

# マイクラフト・プログラミング応用

## 第3回 ブロックの組み立て

- ☆Drone(ドローン)オブジェクトとは
- ☆Drone(ドローン)オブジェクトのメソッド
- ☆ビルを建ててみよう

2018.02.23

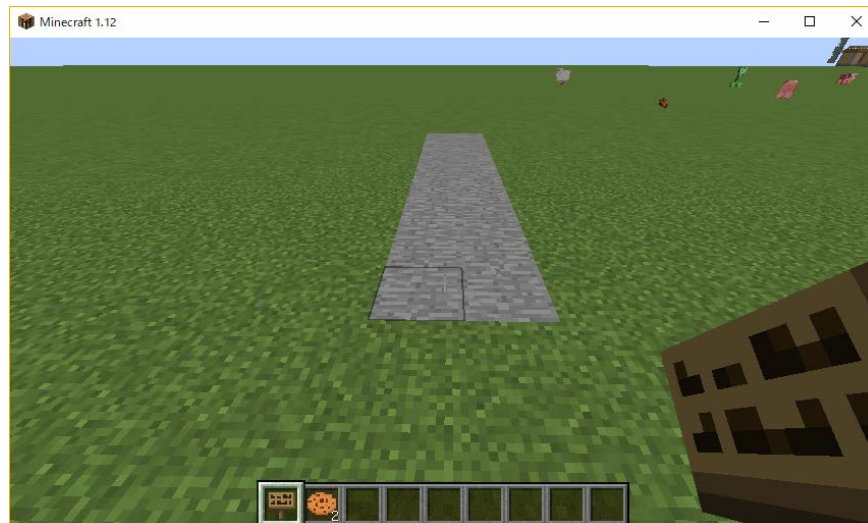
鎌倉シチズンネット(KCN)

# Droneオブジェクト (1)

- Drone(ドローン)オブジェクトとは

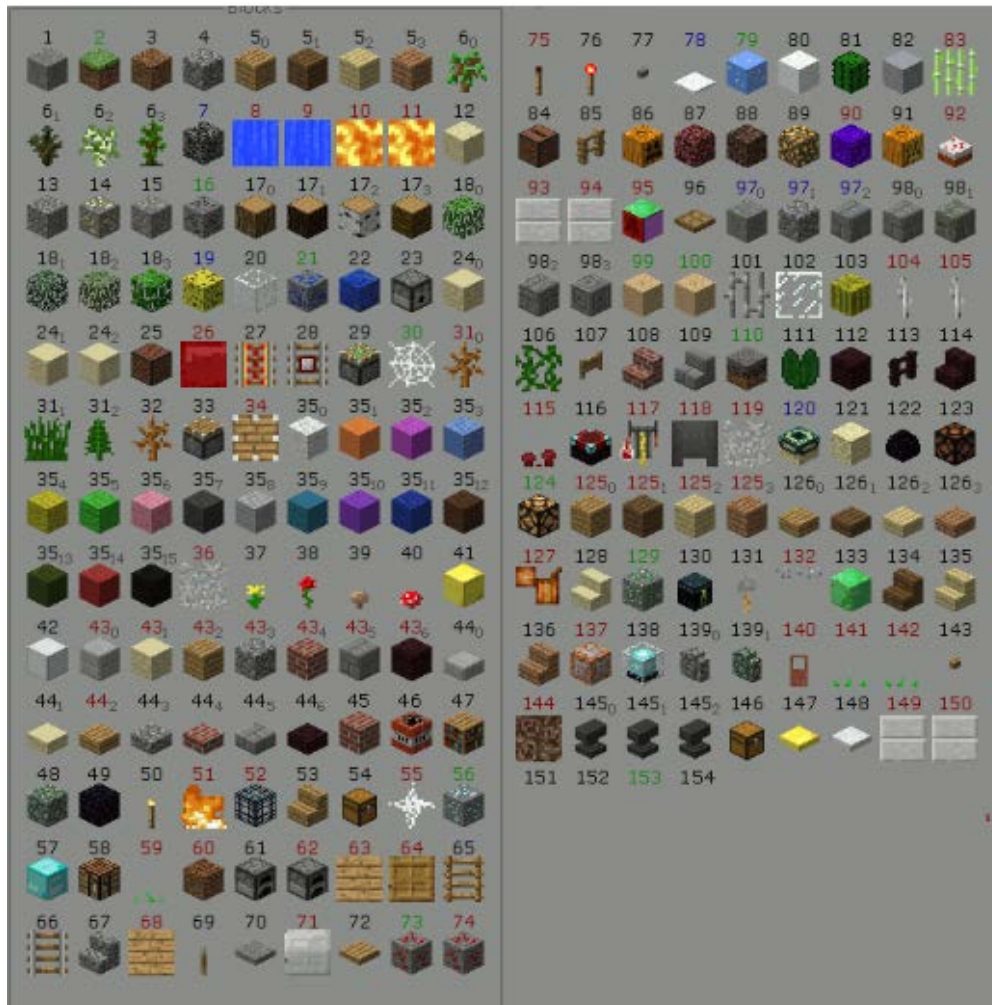
DroneオブジェクトはScriptCraft 固有のオブジェクトで、目には見えないが、みんなの代わりにあちこちに動き回ったり、ブロックを組み立ててくれるオブジェクト(目に見えないドローン)です。

- 1) ターゲットのブロックを選択します(ブロックの周囲に輪郭線が表示されます)
- 2) `/js box( blocks.stone, 2, 1, 10 )` と入力すると、ターゲットのブロックが、幅が2、高さが1、奥行が10の石のブロックになります。



# Droneオブジェクト (2)

- ブロックの種類(ブロック番号)



左図で5<sub>0</sub>、5<sub>1</sub>、5<sub>2</sub>など  
下付き数字があるもの  
は、それぞれ'5'、  
'5:1'、'5:2'などと指定  
する。

```
/js box('5')  
/js box('5:1')  
/js box('5:2')
```

# Droneオブジェクト (3)

- ブロックの種類(ブロック番号)

ブロックの種類(ID)を番号ではなく名前でも指定することもできます。

```
/js box( blocks.oak )      : オーク  
/js box( blocks.spruce )   : マツ  
/js box( blocks.birch )   : シラカバ
```

チャット欄に次のように入力してTABキーを押すと、「st」で始まるブロック名の一覧が表示されます。

```
/js blocks.st
```

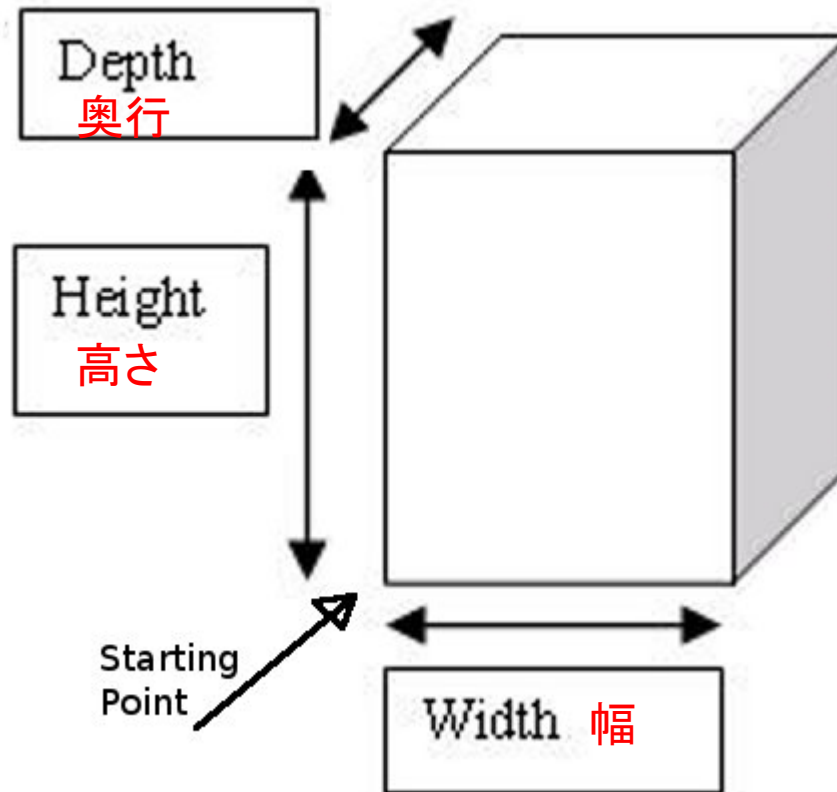
続けてTABキーを押すと、次の候補が表示されます。

☆ScriptCraft のmodules フォルダの中にある blocks.js を見ると、名前と番号の対応が分かります。

[ブロックIDとデータ値](#)

# Droneオブジェクト (4)

- ブロック(群)の幅、高さ、奥行



# Droneオブジェクト (5)

## ①ブロックの組み立て(主なもの)

メソッド(関数)	機能
box( block-id, 幅, 高さ, 奥行 )	直方体の作成
box0( block-id, 幅, 高さ, 奥行 )	// (空洞、床も天井もなし)
cylinder( block-id, 半径, 高さ )	円柱の作成(煙突に向く)
cylinder0( block-id, 半径, 高さ )	// (空洞、床も天井もなし)
prism( block-id, 幅, 奥行)	プリズムの作成(屋根に向く)
prism0( block-id, 幅, 奥行)	// (空洞、床も天井もなし)

# Droneオブジェクト (6)

## ②移動

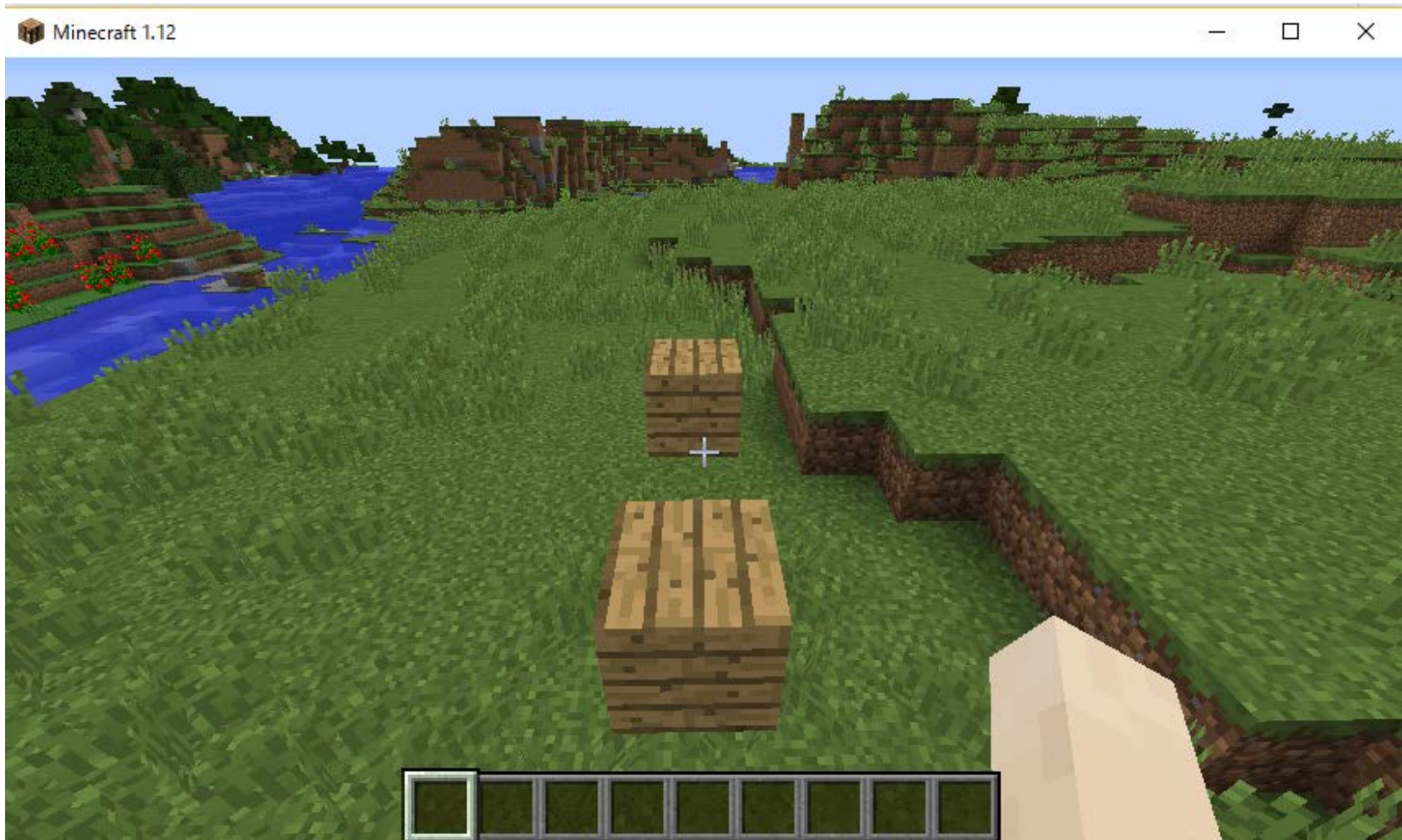
メソッド(関数)	機能
up ( ブロック数 )	上に移動
down ( ブロック数 )	下に移動
left ( ブロック数 )	左に移動
right ( ブロック数 )	右に移動
fwd ( ブロック数 )	プレイヤーから離れる方向に移動
back ( ブロック数 )	プレイヤーに近づく方向に移動
turn ( ターン数 )	時計回りに回転、turn( ) は90度回転、turn ( 2 ) は180度回転

## ③移動とブロックの組み立ての組み合わせ

```
/js up(1).box( blocks.oak ).fwd(3).box( blocks.oak )
```

チャット欄に上記のように入力すると、次頁の画面になります。

# Droneオブジェクト (7)





# Droneオブジェクト (8)

## ⑤その他のメソッド

メソッド(関数)	機能
chkpt ('名前')	現在の位置を記憶する
move ('名前')	記憶された位置に移動する move ('start')はDroneが生成されたときの位置に移動する。

### 【練習問題】

次頁の画面のような家を建ててみよう。

幅:7、高さ:3、奥行:6

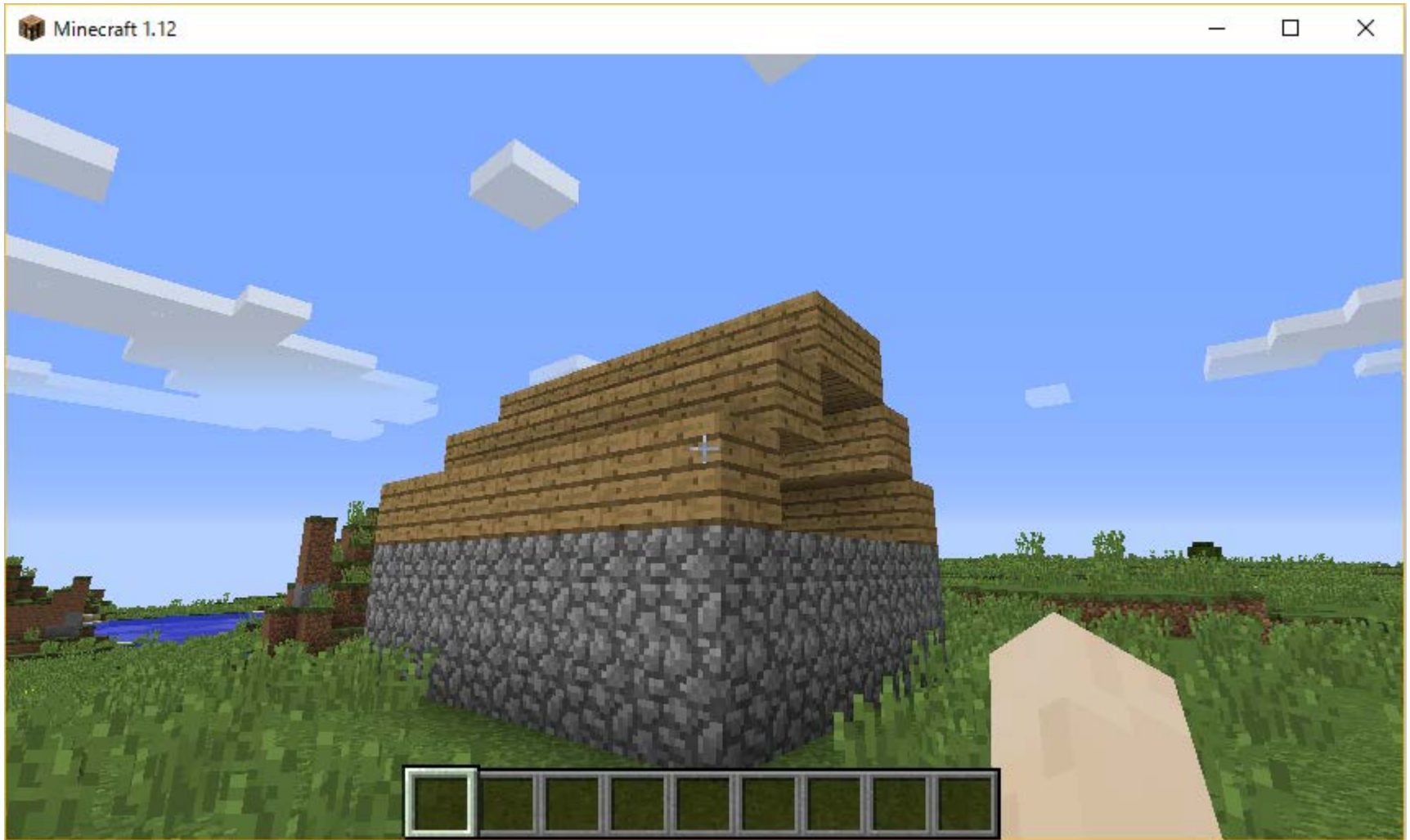
石:id = 4、木材:id=5

利用するメソッド: `box0( blocks.stone, 幅, 高さ, 奥行 )`

`up( 3 )`

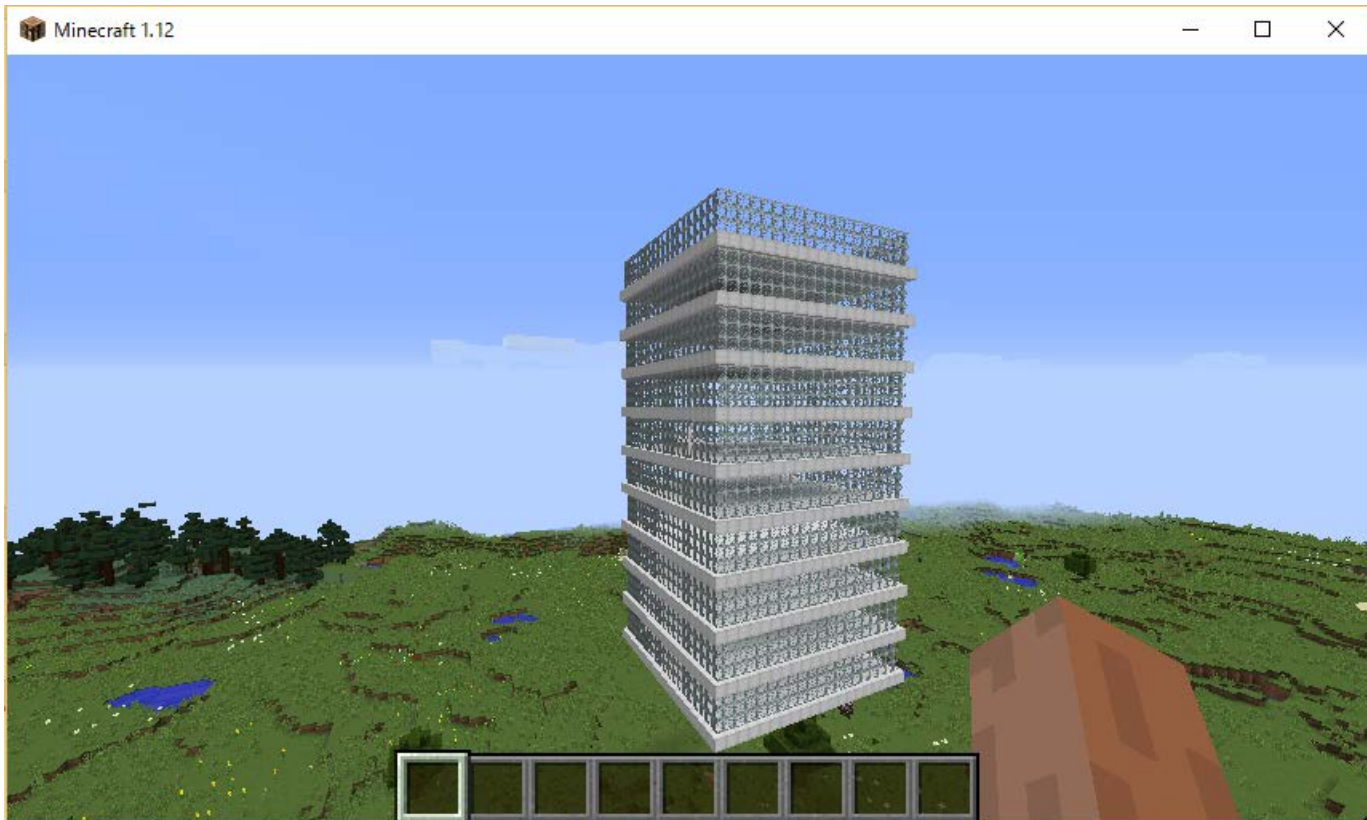
`prism0( blocks.oak, 幅, 奥行 )`

# Droneオブジェクト (9)



# ビルを建ててみよう (1)

- 次のようなビルを建ててみよう



# ビルを建ててみよう (2)

- Drone オブジェクトの生成方法

1. `var d = box ( blocks.oak );`

上記の記述は次のように書くのと同じです。

```
var d = new Drone( self ).box( blocks.oak );
```

2. `var d = new Drone ( self );`

上記の1と2では、Droneはプレイヤーの視線の先にブロックがあればその位置に、なければプレイヤーの位置の2ブロック前の位置に生成されます。

3. `var d = new Drone ( x, y, z, 方向, ワールド );`

4. `var d = new Drone ( location );`

例:

```
events.blockBreak( function( event ) {  
    var location = event.block.location;  
    var drone = new Drone(location);  
    // do more stuff with the drone here...  
});
```

(注) 方向 0 : 東、1 : 南、 2 : 西、3 : 北
---------------------------------------

# ビルを建ててみよう (3)

- mySkyscraper.js のコードです

```
exports.mySkyscraper = function( floors ){
```

```
function mySkyscraper( floors ) {  
  var i ;  
  var drone = new Drone( self );  
  if ( typeof floors == 'undefined' ) {  
    floors = 10;  
  }  
}
```

```
// bookmark the drone's position so it can return there later
```

```
drone.chkpt('myskyscraper');
```

```
for ( i = 0; i < floors; i++ ) {  
  drone  
  .box(blocks.iron,20,1,20)  
  .up()  
  .box0(blocks.glass_pane,20,3,20)  
  .up(3);  
}
```

```
// return the drone to where it started
```

```
drone.move('myskyscraper');
```

```
}  
mySkyscraper( floors );  
}
```

mySkyscraper.js をc:¥Users¥(名前)  
¥spigot¥scriptcraft¥plugins¥{Minecraft  
のユーザ名} に保存し、チャット欄に次の  
ように入力します。  
/js refresh()  
/js mySkyscraper(階数)

# ビルを建ててみよう (4)

## [ 練習問題1 ]

ビルのガラスの色を青くしてみよう。

```
.box0(blocks.stained_glass_pane.blue,20,3,20)
```

## [ 練習問題2 ]

ビルのガラスの色を階ごとに交互に変えてみよう。

```
.box0(blocks.stained_glass_pane.blue,20,3,20)
```

```
.box0(blocks.stained_glass_pane.green,20,3,20)
```

## [ 練習問題3 ]

ビルを建てるプログラムを参考にして、P.9の練習問題の家(P.10)を建てるプログラムを作成しよう。

```
/js myHouse() でプログラムを起動する
```

# 参考資料

- 参考資料
  - [Minecraft のブロックID](#)
  - [Minecraft のデータ値](#)