

JavaScript

for文(ループ処理) 応用

11 時間目

for文のネスト

for文でも、if文同様に
ネスト（入れ子）を作ることが可能です。

補足

ネスト（入れ子）とは、if文の中にif文があるプログラム、for文の中にfor文があるプログラムなどの事です。

for文と剰余のネスト

1～10までの数字で偶数のみを表示

script.js(JavaScriptファイル)

```
for(var i=1; i<=10; i++) {  
  if(i % 2 == 0){  
    document.write(i+"<br>");  
  }  
}
```

1～10回目までループをまわし、
さらに、iが2で割り切れる場合のみ
document.writeで、iを表示。

← → ↺ 🏠

2
4
6
8
10

補足

$i \% 2 == 0$ は、“iを2で割った際に余りが0になる”という意味。
つまり、iが偶数という意味。

for文と剰余のネスト 解説



for文とforのネスト

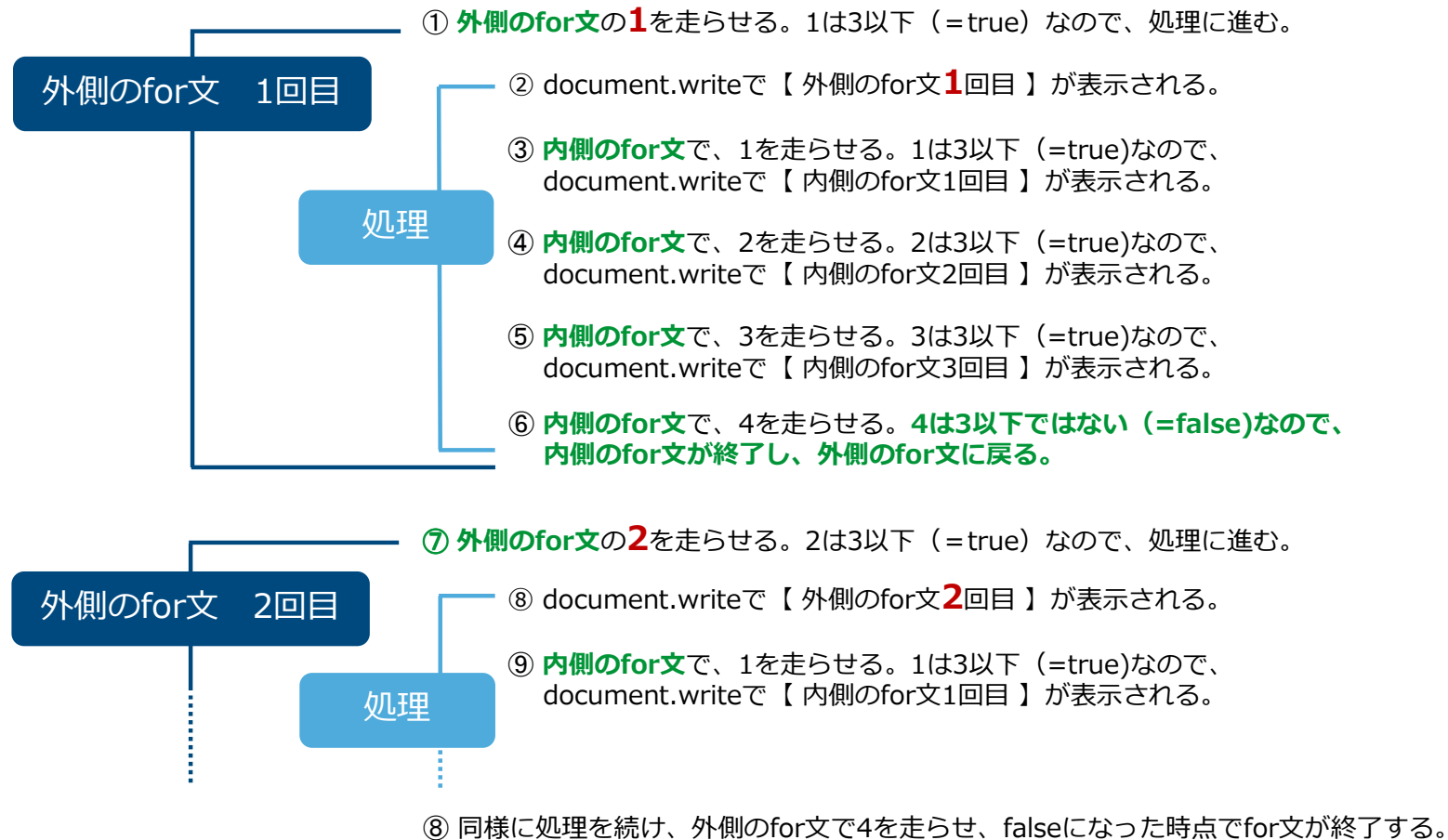
script.js(JavaScriptファイル)

```
for(var i=1; i<=3; i++) {  
  document.write(" ■外側のfor文"+i+"回目<br>" );  
  for(var a=1; a<=3; a++){  
    document.write("内側のfor文"+a+"回目<br>");  
  }  
}
```



■外側のfor文1回目
内側のfor文1回目
内側のfor文2回目
内側のfor文3回目
■外側のfor文2回目
内側のfor文1回目
内側のfor文2回目
内側のfor文3回目
■外側のfor文3回目
内側のfor文1回目
内側のfor文2回目
内側のfor文3回目

for文とfor文のネスト 解説



for文とfor文のネスト 解説

解説

前ページと比較をしてみましょう。for文の{}の場所が異なります。このfor文は、ネストになっていないのに注目してください。

script.js(JavaScriptファイル)

```
for(var i=1; i<=3; i++){  
  document.write("■ 外側のfor文"+i+"回目<br>");  
}  
  
for(var a=1; a<=3; a++){  
  document.write("内側のfor文"+a+"回目<br>");  
}
```

← → ↺ 🏠

■ 外側のfor文1回目
■ 外側のfor文2回目
■ 外側のfor文3回目
内側のfor文1回目
内側のfor文2回目
内側のfor文3回目

九九を表示する

九九の答えを1の段、2の段・・・ごとに表示し、
最後に →●の段と表示する。

script.js(JavaScriptファイル)

```
for(var i =1; i <= 9; i ++){  
  for(var a =1; a<= 9; a++){  
    document.write( i * a );  
  }  
  document.write("→"+i+"の段<br>");  
}
```

← → ↺ 🏠

123456789→1の段
24681012141618→2の段
369121518212427→3の段
4812162024283236→4の段
51015202530354045→5の段
61218243036424854→6の段
71421283542495663→7の段
81624324048566472→8の段
91827364554637281→9の段

「九九を表示する」の解説

