U-16プログラミングコンテスト
京都大会　事前講習会

2024年9月14日

同志社中学校・高等学校　知創館（ちそうかん）

プログラム

- 参加者自己紹介
- 「Chaser(チェイサー)」って何？
- Pythonを使ったプログラミングの基本
- Pythonで「Chaser」のキャラクターを動かす

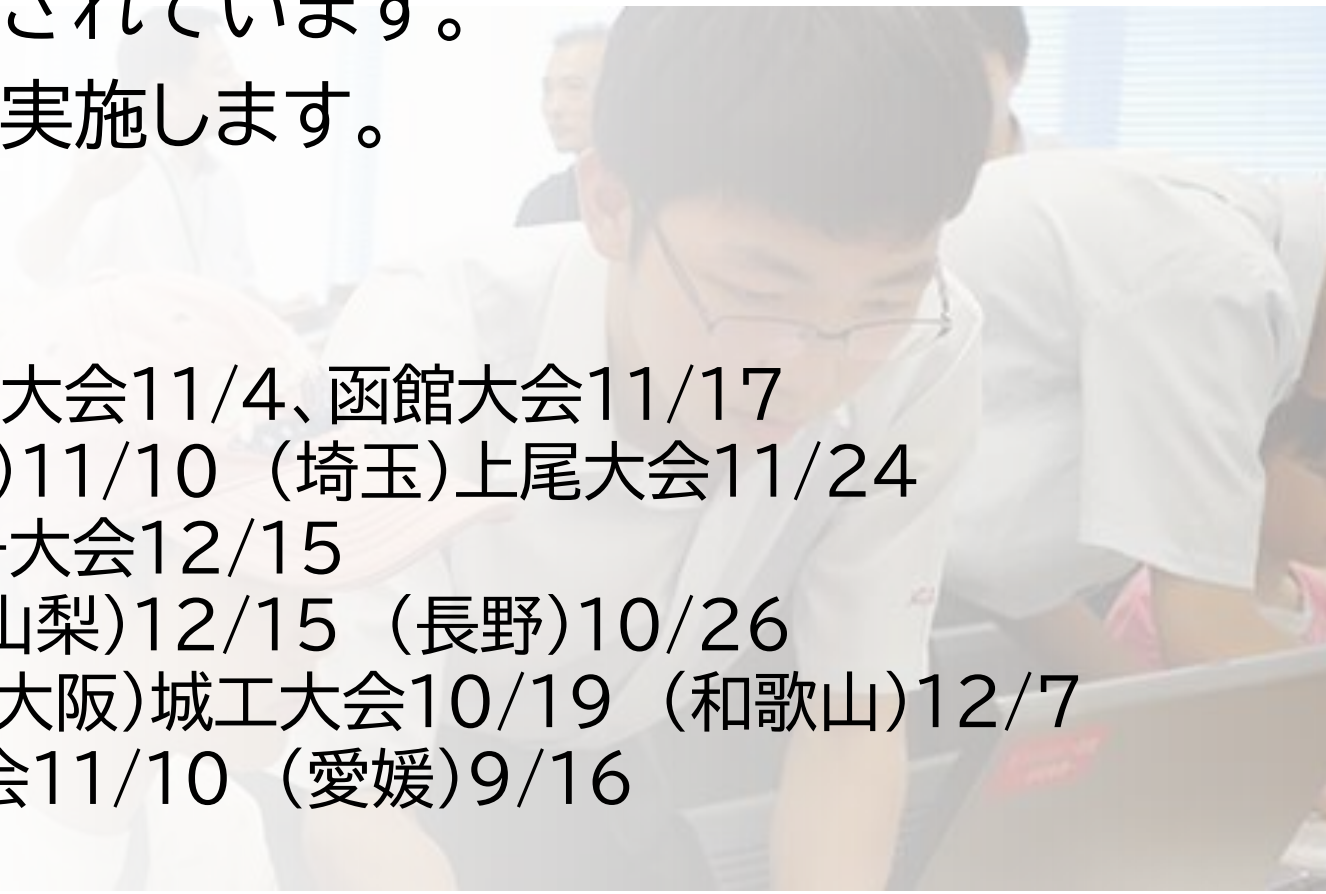
●参加者自己紹介

- 名前
- 学校名・学年
- 今日の講習会で知りたいこと・がんばりたいこと

●CHaser(チェイサー)って何？

U-16プログラミングコンテストとは？

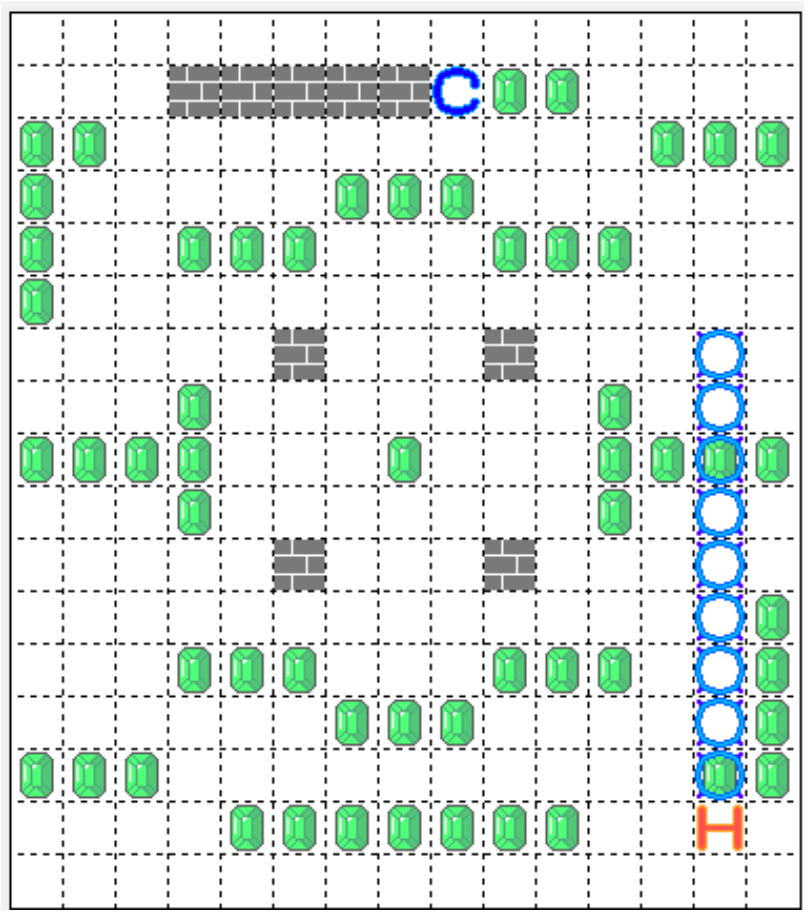
2011年に北海道旭川市ではじまった取り組みです。
いまは全国の様々な地域で開催されています。
2024年に京都で初めて大会を実施します。

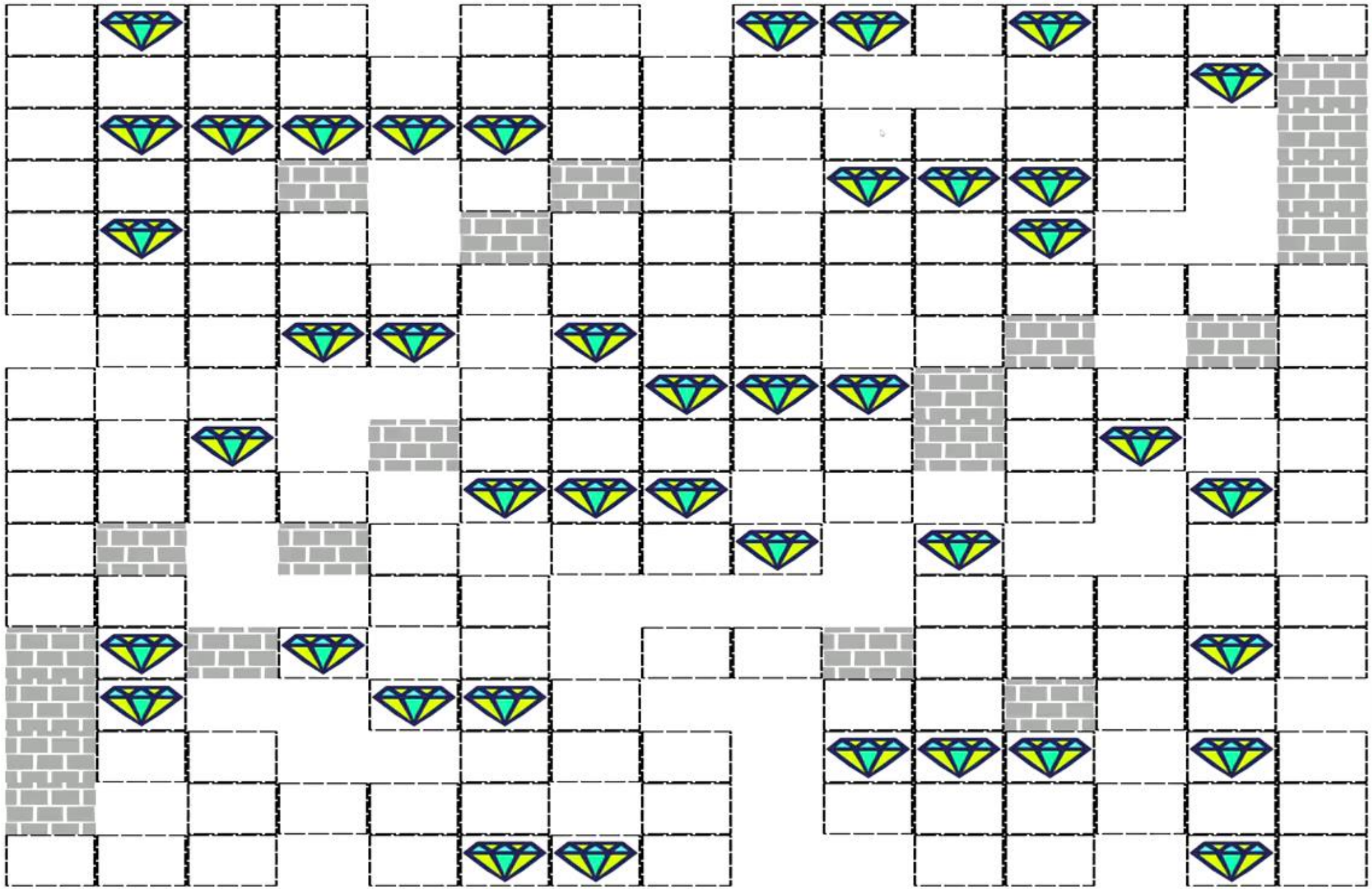


2024年度の大会
(北海道)旭川大会11/3、札幌大会11/4、函館大会11/17
(岩手)一関大会9/23 (栃木)11/10 (埼玉)上尾大会11/24
(東京)東京大会12/8、八王子大会12/15
(新潟)上越妙高大会11/3 (山梨)12/15 (長野)10/26
(三重)9/14 (京都)11/3 (大阪)城工大会10/19 (和歌山)12/7
(山口)11/23 (徳島)阿南大会11/10 (愛媛)9/16

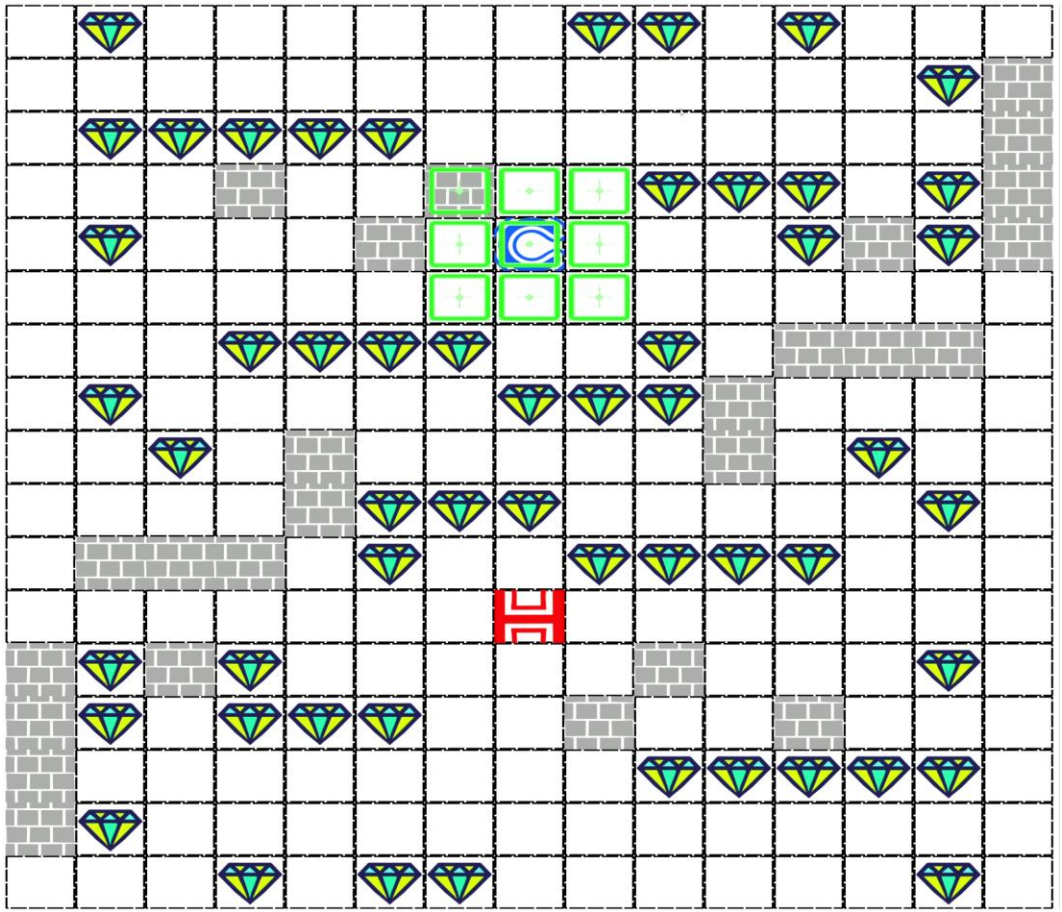
CHaser(チェイサー)とは

- U-16プログラミングコンテストで使われます
- 2人で対戦するゲームです
- Cool(先攻)とHot(後攻)に分かれて、
  取ったアイテムの数を競います。




Turn : 100
NAME : 旭川　下村
SCORE : 0
NAME : 附属中松野
SCORE : 0
残りアイテム数 : 54

CHaserのルール

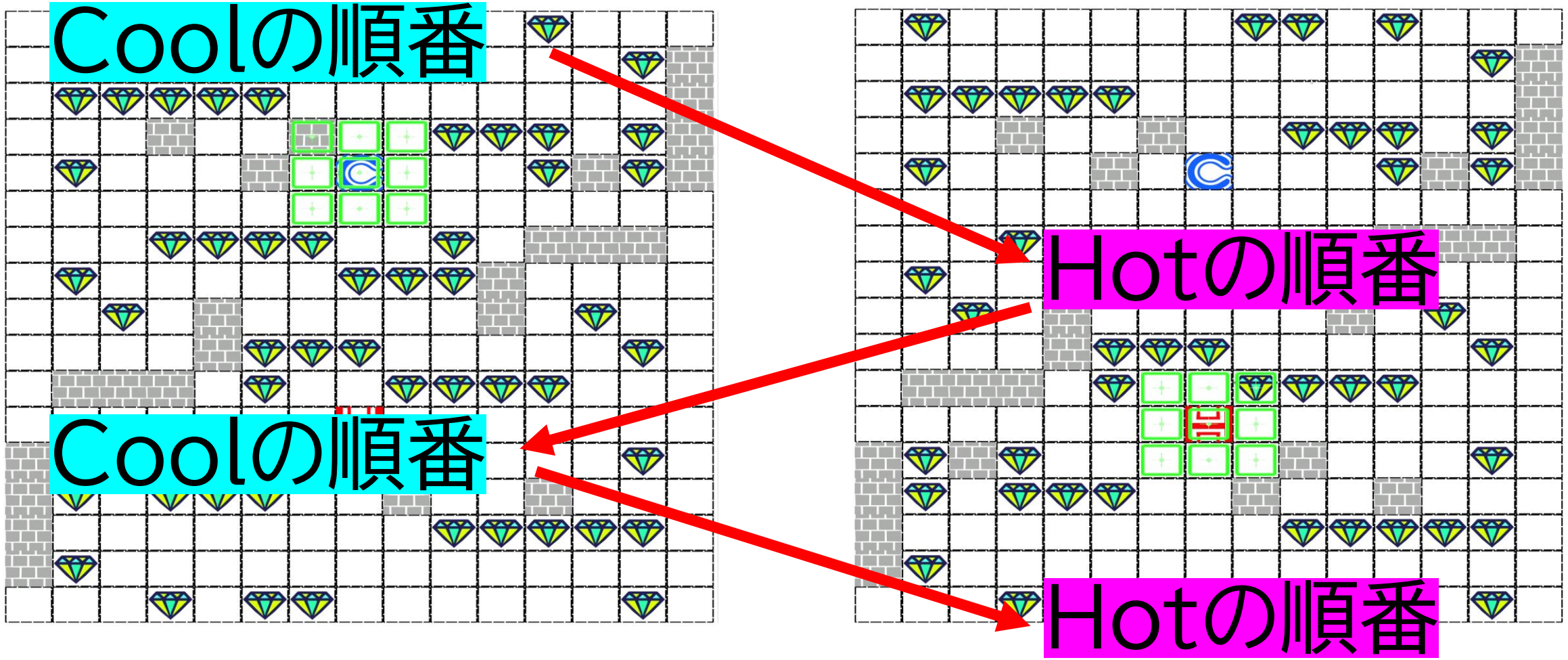


→「ターン制」
（将棋やチェスのようなもの）
CoolとHotが順番に命令を実行

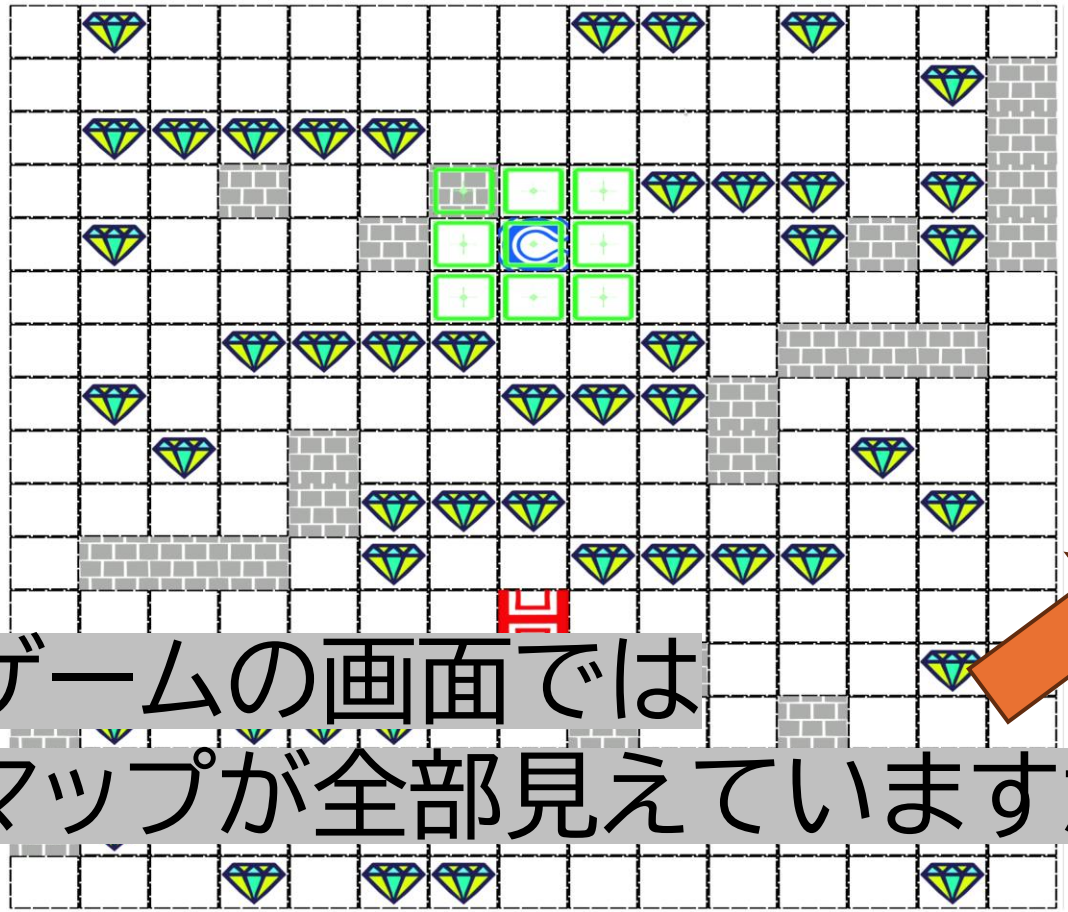
→自分のまわりしか見えない

→できること（命令）は4種類

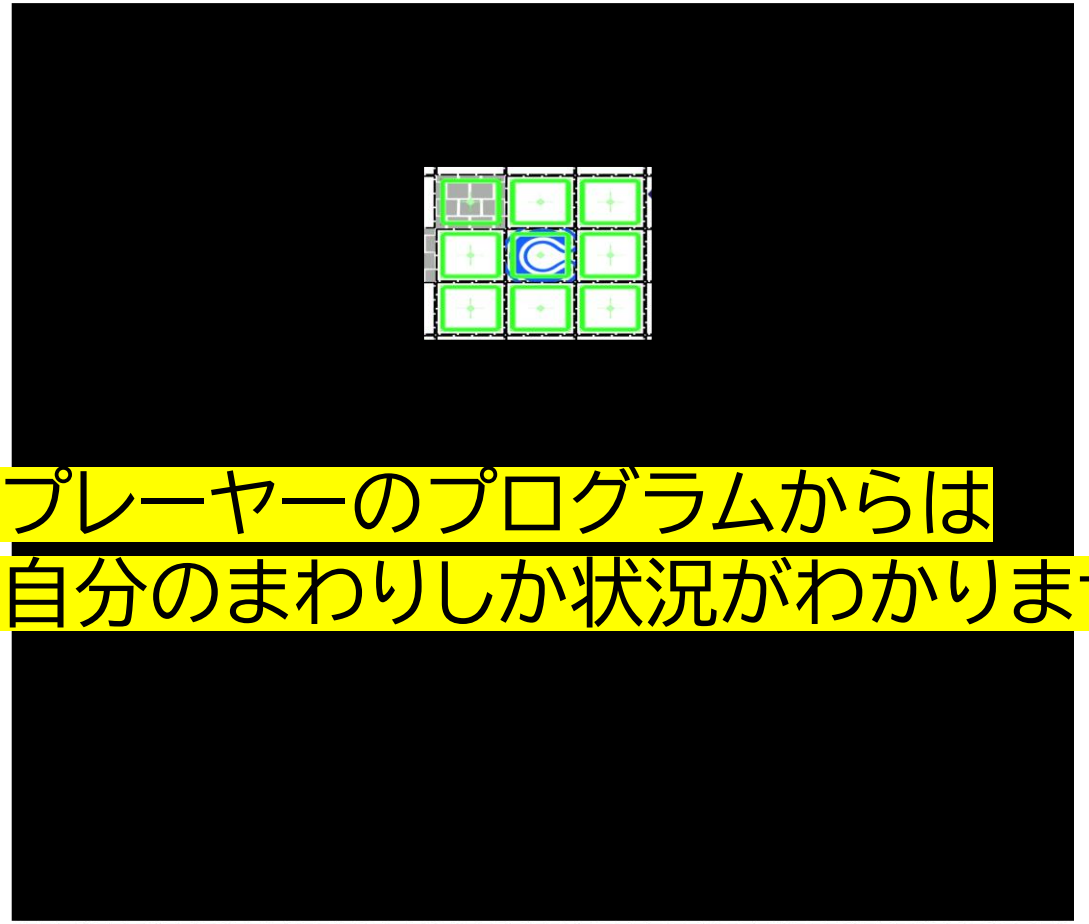
「ターン制」



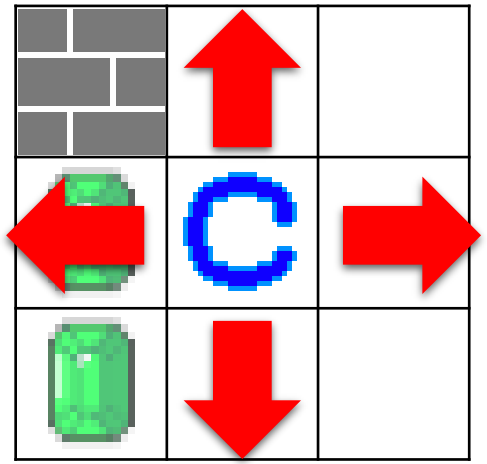
自分のまわりしか見えない



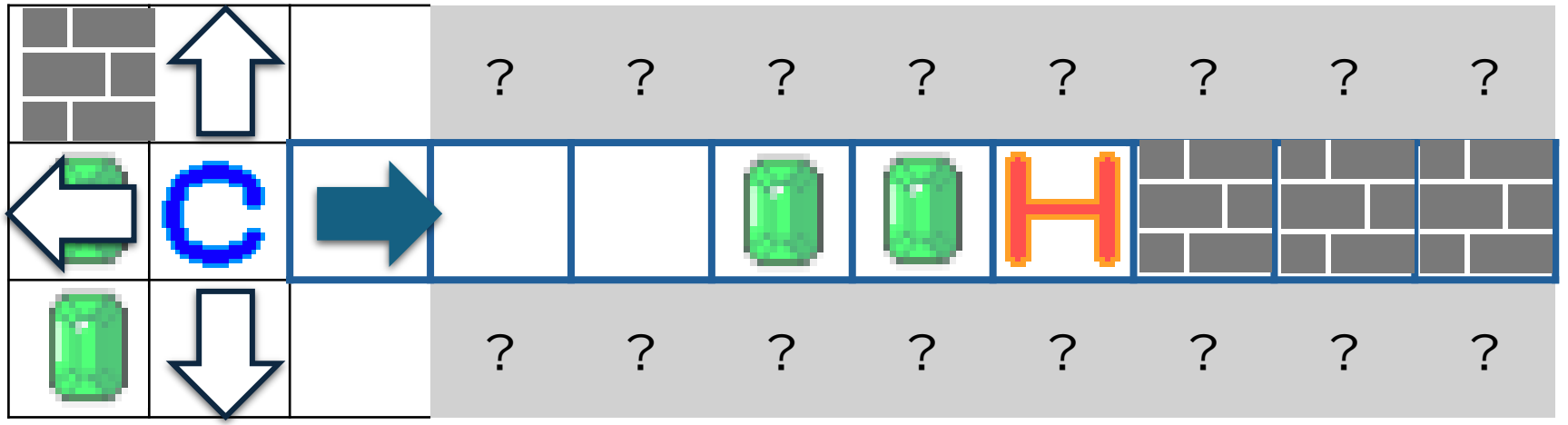
ゲームの画面では
マップが全部見えていますが



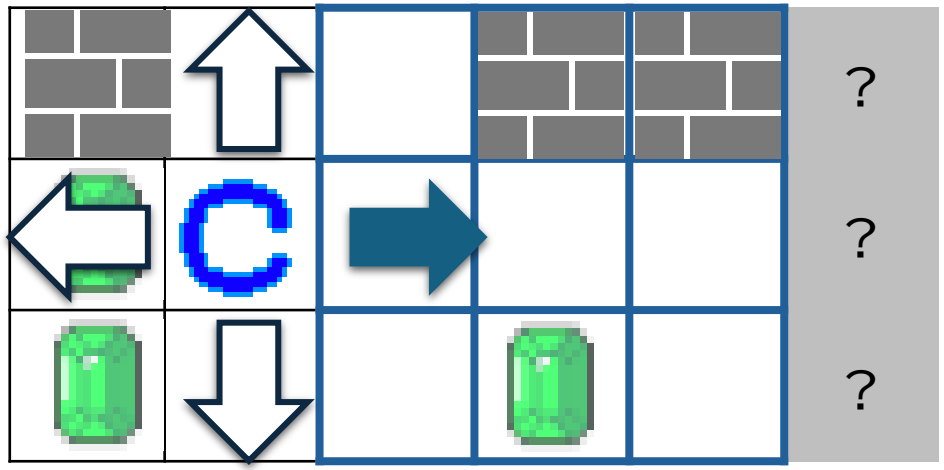
プレーヤーのプログラムからは
自分のまわりしか状況がわかりません



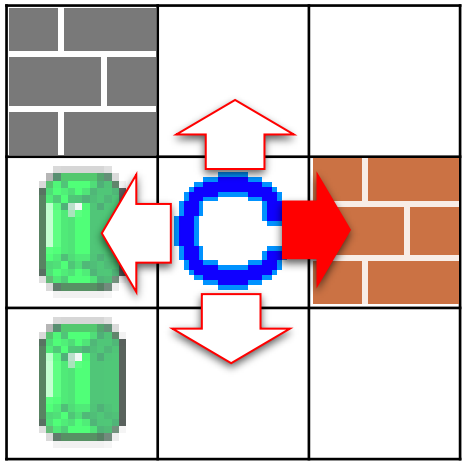
歩く（1マス移動する）



調べる（一方向を遠くまで見る）

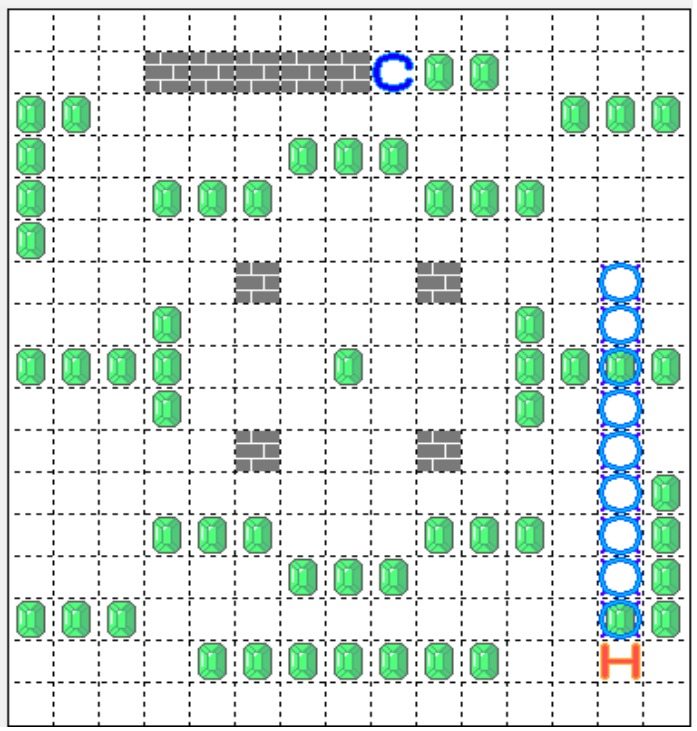


見る（一方向を広く見る）



置く（ブロックを置く）

できることは4種類
×
上下左右の4つの向き



クール(先攻)

ホット(後攻)

何もないマス
…移動できる

ブロック
…ここに移動すると負け
…相手の上に置くと勝ち

アイテム
…上に乗る(アイテムを取る)とポイント
…アイテムを取ったら
前にいた場所はブロックになる

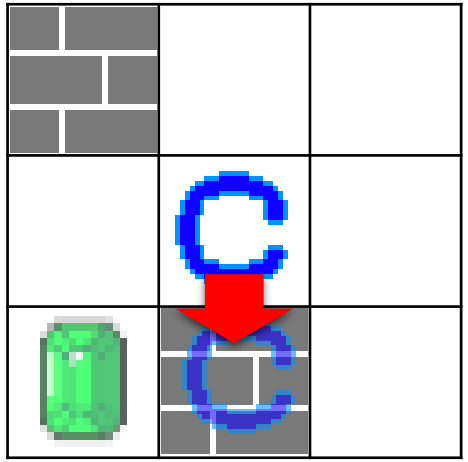
勝ち条件

・アイテムを多くとったら勝ち
・相手にブロックを置くと勝ち

負け条件

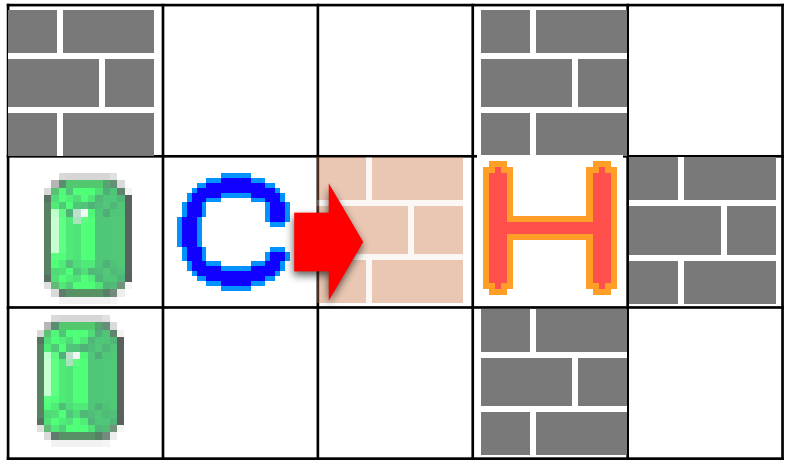
・自分がブロック(マップの外)に移動する
・自分の上下左右がブロックに囲まれる

自分がブロックに移動する



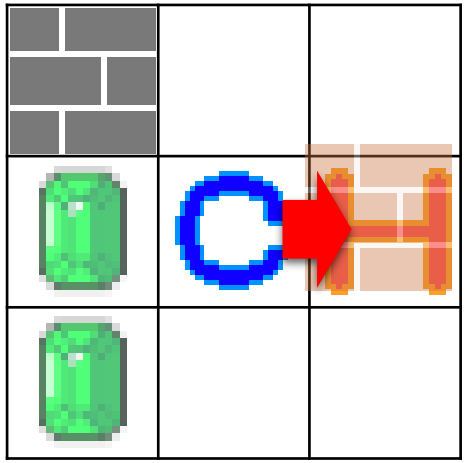
COOLがブロックの上に移動
→COOLの負け　＝　HOTの勝ち

上下左右がブロックに囲まれる



HOTの上下左右にブロックがある
→HOTの負け　＝　COOLの勝ち

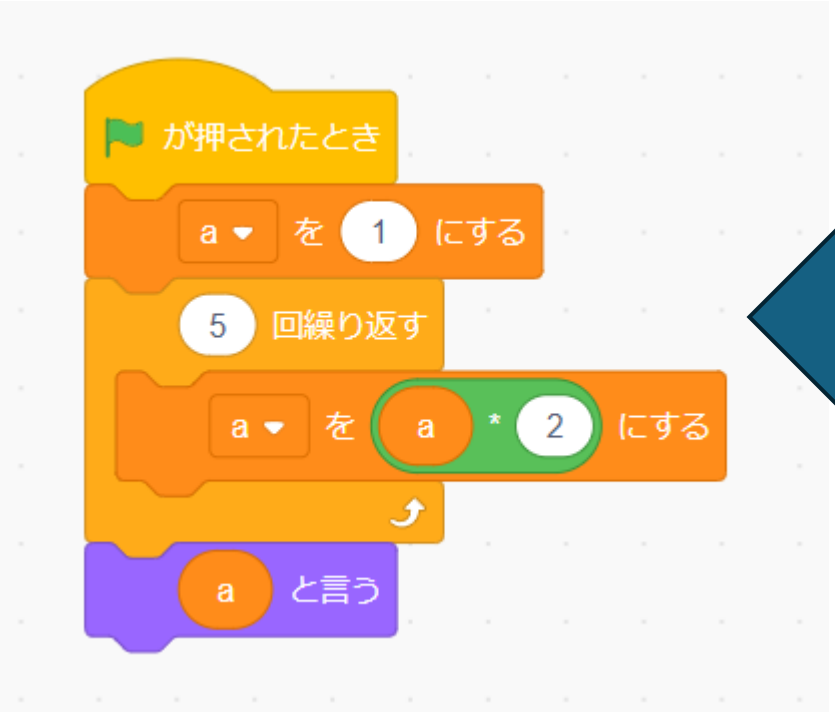
相手にブロックを置くと勝ち



COOLがHOTの上にブロックを置く
→COOLの勝ち

●Python(パイソン)を使ったプログラミングの基本

プログラミング言語



同じ

```
1 a=1
2 for i in range(5):
3     a=a*2
4 print(a)
```

Scratch(スクラッチ)

ビジュアルプログラミング

Python(パイソン)

テキストプログラミング

言葉は違っても意味は同じ

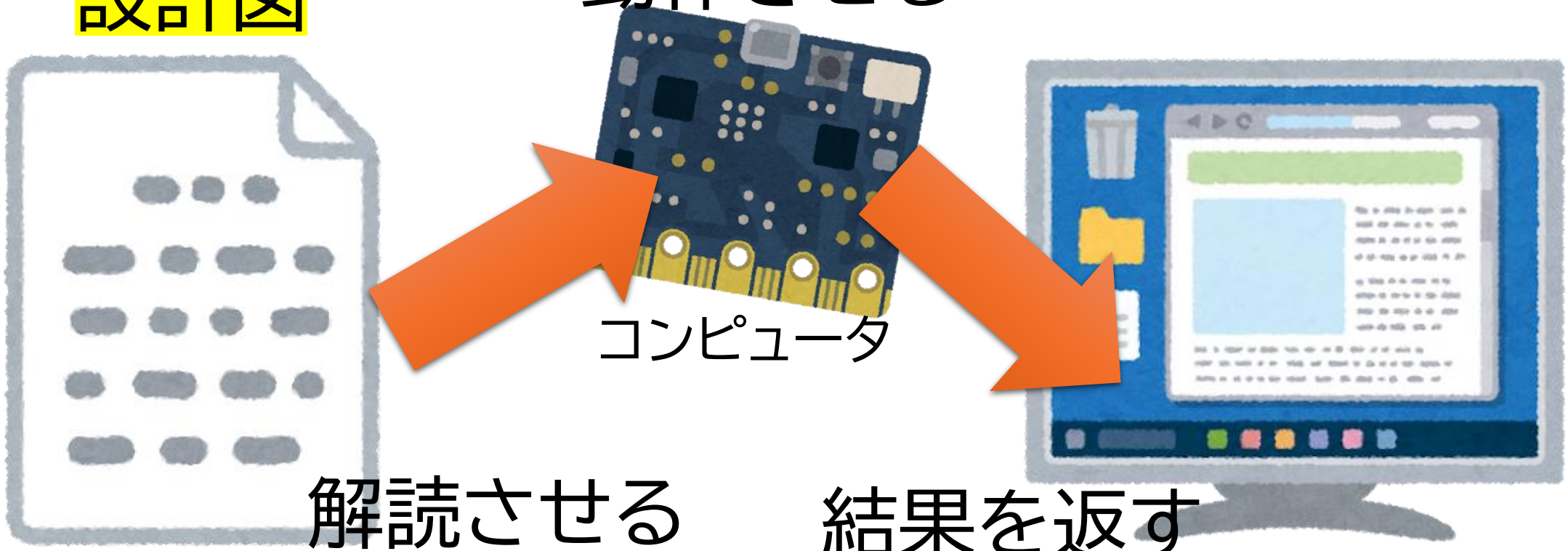
これは犬です。

This is a dog.

プログラミング＝設計図をつくる

設計図

動作させる



コンピュータ

解読させる

結果を返す

ソースコード

Pythonの書き方と注意点

- プログラムは上から下に順番に動作する
- コードは必ず半角文字で打ちこむ

○ print

× ｐｒｉｎｔ

- 行のあたまを下げるときは「Tab」キー

Pythonの命令

- print　　プリント：画面に文字を出す
- input　　インプット：キーボードで打ち込む
- if　　イフ：もし～なら
- while　　ホワイル：～の間ずっと
- for　　フォー：～回繰り返す

- 変数　　へんすう
- リスト

画面に文字を出す

```
print("こんにちは")
```

画面に計算結果を出す

```
print(1+1)
print(5-2)
print(3*5)
print(15/3)
print(17%5)
```

結果を書きましょう

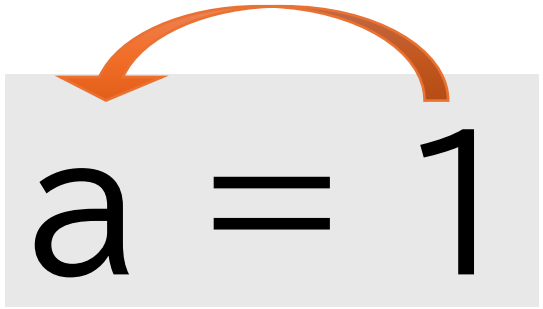
＊は　かけ算
／は　わり算
%は　あまりのある割り算の　あまりの部分

変数に情報を覚えさせる

```
a=1
b=2
c=3
d=a+b+c
print(d)
```

結果を書きましょう

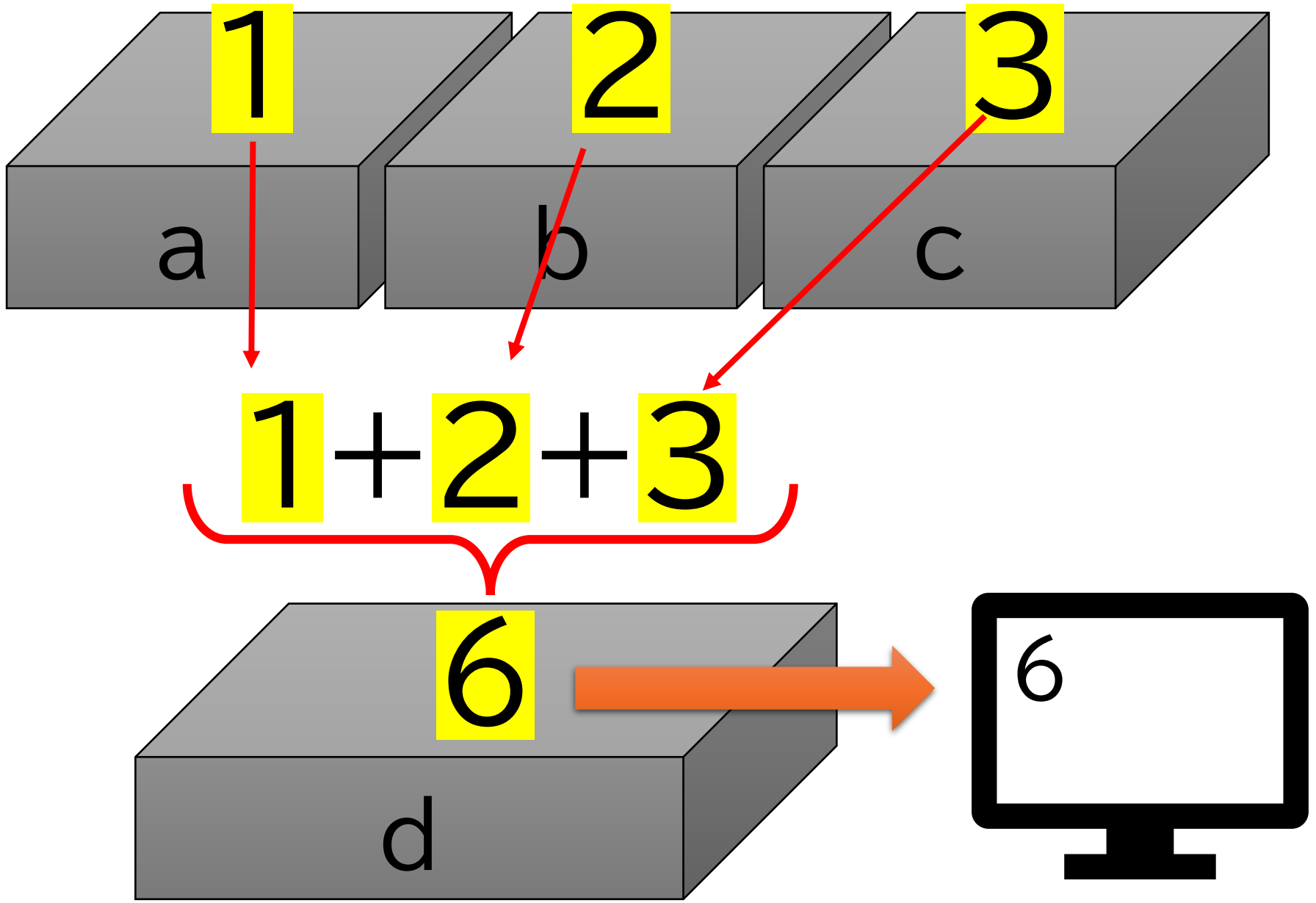
へんなかず？　→　ころころ変わる数



プログラミングでは
算数・数学での
左と右が等しい
という意味ではありません

変数(へんすう)a は
いろんな文字や数字を保存できる入れ物です

```
a=1
b=2
c=3
d=a+b+c
print(d)
```



画面に計算結果を出す

```
print(1+1)
print(5-2)
print(3*5)
print(15/3)
print(17%5)
```

結果を書きましょう

＊は　かけ算
／は　わり算
%は　あまりのある割り算の　あまりの部分

プログラムに情報を入力する

```
z=input()
print(z)
y=input("名前は?")
print("こんにちは",y,"さん")
```

if(イフ) もし～なら

```
nenrei=8
if␣nenrei==8:
␣␣␣␣print("８歳")
```

Tabキーで字下げする

if(イフ) もし～なら～でなければ

```
nenrei=2
if␣nenrei==8:
␣␣␣␣print("８歳")
else:
␣␣␣␣print("８歳じゃない")
```

if(イフ) もし～なら～でなければ

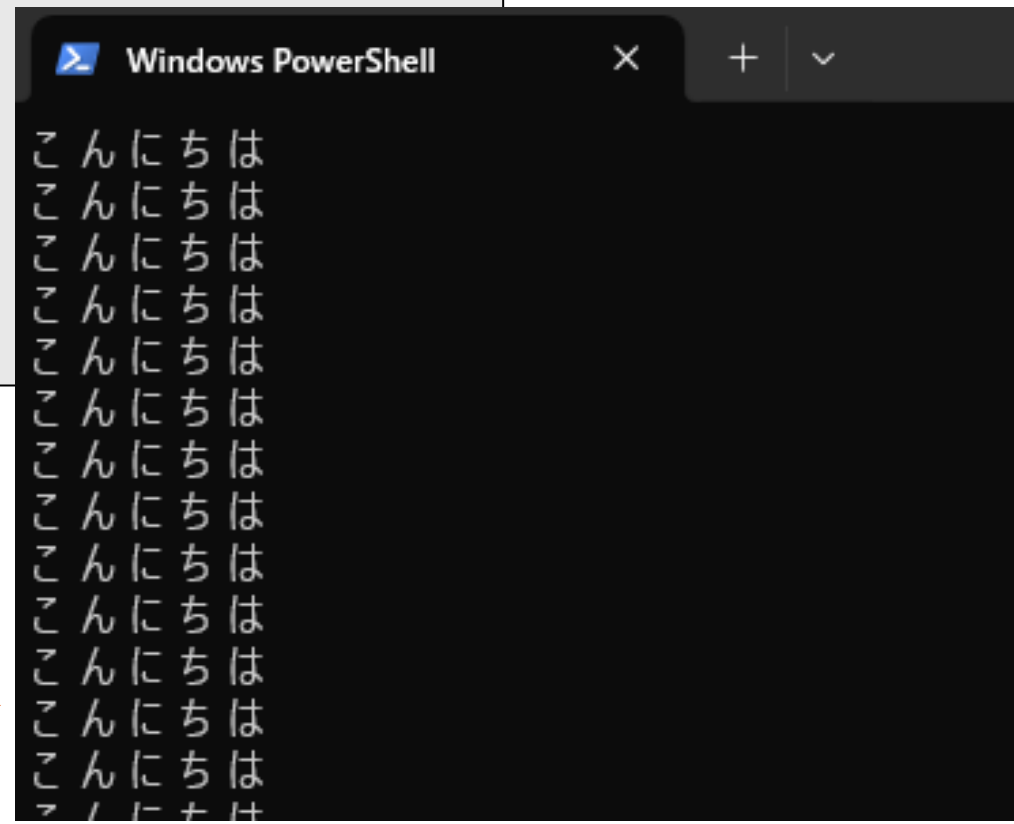
```
nenrei=2
if␣nenrei==8:
␣␣␣␣print("８歳")
else:
␣␣␣␣print("８歳じゃない")
```

if(イフ) もし～なら～でなければ

```
nenrei=2
if␣nenrei>=12:
␣␣␣␣print("中学生以上")
elif␣nenrei>=6:
␣␣␣␣print("小学生")
else:
␣␣␣␣print("幼児")
```

while(ホワイル) ずっと

```
while␣True:
␣␣␣␣print("こんにちは")
```

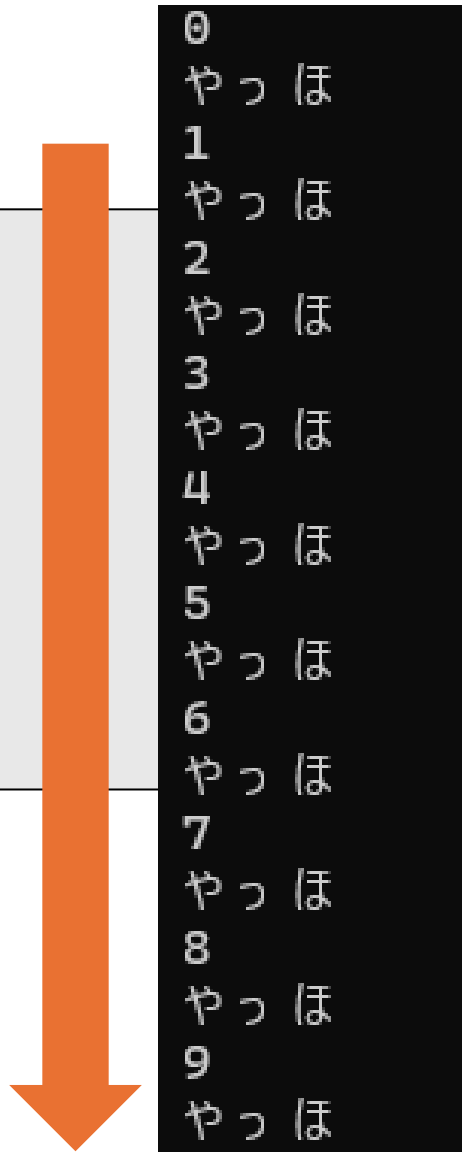


無限にループしてしまうので

キーボードで Ctrl+C を押し中断させる

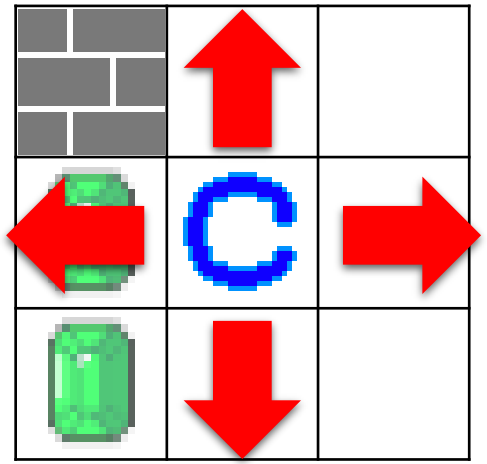
for(フォー) ○回くりかえす

```
for␣i␣range(10):
␣␣␣␣print(i)
␣␣␣␣print("やっほ")
```

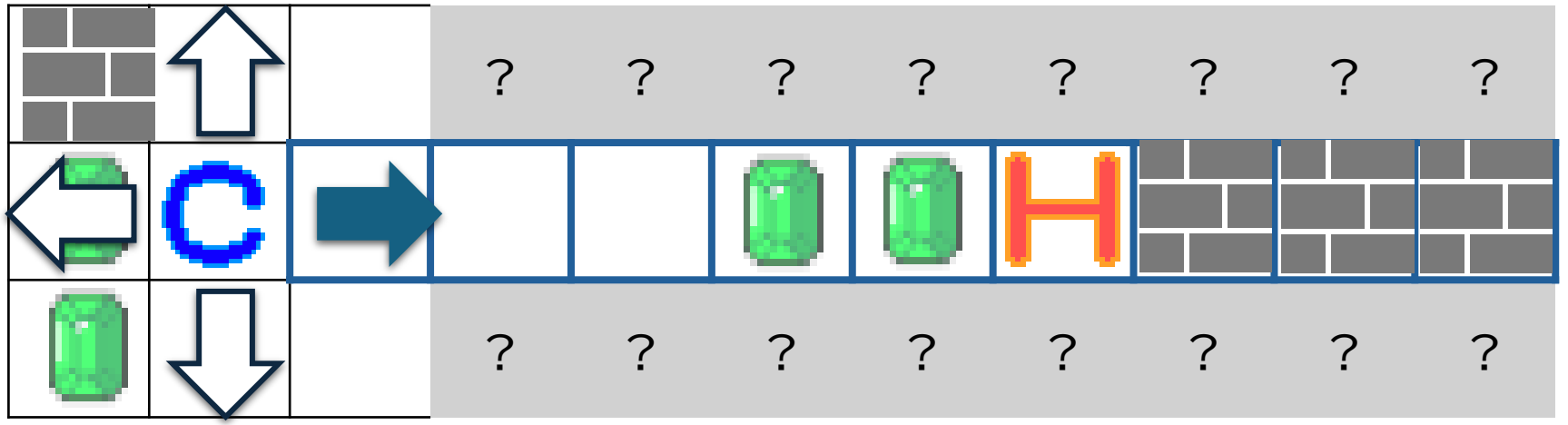


0から10までの 10回くりかえす

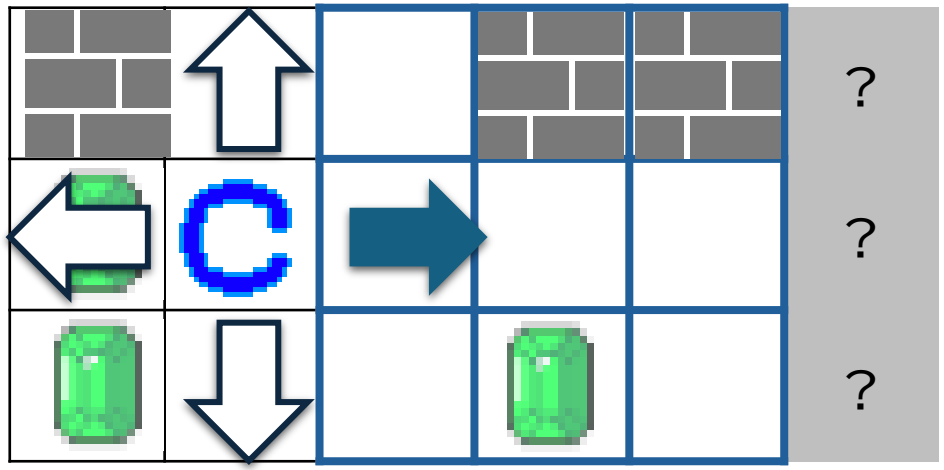
●CHaserのキャラクターを動かしてみる



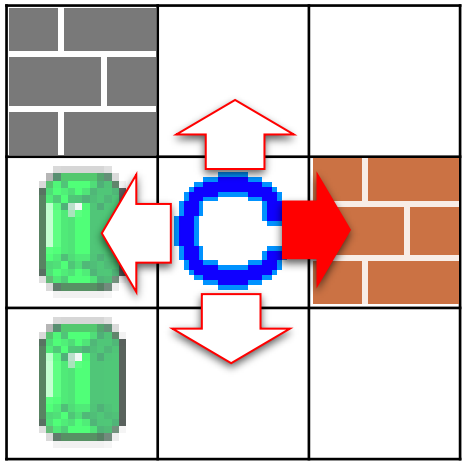
歩く（1マス移動する）



調べる（一方向を遠くまで見る）



見る（一方向を広く見る）



置く（ブロックを置く）

できることは4種類
×
上下左右の4つの向き

Chaserのキャラクタを動かす命令

cl.get_ready()　…行動の前に必ず実行する
サーバに準備完了を伝えて、周囲の状況を調べる

歩く（1マス移動する）

cl.walk_up()	上
cl.walk_down()	下
cl.walk_left()	左
cl.walk_right()	右

調べる（一方向を遠くまで見る）

cl.search_up()	上
cl.search_down()	下
cl.search_left()	左
cl.search_right()	右

見る（一方向を広く見る）

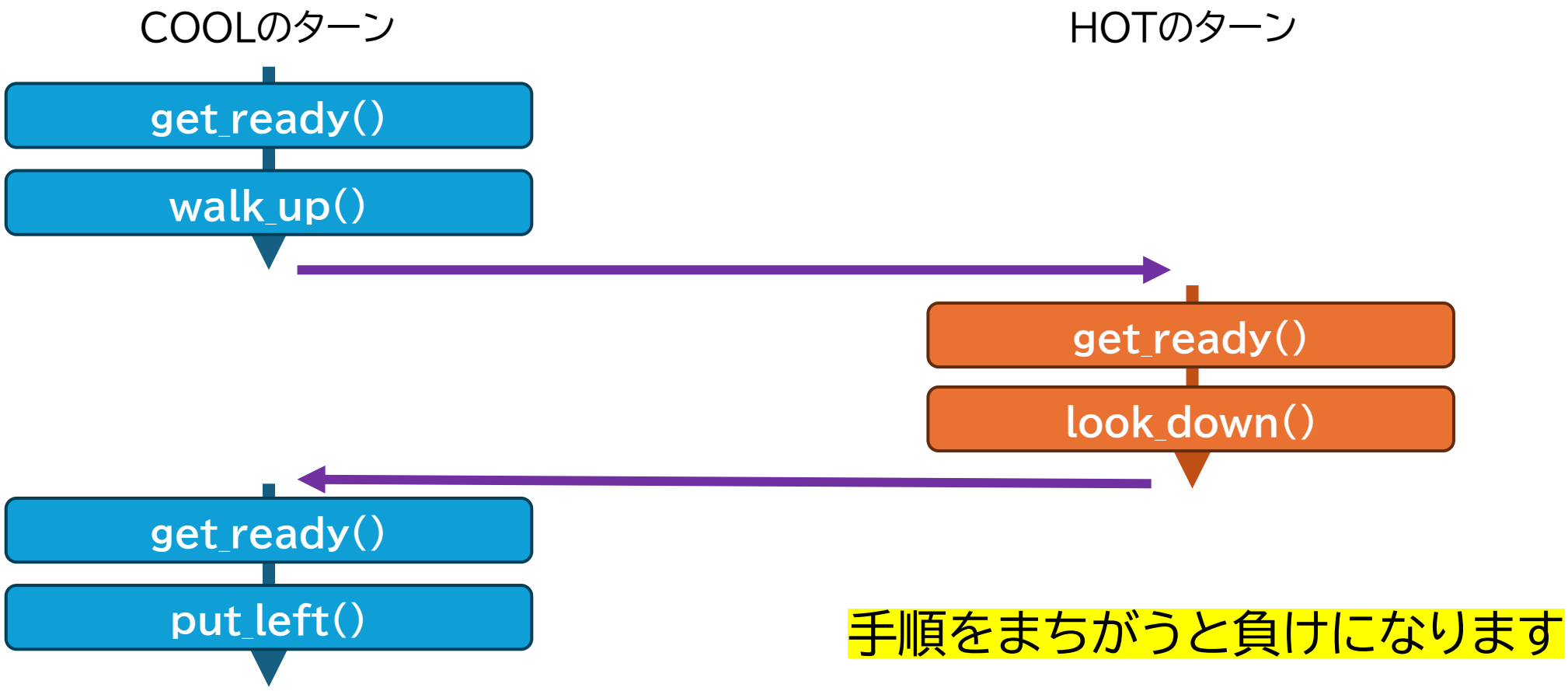
cl.look_up()	上
cl.look_down()	下
cl.look_left()	左
cl.look_right()	右

置く（ブロックを置く）

cl.put_up()	上
cl.put_down()	下
cl.put_left()	左
cl.put_right()	右

CHaserのプログラムの約束

- 命令の前に、必ず **get_ready()** を実行する。



手順をまちがうと負けになります

●演習問題にチャレンジ

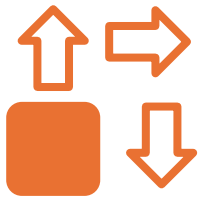
レベル1　client.walk()　移動するメソッド

もんだい1

左に歩き続けよう

もんだい2

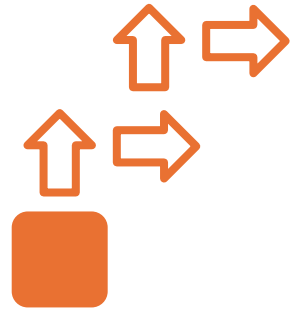
その場でぐるぐるしよう



もんだい3

じぐざぐと上に行こう

（じぐざぐは自分の好きな感じでOK）



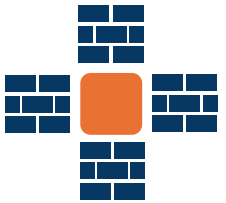
レベル1　client.put()　ブロックを置くメソッド

もんだい4

左にブロックを置こう

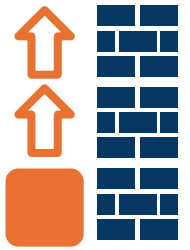
もんだい5

自分の周りにブロックを置いて自滅しよう



もんだい6

上に行きながら右にブロックを置こう

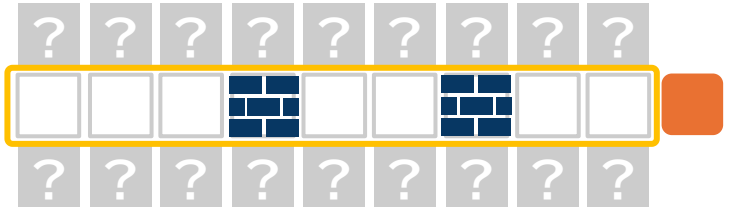


レベル1　client.look()　client.search()

周りを見るメソッド

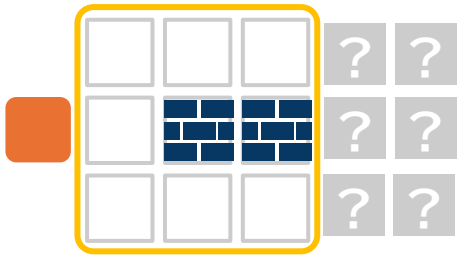
もんだい7

左にsearchしよう



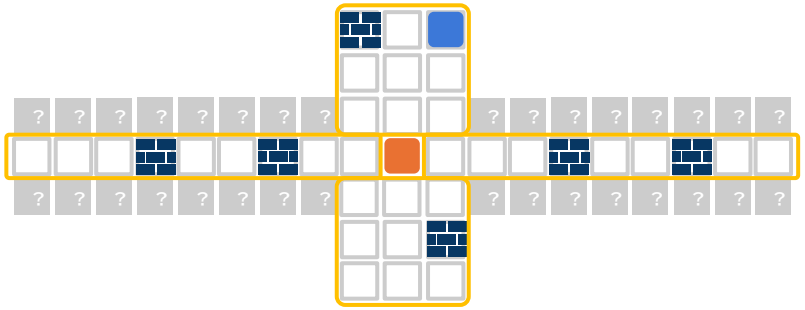
もんだい8

右にlookしよう



もんだい9

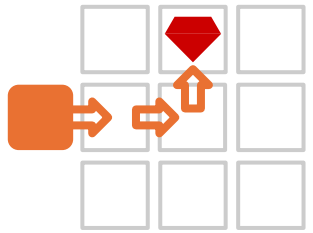
上下にlook、左右にsearchしよう



レベル2　if 分岐:もし～～だったら

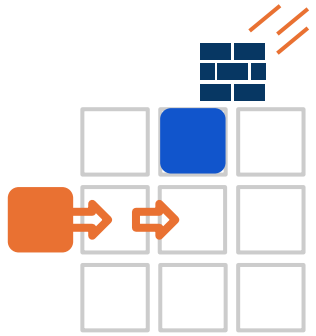
もんだい10

上下左右にアイテムがあったら取りに行こう



もんだい11

敵がいたらputしよう



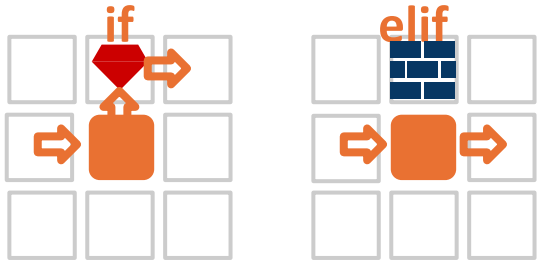
レベル2　elif　分岐:もし ～～じゃなくて〇〇だったら

もんだい12

右に進み続けて上にアイテムがあれば取りに行こう

そうでなければブロックを置こう

（ifとelifは逆でもOK）



もんだい13

右に進み続けてブロックにぶつかったときランダムに移動してみよう

import random が必要

randint(a, b)は a 以上 b 以下の 数字がランダムで作られる