

コーナー 2: 以下の式が複雑すぎてわからない

Q. 以下の式が複雑すぎてわからない。

```
Nest[ArrayFlatten[# /. {  
  1 → {{1, 1, 1}, {1, 0, 1}, {1, 1, 1}},  
  0 → {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}] &, 1, 3];
```

1. ルールの集合を明示的に定義し、

```
rule = {  
  1 → {{1, 1, 1}, {1, 0, 1}, {1, 1, 1}},  
  0 → {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}};
```

原式をruleで置き換えてみる：

```
In[ ]:= Nest[ArrayFlatten[# /. rule] &, 1, 3];
```

2. 未定義関数fを使い、**ArrayFlatten[#/.rule]&**をfで置き換えてみる。

```
In[ ]:= Nest[f, 1, 3]
```

Out[]:=

```
f[f[f[1]]]
```

1にfを適用 (f[1]) し、その結果に再度、fを適用する (f[f[1]])、以下同様。

3. 今度はfの代わりにArrayFlatten[#/.rule]&を使う。ただし、そのままでは評価（実行）が動いてしまうので、形は似てるが未評価シンボルを使う。たとえば、

ArrayFlatten1[#/.rule1]&

以下、未定義なのでエラーが出力されるが、「展開形式」は見える。

```
In[ ]:= Nest[ArrayFlatten1[# /. rule1] &, 1, 3]
```

⋮ **ReplaceAll:** {rule1}は置換規則のリストでも有効なディスパッチ表でもないため、置換には使用できません. ⓘ

⋮ **ReplaceAll:** {rule1}は置換規則のリストでも有効なディスパッチ表でもないため、置換には使用できません. ⓘ

⋮ **ReplaceAll:** {rule1}は置換規則のリストでも有効なディスパッチ表でもないため、置換には使用できません. ⓘ

⋮ **General:** この計算中に、ReplaceAll::repsのこれ以上の出力は表示されません. ⓘ

Out[]:=

```
ArrayFlatten1[ArrayFlatten1[ArrayFlatten1[1 /. rule1] /. rule1] /. rule1]
```

4. 一番奥底の **ArrayFlatten1[1/.rule1]** をArrayFlatten[1/.rule]で置き換えて、動きを知る。

```
In[ ]:= ArrayFlatten[1 /. rule]
```

Out[]:=

```
{{1, 1, 1}, {1, 0, 1}, {1, 1, 1}}
```

未評価の関数 f を使い、展開の姿を理解する

Function &

```
In[ ]:= Function[a, a + 2] [3]
Out[ ]=
5
```

```
In[ ]:= # + 2 & [3]
Out[ ]=
5
```

ReplaceAll /.

```
In[54]:= f[1, 2] /. f → Plus
Plus[1, 2]
Out[54]=
3
Out[55]=
3
```

```
In[ ]:= a /. {a → a b}
Out[ ]=
a b
```

@, @@, @@@, /@

```
In[ ]:= f@{1, 2}
Out[ ]=
f[{1, 2}]
```

```
In[56]:= f@@{1, 2}
Apply[f, {1, 2}]
Out[56]=
f[1, 2]
Out[57]=
f[1, 2]
```

```
In[58]:= f@@@{{1}, {2}}
Map[Apply[f, #] &, {{1}, {2}}]
Out[58]=
{f[1], f[2]}
Out[59]=
{f[1], f[2]}
```

```
In[61]:= f /@ {1, 2}
Map[f, {1, 2}]
```

```
Out[61]= {f[1], f[2]}
```

```
Out[62]= {f[1], f[2]}
```

Nest

```
In[*]:= Nest[f, a, 3]
```

```
Out[*]= f[f[f[a]]]
```

```
In[*]:= Nest[# /. {a -> a b} &, a, 3]
```

```
Out[*]= a b3
```

```
In[*]:= Nest[(# /. {a -> a b}) &, a, 3]
```

```
Out[*]= a b3
```

```
In[*]:= NestList[f, a, 3]
```

```
Out[*]= {a, f[a], f[f[a]], f[f[f[a]]]}
```

Fold

```
In[*]:= Fold[f, a, {1, 2, 3, 4}]
```

```
Out[*]= f[f[f[f[a, 1], 2], 3], 4]
```