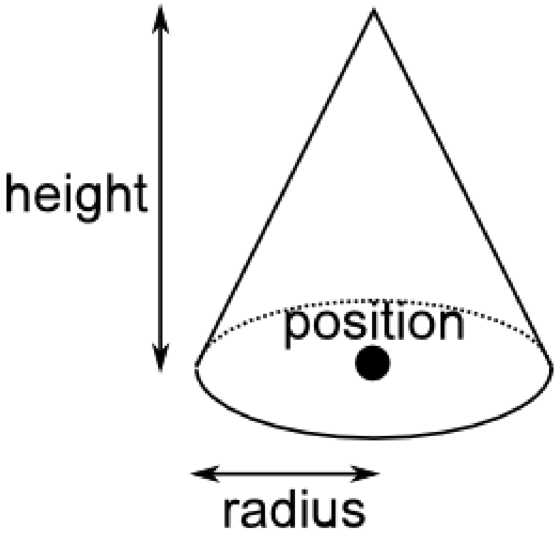


A cylinder image

cone(blocktype, position_point, radius, height):

example: `cone(WOOD,point(0,0,0),6,20)`

Same as cylinder, but with a sharp point at the top.

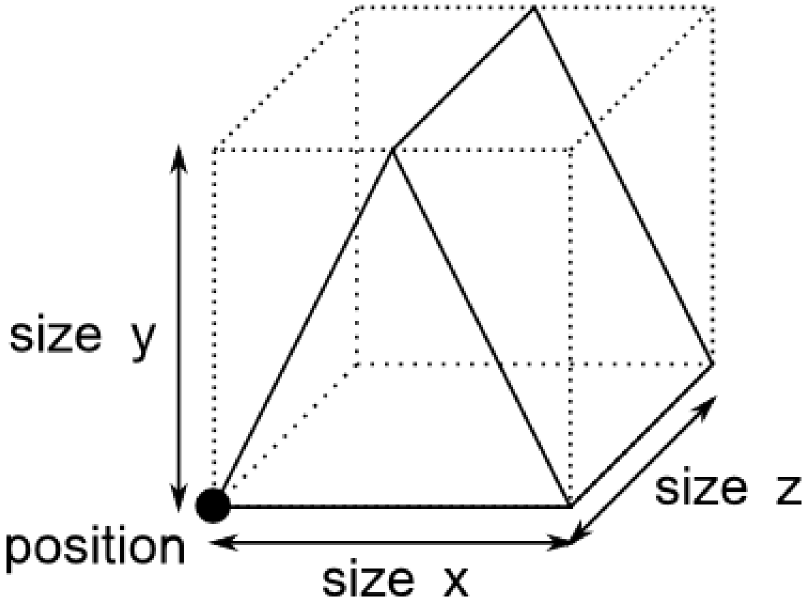


A cone image

toblerone(blocktype, position_point, size_point)

example: `toblerone(GLASS,point(0,0,0),point(10,10,3))`

Makes a 'toblerone', referred to by lesser mortals as a prism. Useful for roof building.



A toblerone image

player info

my_pos()

Returns the player position point

move_me_to(position_point)

Teleport the player somewhere

i_am_lost()

Quick way to get back to the centre of the world (spawn point)

randomness

random_range(from, to)

example: `random_range(0, 10)`

Returns a random number between from and to.

choose_one(list, of, things)

example: `choose_one(STONE, GRASS, DIRT, BEDROCK)`

Returns a random choice of the things given.

random_point(from, to)

example: `random_point(point(0,0,0), point(10,10,10))`

Returns a random point inside the box you specify with the two points, different every time.

functions

You can make your own functions from these simple ones. This is the essence of programming, as you can break problems down into simpler ones. We use 'def' to create a new function:

```
def hollow_cylinder(blocktype, position, inner_radius, outer_radius, length):
    cylinder(blocktype, position, outer_radius, length)
    cylinder(AIR, position, inner_radius, length)`
```

Will make a function to create a hollow cylinder (by building one then cutting the inner one out) which you can then use like so:

```
hollow_cylinder(CLAY, point(0,0,0), 4, 6, 10)
```

looping

We use 'for' for looping:

```
for i in range(0, 10):
    cube(GOLD_BLOCK, point(i*10, 0, 0), point(5, 5, 5))'
```

Will make a row of gold cubes.

block types

All the block type listed, there may well be more... be careful with lava.

AIR, STONE, GRASS, DIRT, COBBLESTONE, WOOD_PLANKS, SAPLING, BEDROCK, WATER_FLOWING, WATER, WATER_STATIONARY, LAVA_FLOWING, LAVA, LAVA_STATIONARY, SAND, GRAVEL, GOLD_ORE, IRON_ORE, COAL_ORE, WOOD, LEAVES, GLASS, LAPIS_LAZULI_ORE, LAPIS_LAZULI_BLOCK, SANDSTONE, BED, COBWEB, GRASS_TALL, WOOL, FLOWER_YELLOW, FLOWER_CYAN, MUSHROOM_BROWN, MUSHROOM_RED, GOLD_BLOCK, IRON_BLOCK, STONE_SLAB_DOUBLE, STONE_SLAB, BRICK_BLOCK, TNT, BOOKSHELF, MOSS_STONE, OBSIDIAN, TORCH, FIRE, STAIRS_WOOD, CHEST, DIAMOND_ORE, DIAMOND_BLOCK, CRAFTING_TABLE, FARMLAND, FURNACE_INACTIVE, FURNACE_ACTIVE, DOOR_WOOD, LADDER, STAIRS_COBBLESTONE, DOOR_IRON, REDSTONE_ORE, SNOW, ICE, SNOW_BLOCK, CACTUS, CLAY, SUGAR_CANE, FENCE, GLOWSTONE_BLOCK, BEDROCK_INVISIBLE, STONE_BRICK, GLASS_PANE, MELON, FENCE_GATE, GLOWING_OBSIDIAN, NETHER_REACTOR_CORE