

ご注文は関数ですか?



akouryy



# 関数型プログラミング

# 関数型言語Haskell

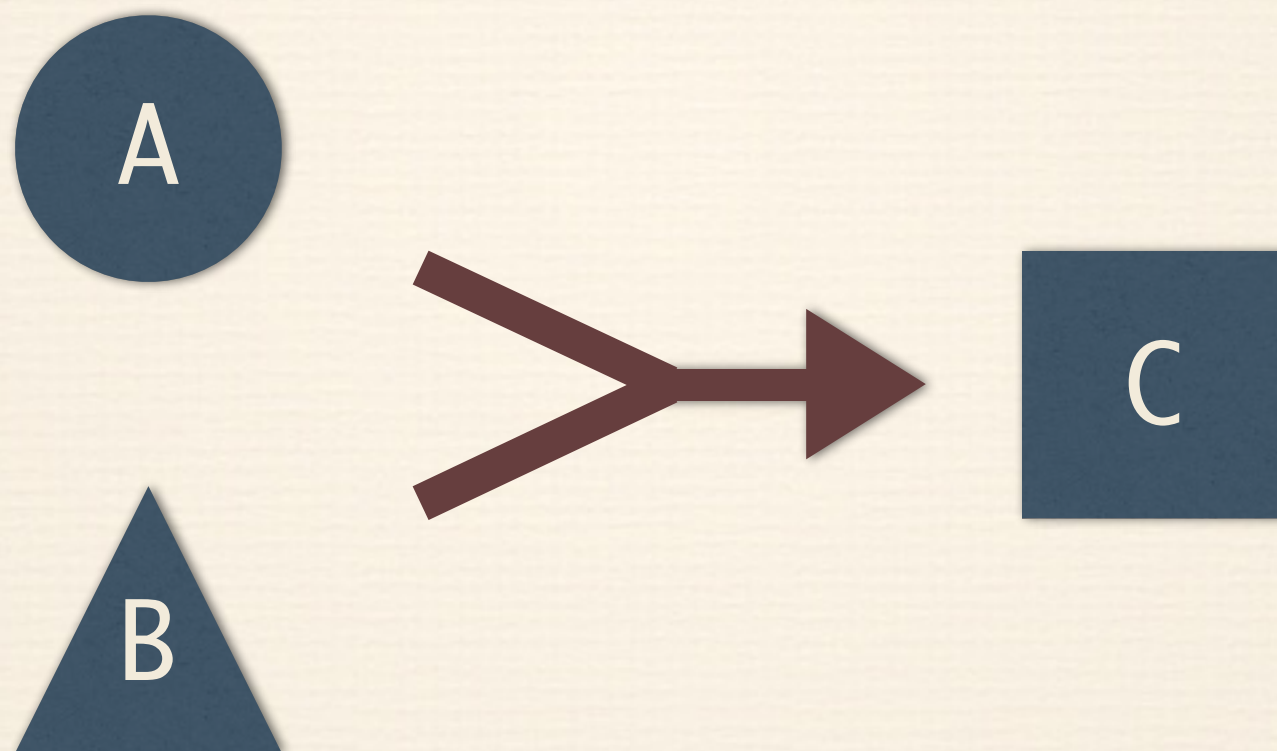
- ❖ 値は不変(**変数**は存在しない)
- ❖ 型がとても大事
- ❖ 配列より**片方向連結リスト**を多用
- ❖ リスト操作が豊富(**高階関数**)



# Haskellの関数

# カリー化

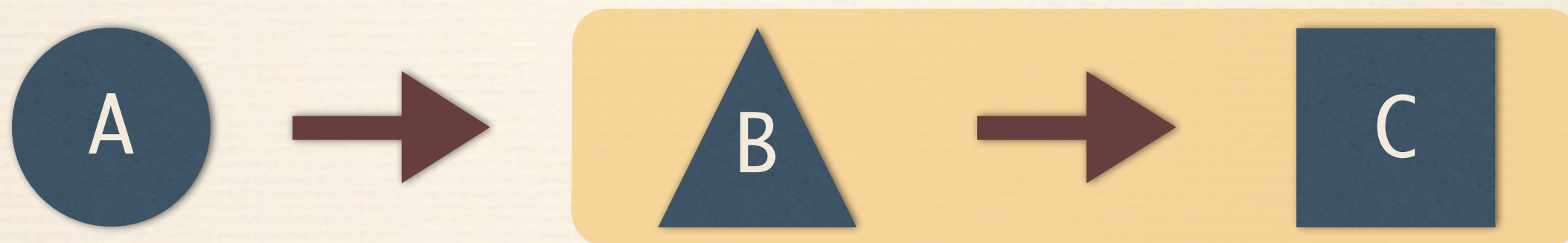
❖ 複数の引数をとる関数





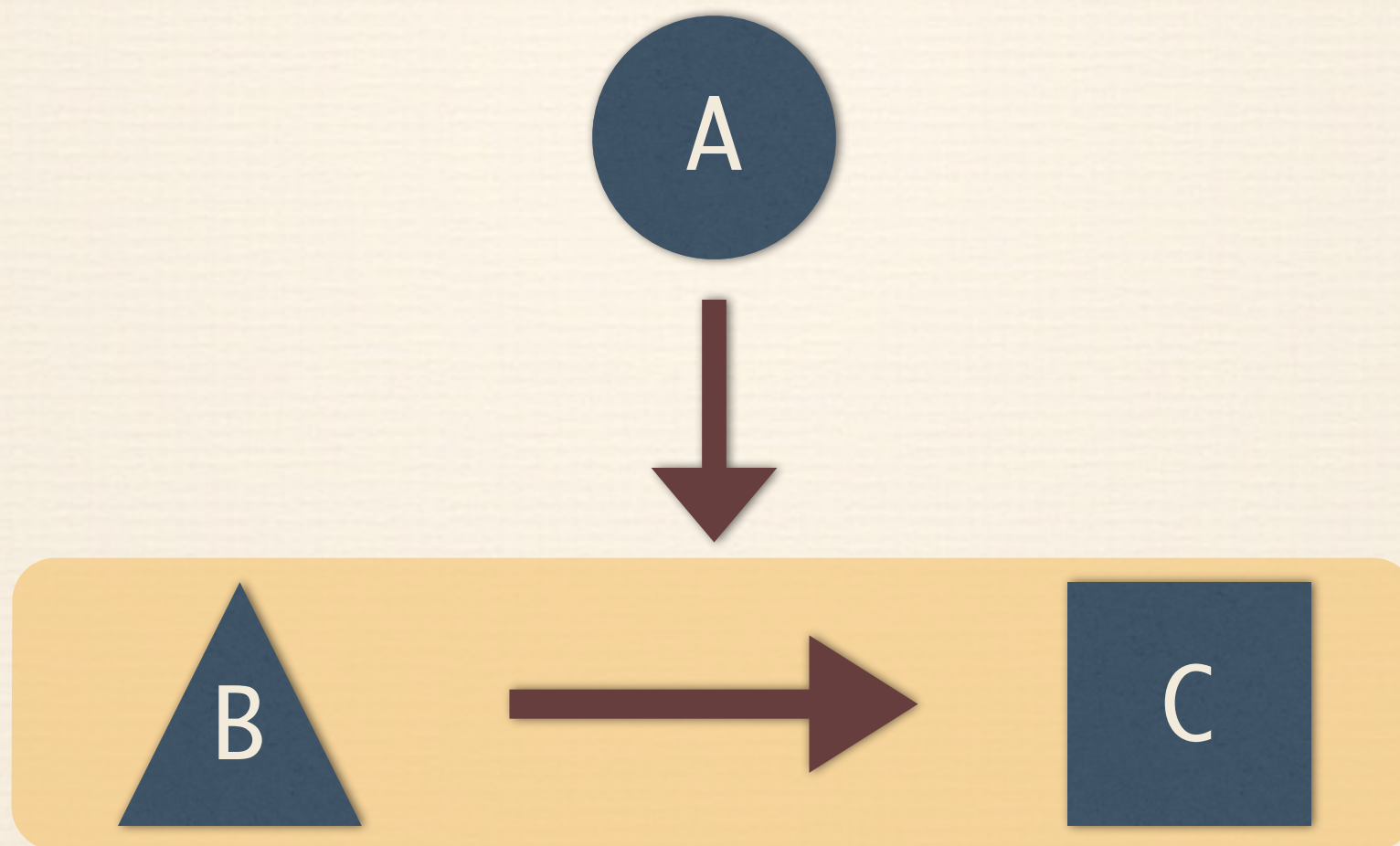
# カリー化

❖ 複数の引数をとる関数



# カリー化

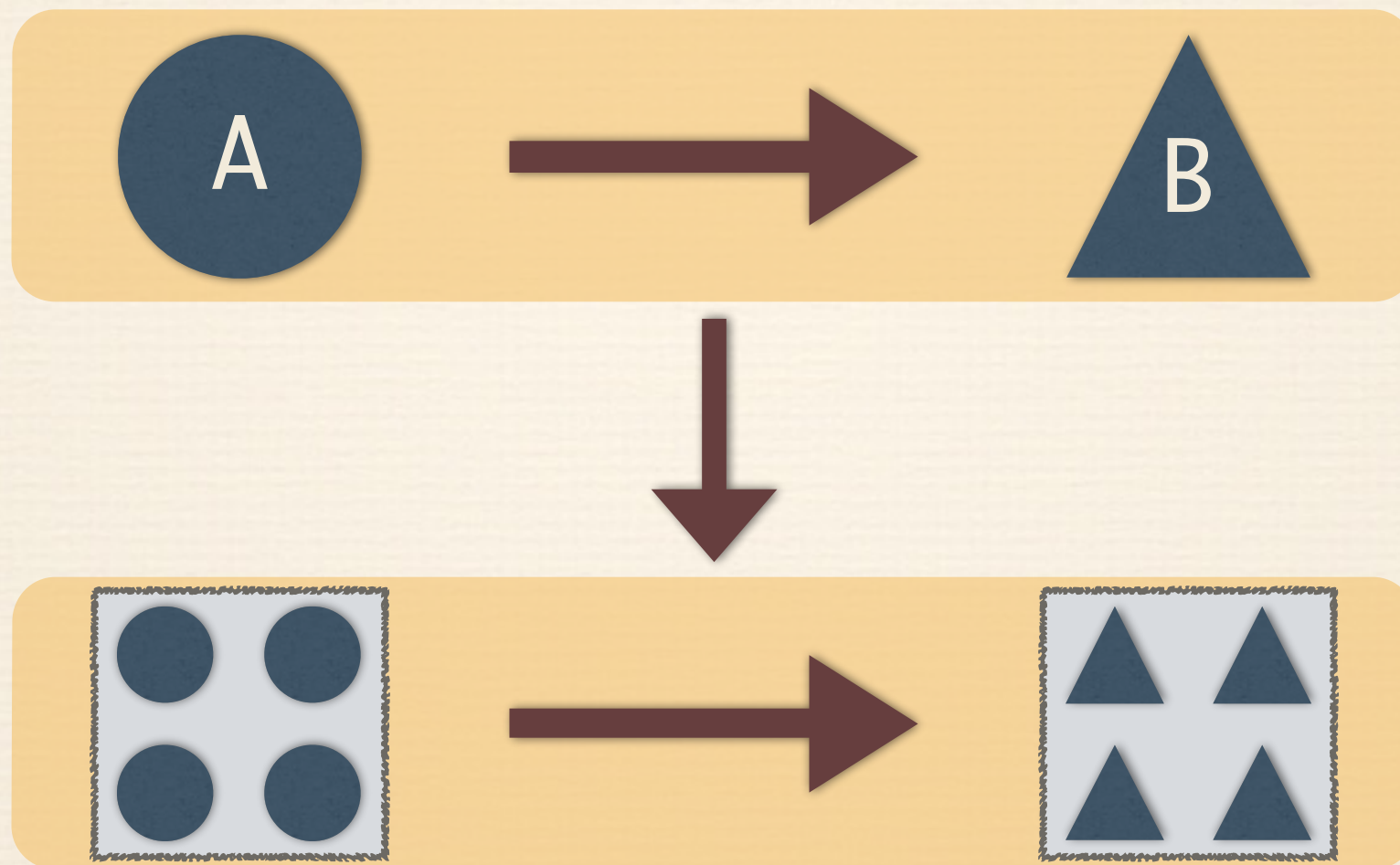
❖ 複数の引数をとる関数





# map関数

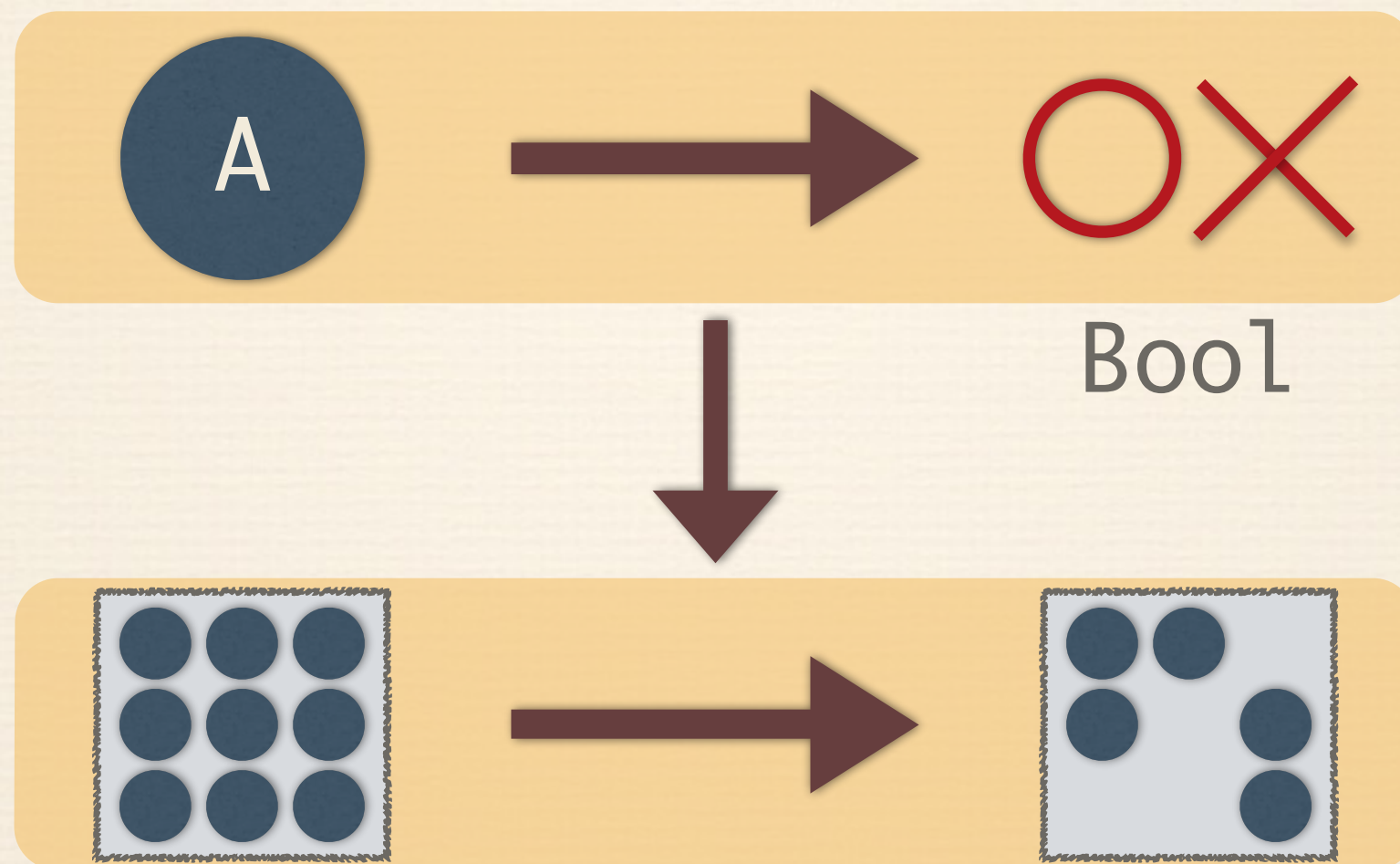
❖ 関数をリストの各要素に適用する関数





# filter関数

❖ 述語を満たす要素のみを集める関数





# 数独



		4			5	7		
					9	4		
3	6							8
7	2			6				
			4		2			
				8			9	3
4							5	6
		5	3					
		6	1			9		



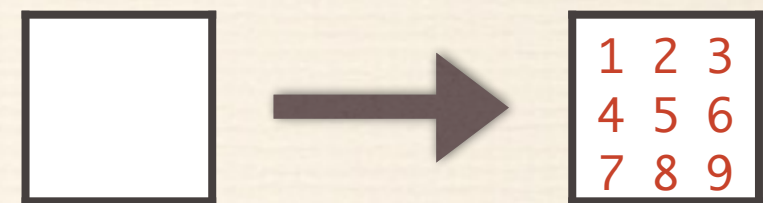
1	8	4	6	2	5	7	3	9
5	7	2	8	3	9	4	6	1
3	6	9	7	4	1	5	2	8
7	2	8	9	6	3	1	4	5
9	5	3	4	1	2	6	8	7
6	4	1	5	8	7	2	9	3
4	1	7	2	9	8	3	5	6
2	9	5	3	7	6	8	1	4
8	3	6	1	5	4	9	7	2

選択肢を列举



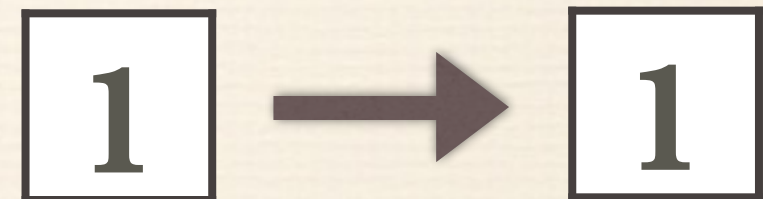
# 1マスの選択肢

❖ 0 → 1～9の9通り



choice

❖ 1～9 → その数字で固定



		4			5	7		
--	--	---	--	--	---	---	--	--



		4			5	7		
--	--	---	--	--	---	---	--	--

choice



1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	4	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	5	7	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
-------------------------	-------------------------	---	-------------------------	-------------------------	---	---	-------------------------	-------------------------

		4			5	7		
--	--	---	--	--	---	---	--	--



map choice

1 2 3	1 2 3	4	1 2 3	1 2 3	5	7	1 2 3	1 2 3
4 5 6	4 5 6		4 5 6	4 5 6			4 5 6	4 5 6
7 8 9	7 8 9		7 8 9	7 8 9			7 8 9	7 8 9



		4			5	7		
					9	4		
3	6							8
7	2			6				
			4		2			
				8			9	3
4							5	6
		5	3					
		6	1			9		

# map (map choice)

1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	4	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	5	7	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	9	4	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
3	6	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	8
7	2	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	6	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	4	1 2 3 4 5 6 7 8 9	2	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	8	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	9	3
4	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	5	6
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	5	3	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	6	1	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9



# 直積

❖ ここでリストの直積(cp)を考える

cp  $[[1, 2], [3], [4, 5, 6]]$

$==$   $[[1, 3, 4], [1, 3, 5], [1, 3, 6],$   
 $[2, 3, 4], [2, 3, 5], [2, 3, 6]]$

# 盤面全体の選択肢

❖ 盤面全体の選択肢はchoicesの直積

map (map cp)



1	1	4
1	1	1
3	6	1

1	1	4
1	1	1
3	6	2

1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	4
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
3	6	1 2 3 4 5 6 7 8 9

1	1	4
1	1	1
3	6	3

1	1	4
1	1	1
3	6	4

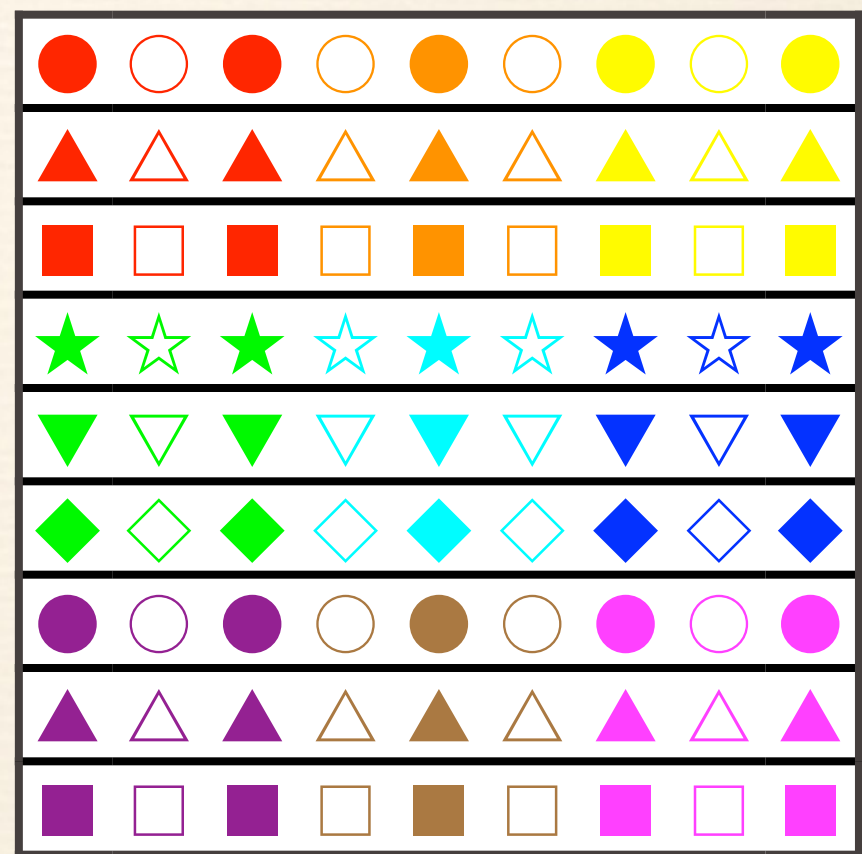
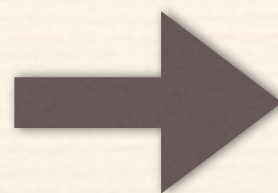
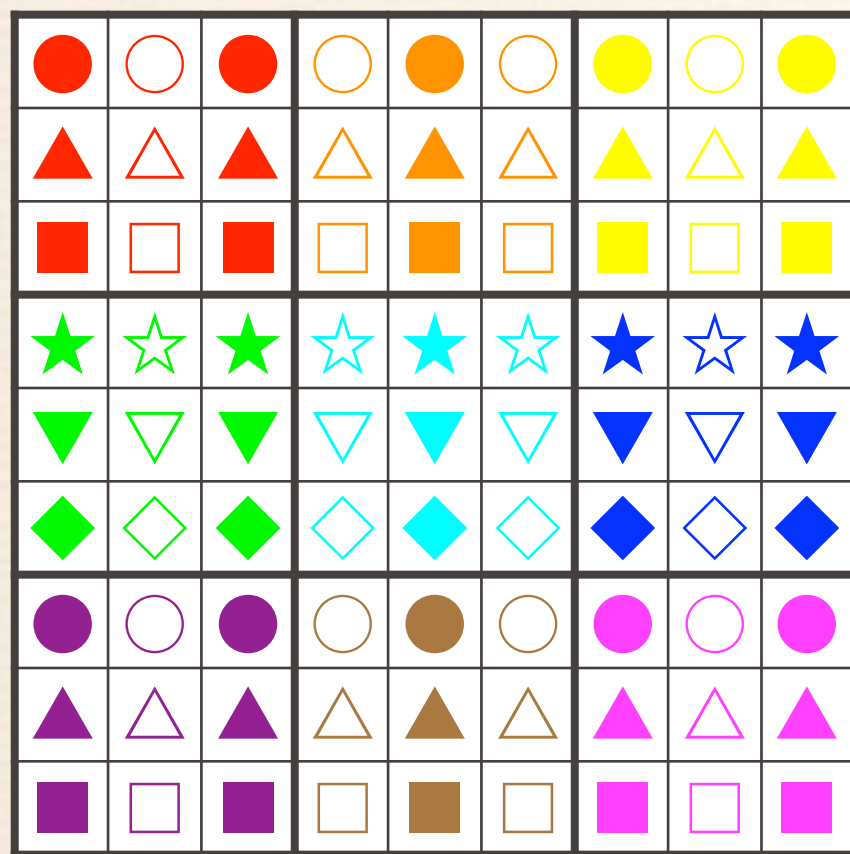
9	9	4
9	9	9
3	6	9

# 正解判定



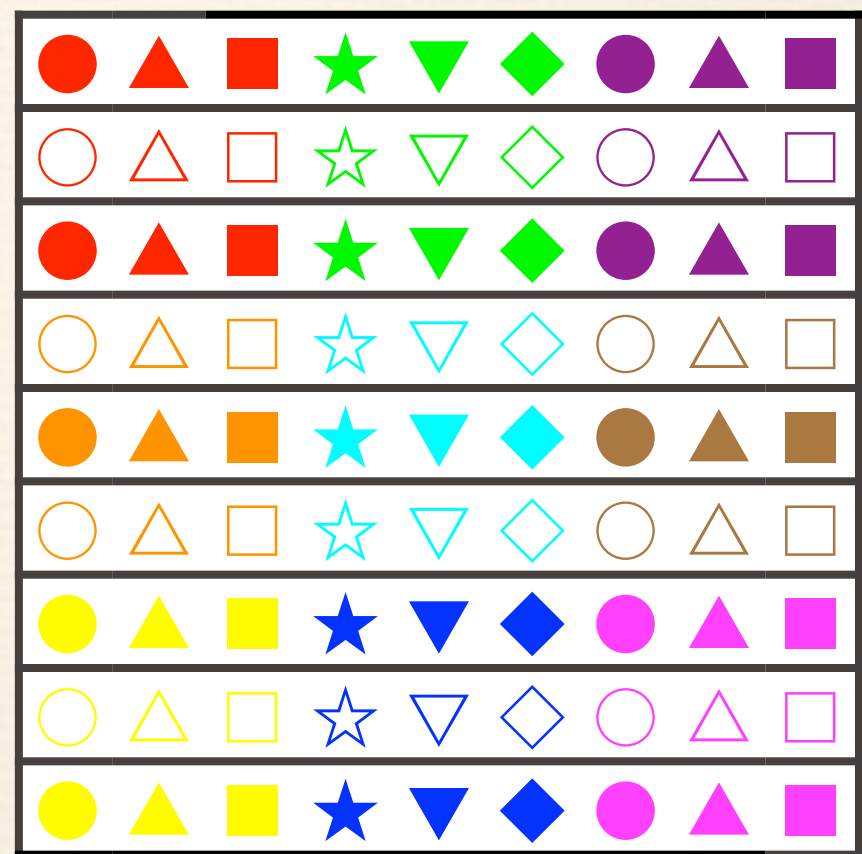
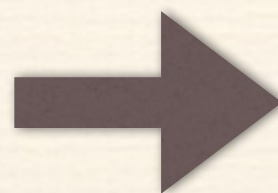
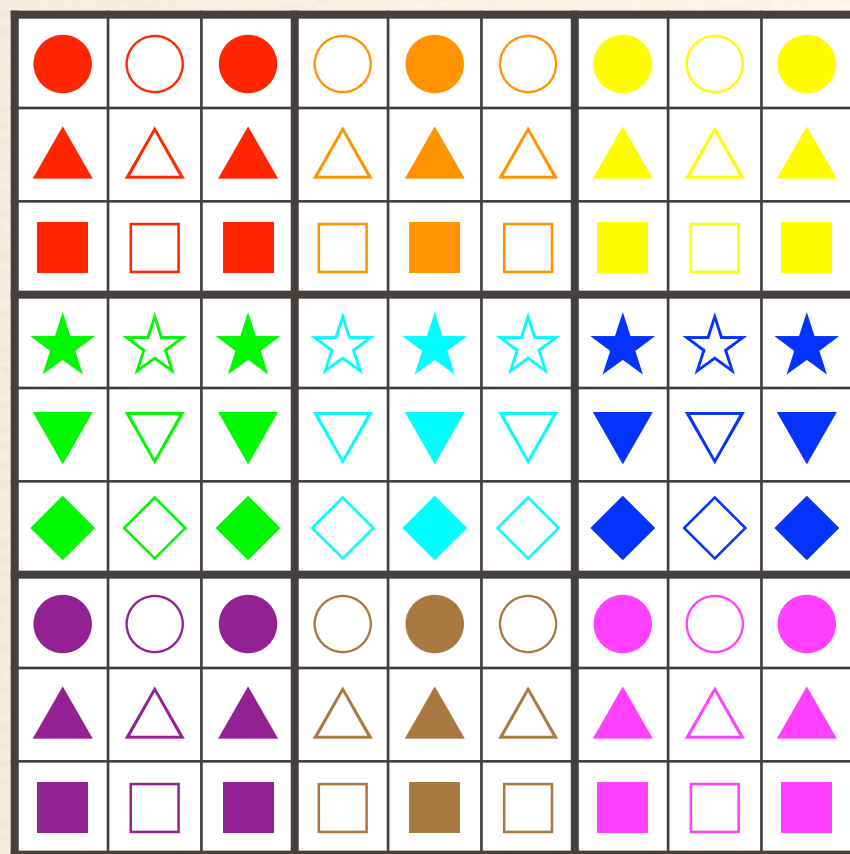
# 行、列、ブロック

行



# 行、列、ブロック

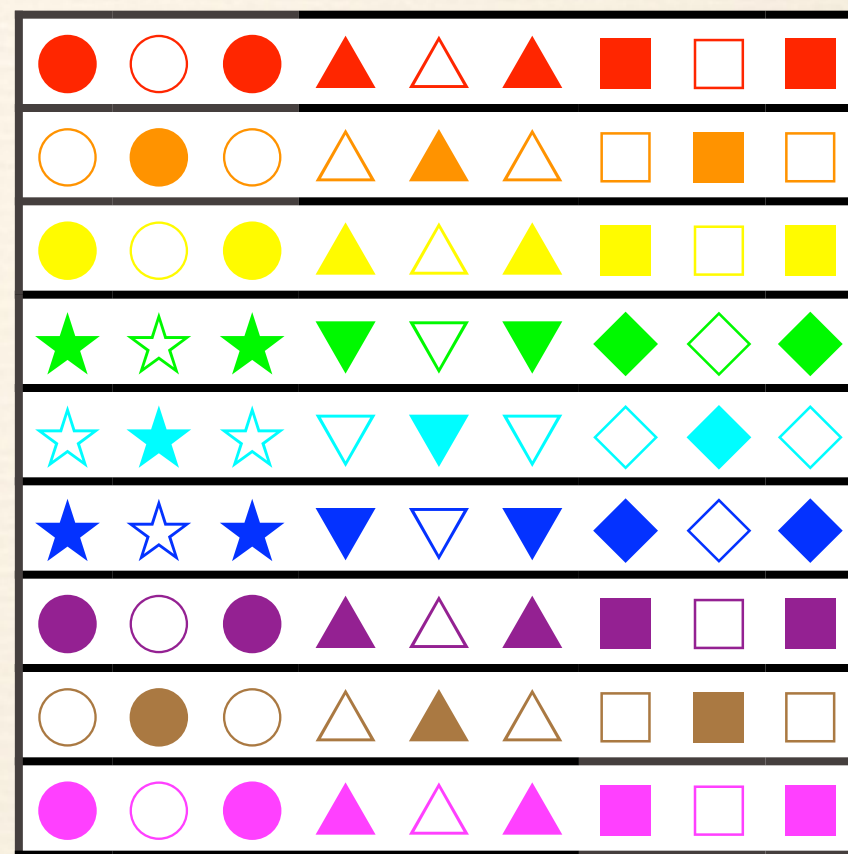
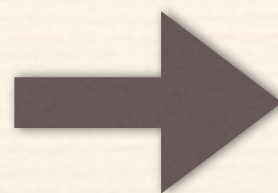
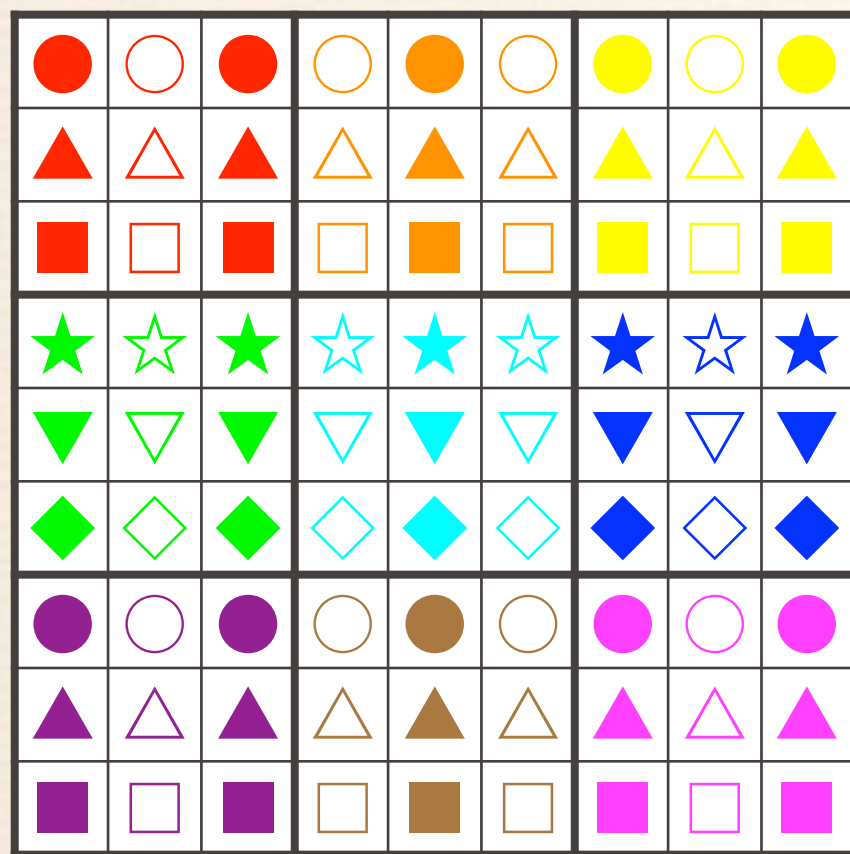
列





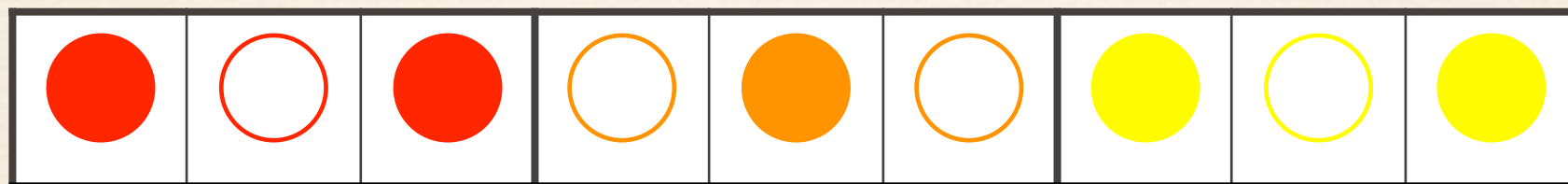
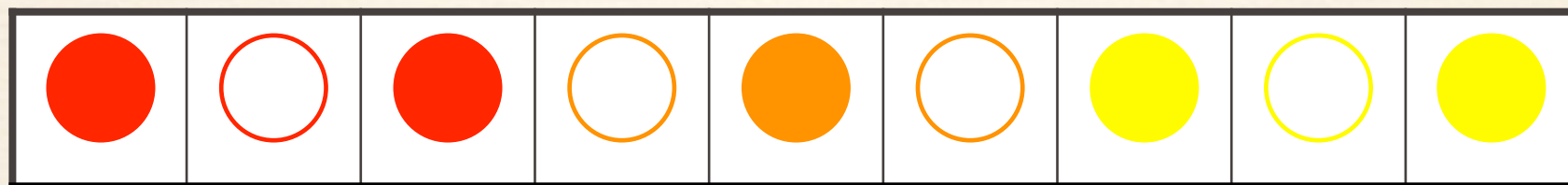
# 行、列、ブロック

## ブロック



# 行、列、ブロック

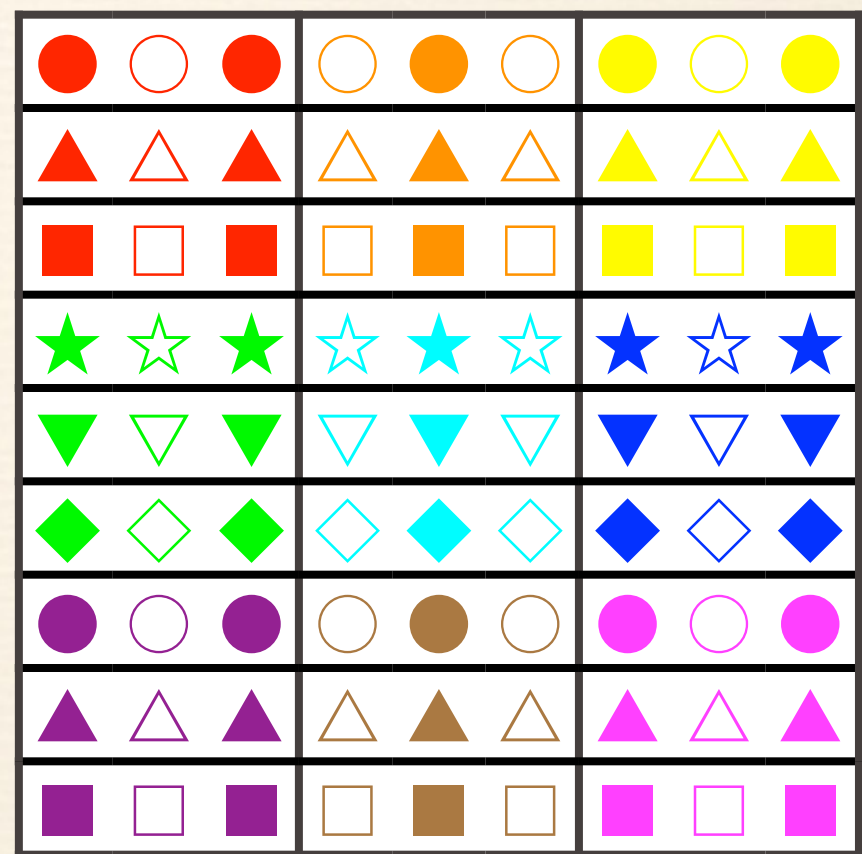
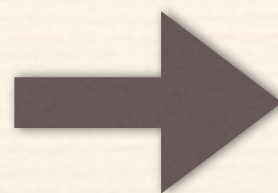
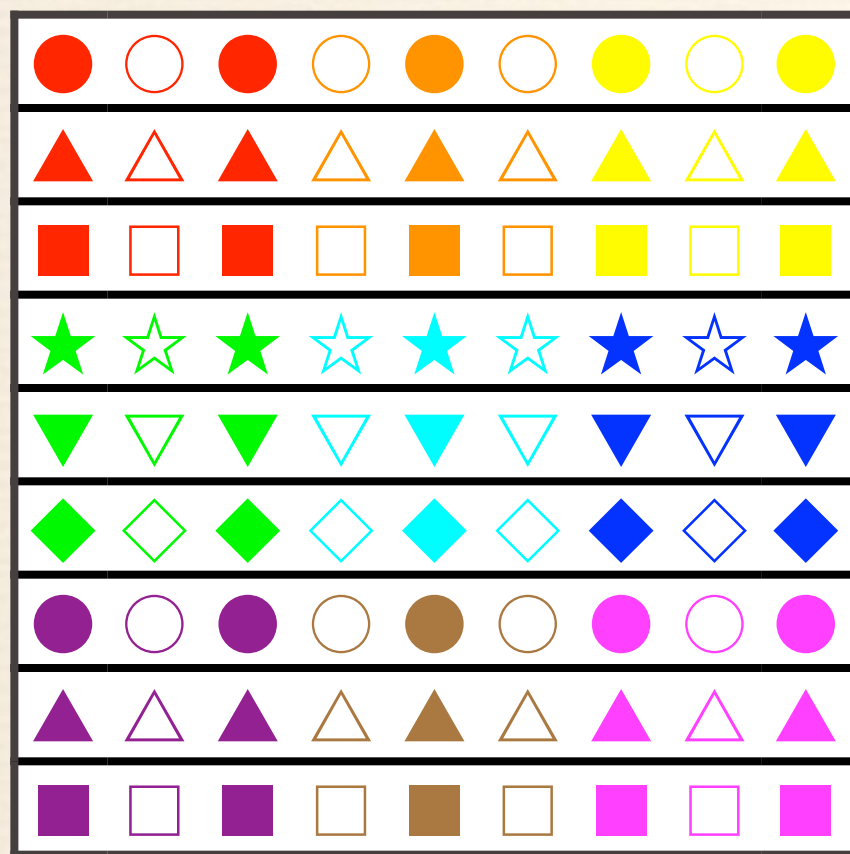
ブロック





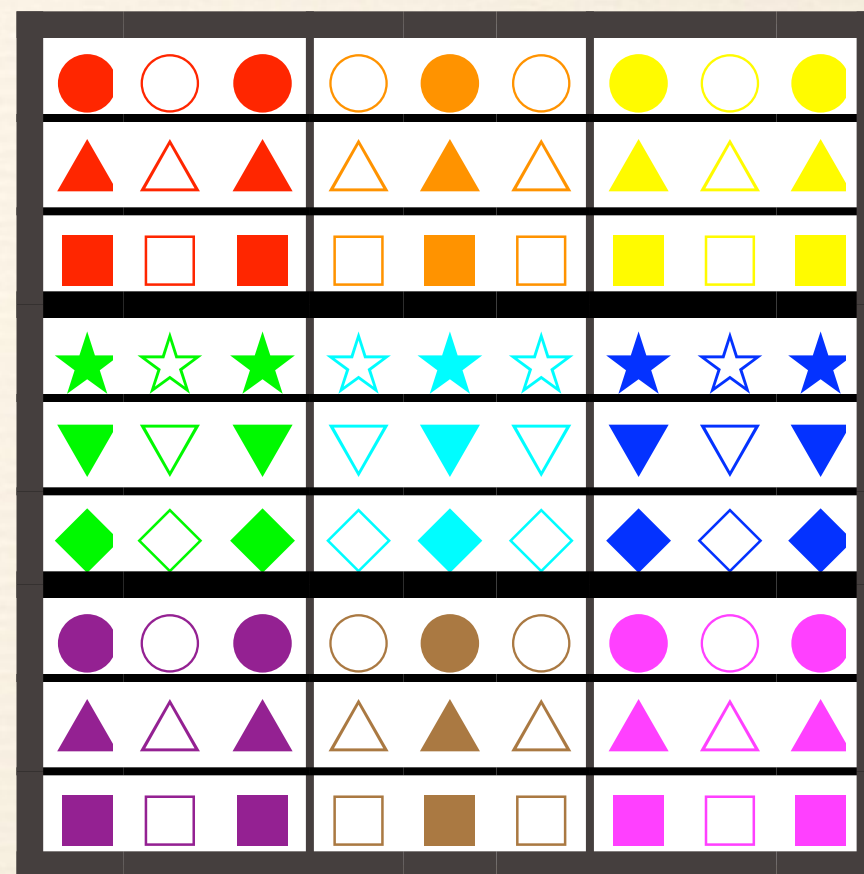
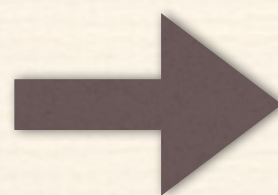
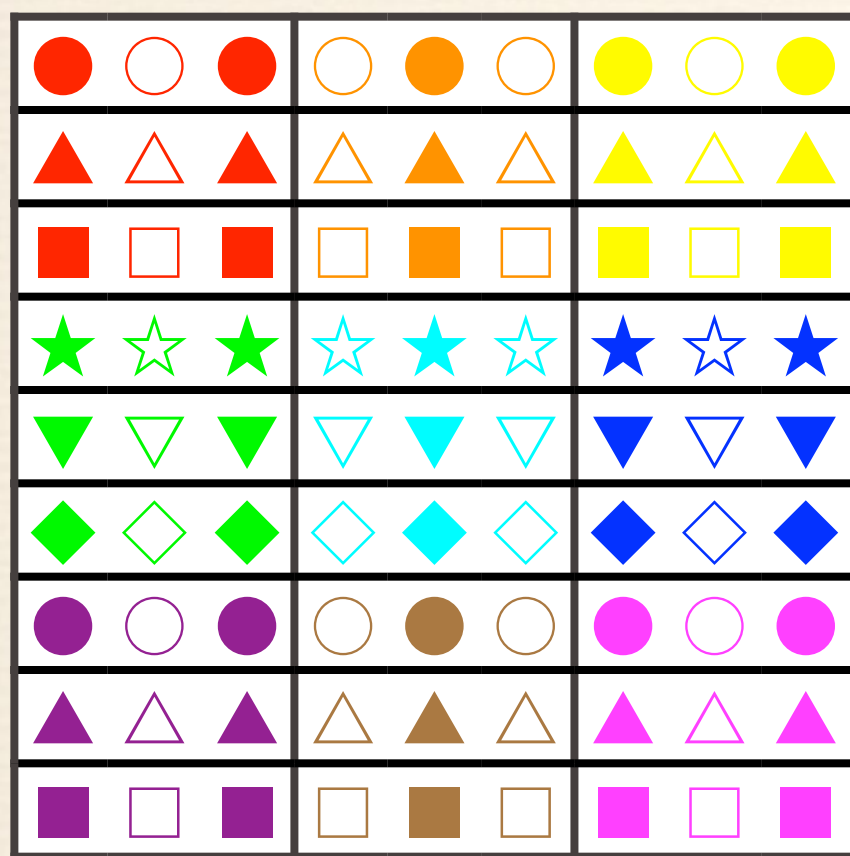
# 行、列、ブロック

## ブロック



# 行、列、ブロック

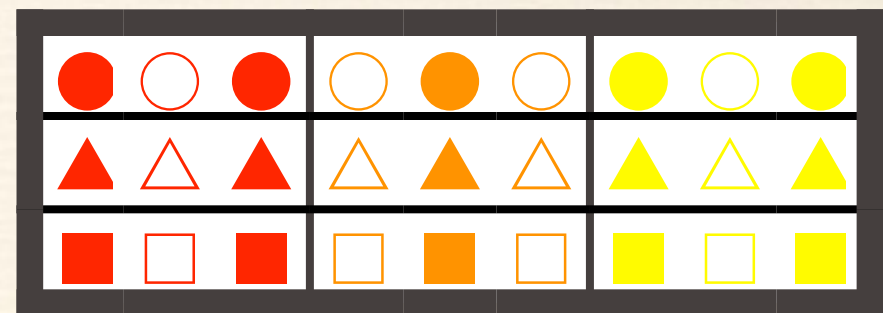
## ブロック





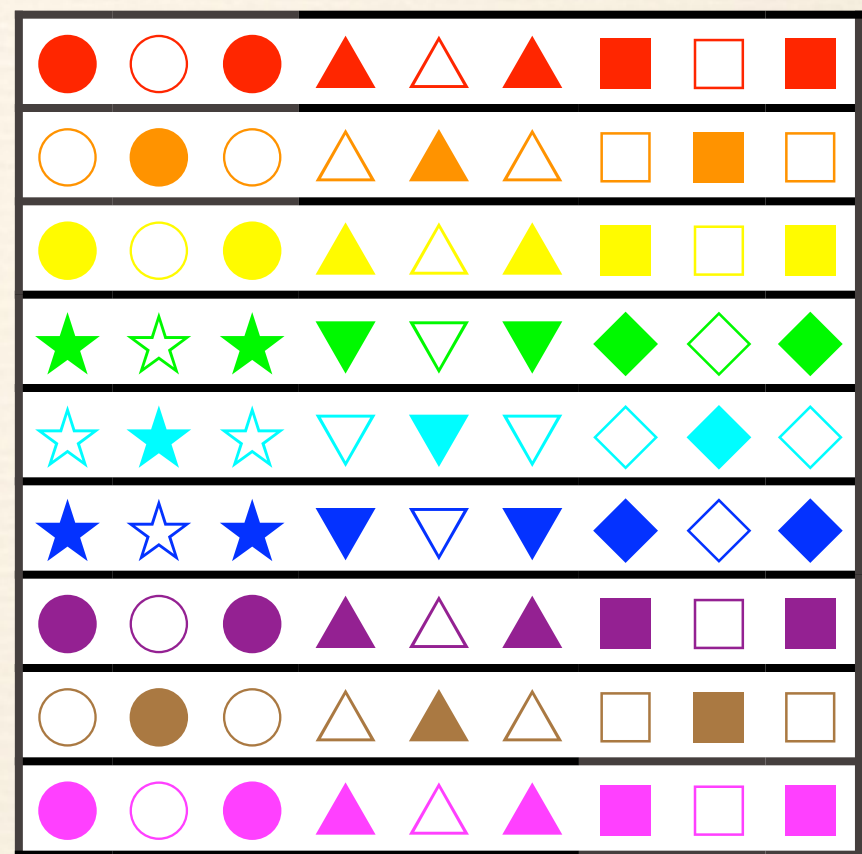
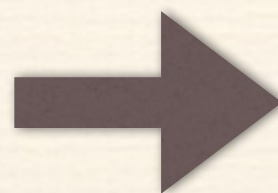
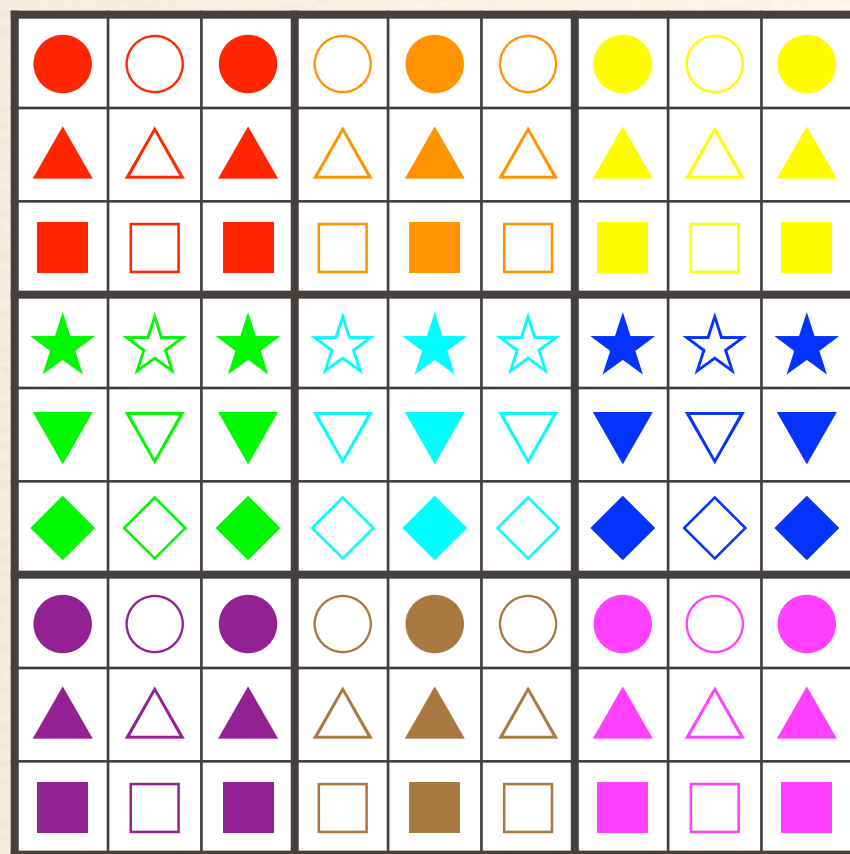
# 行、列、ブロック

ブロック



# 行、列、ブロック

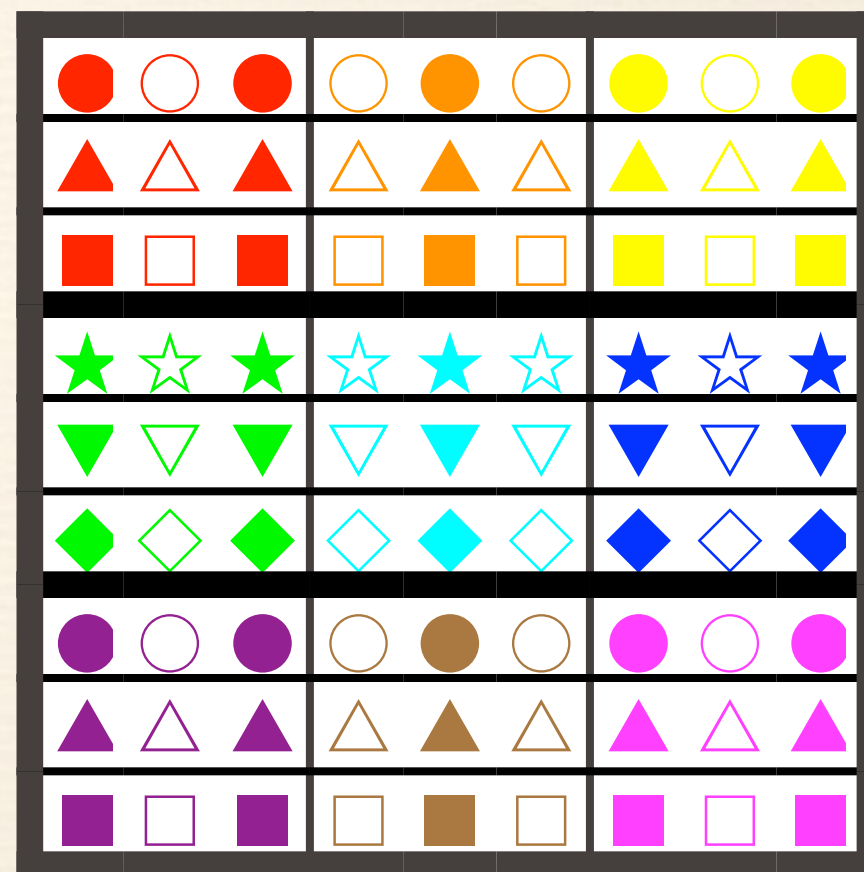
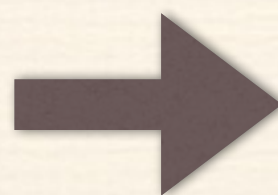
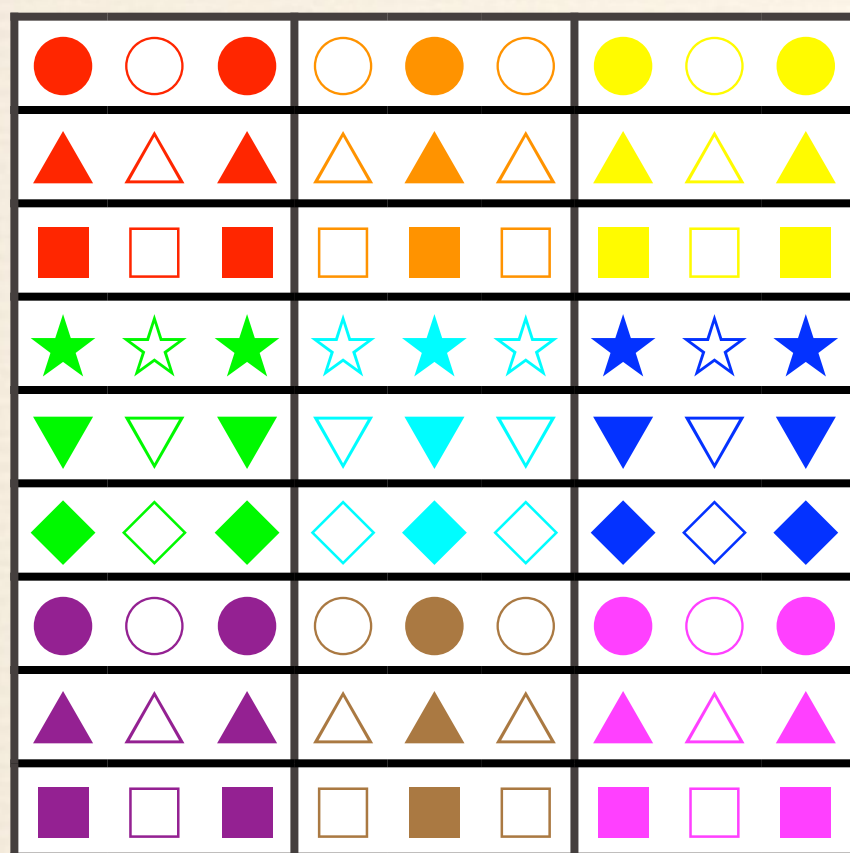
## ブロック





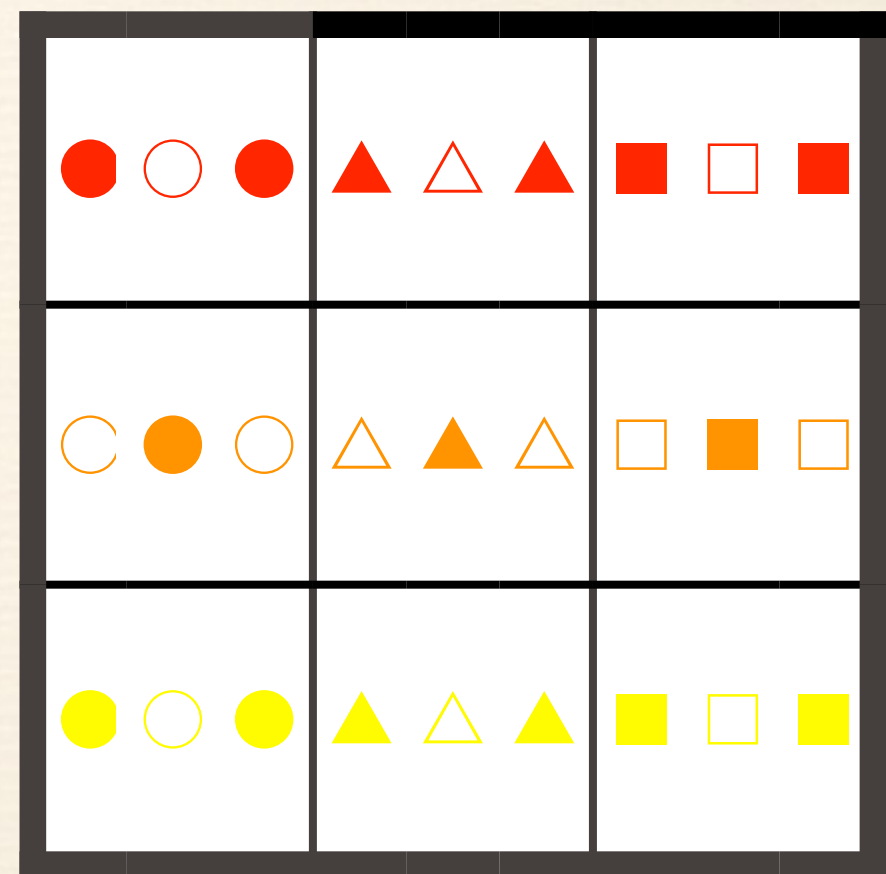
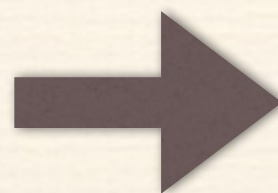
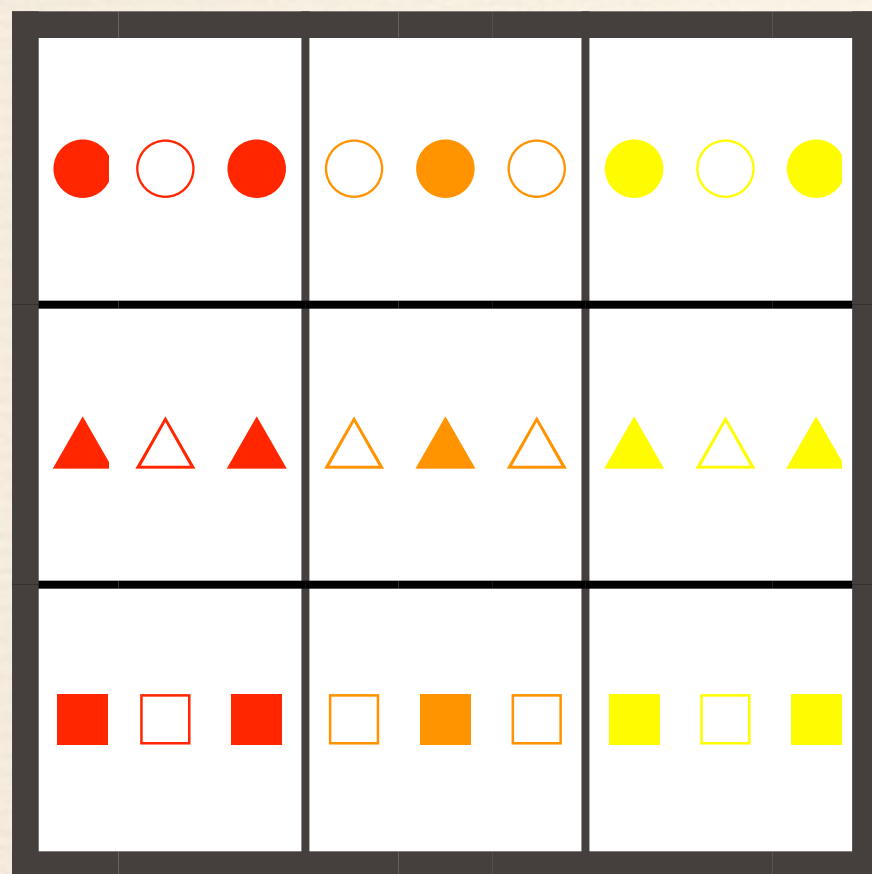
# 行、列、ブロック

## ブロック



# 行、列、ブロック

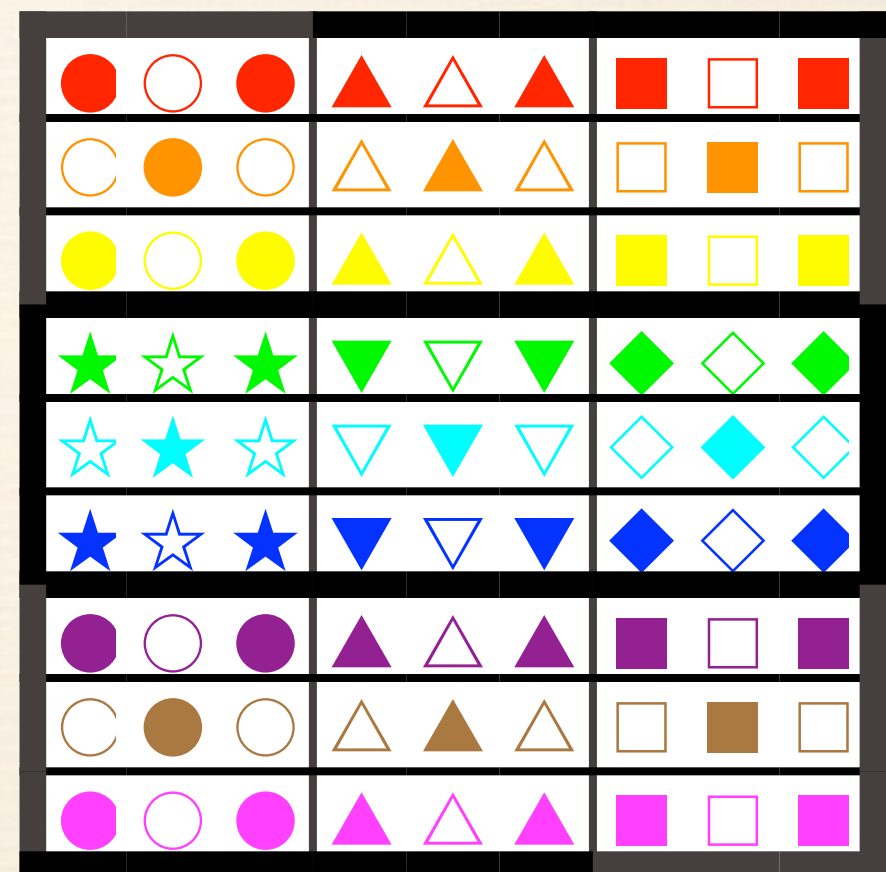
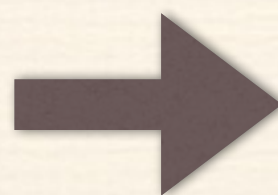
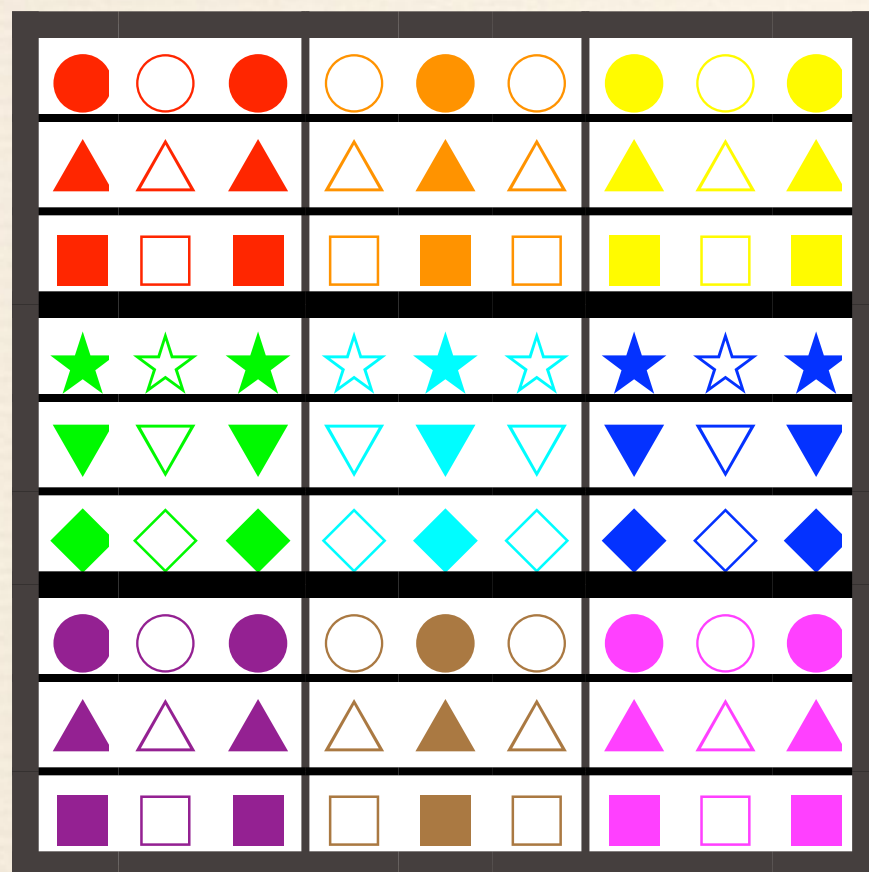
## ブロック





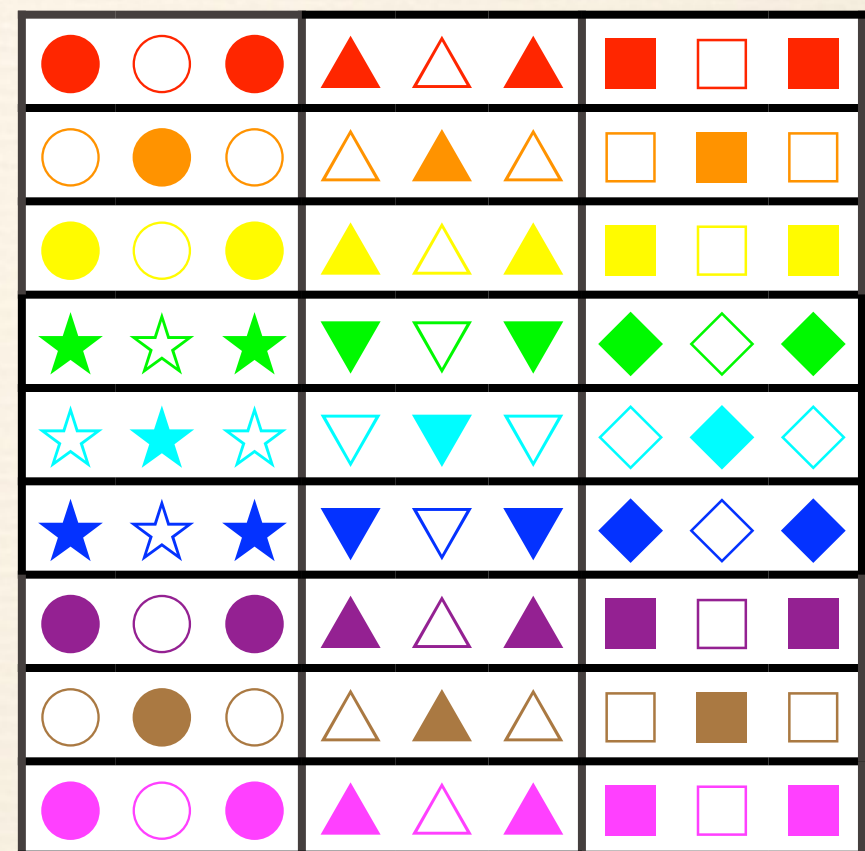
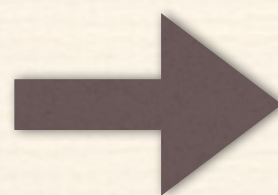
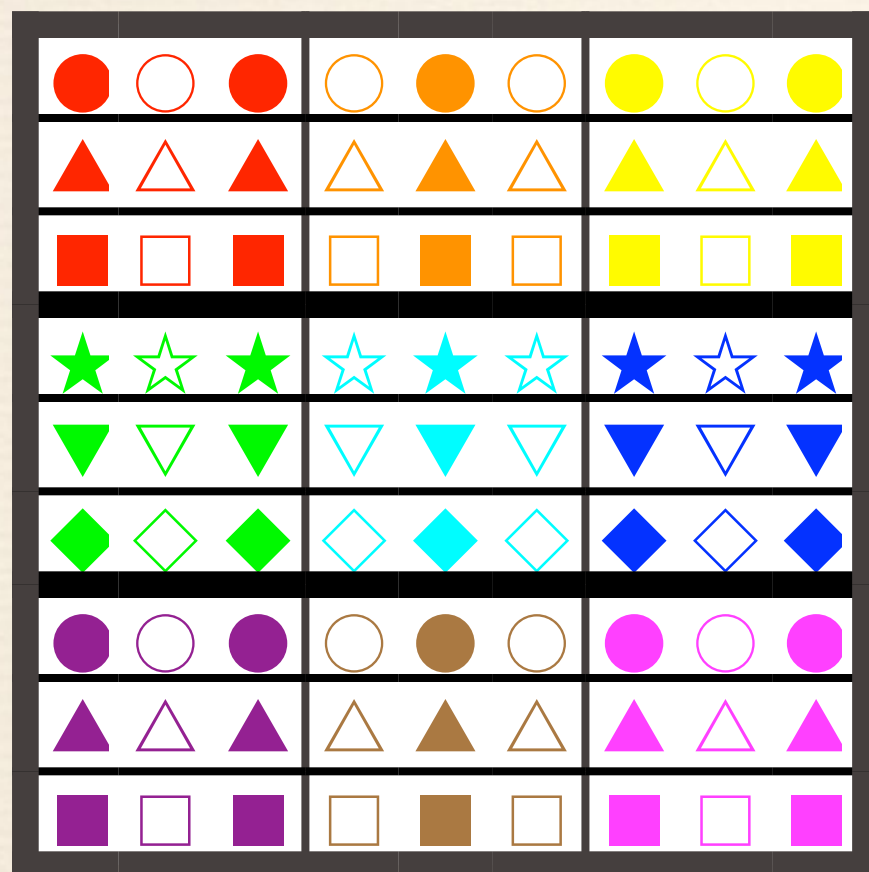
# 行、列、ブロック

## ブロック



# 行、列、ブロック

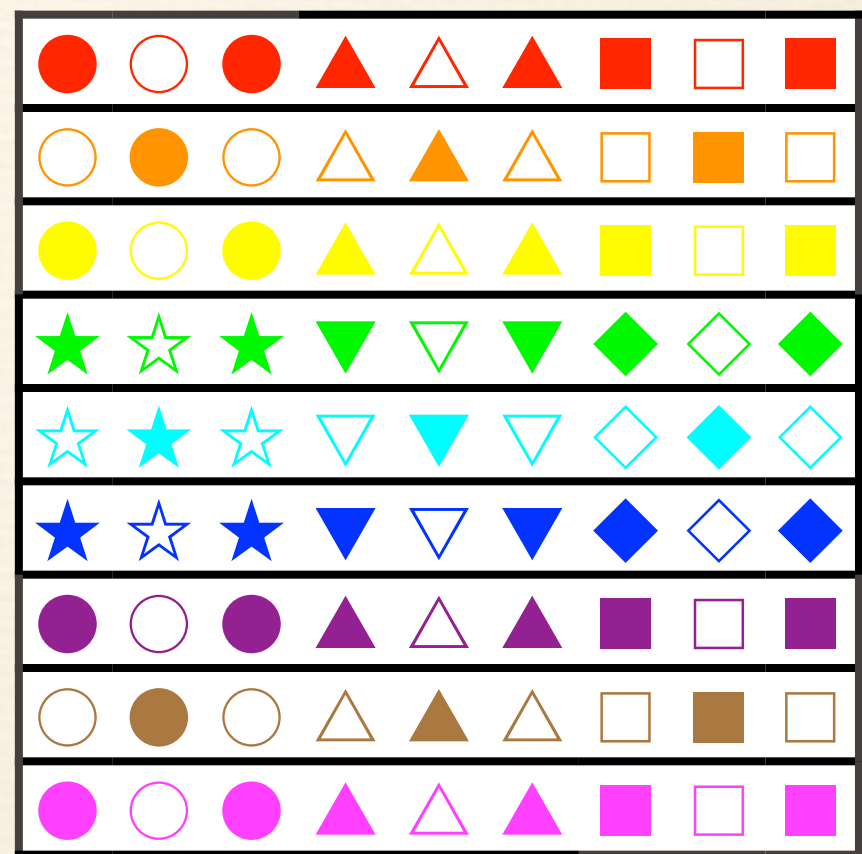
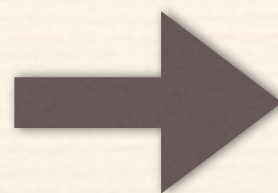
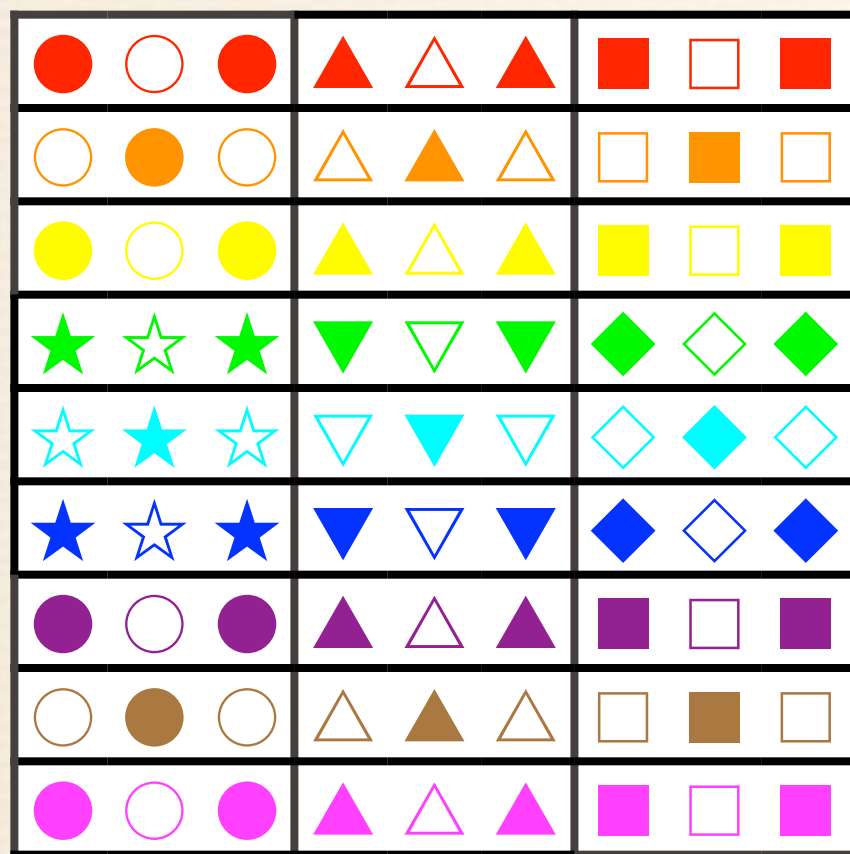
## ブロック





# 行、列、ブロック

## ブロック



# 候補の絞り込み



# 絞り込み

- ❖ 全探索 →  $O(n^{n^2})$
- ❖ 重複を除いていく

# 行の重複排除

❖ 同じ列に現れる数を候補から消す

1	2	3
4	5	6
7	8	9

remove



1	2	3
		6
	8	9



# 行の重複排除

- ❖ 同じ列に現れる数を候補から消す

1 2 3	1 2 3	4	1 2 3	1 2 3	5	7	1 2 3	1 2 3
4 5 6	4 5 6		4 5 6	4 5 6			4 5 6	4 5 6
7 8 9	7 8 9		7 8 9	7 8 9			7 8 9	7 8 9



map remove

1 2 3	1 2 3	4	1 2 3	1 2 3	5	7	1 2 3	1 2 3
6	6		6	6			6	6
8 9	8 9		8 9	8 9			8 9	8 9

# 重複排除

❖ 列、ブロックはcolsやboxsで変換、逆変換



# 絞り込み

- ❖ 重複排除を繰り返す
- ❖ 候補の数が $m$ 個に絞り込めた  $\rightarrow O(mn^3+n^6)$
- ❖  $m$ が小さいとき全探索 $O(n^{n^2})$ から圧倒的改善

# 候補の列举



# 候補の列举

- ❖ これまでは候補を全て同時に列举していた
- ❖ 1マスごとに候補列举→絞り込みを繰り返す

# 候補の列挙

- ❖ 候補が少ないマスから列挙していく
- ❖  $O(mn^3+n^5) \rightarrow O(n^6)$



# まとめ

ソースコード

joiss.ぴょんぴょん.net



ご注文はHaskellですか?



akouryy