

# カレンダー



# カレンダー

## 仕様

- 初期表示は「今月」のカレンダーが表示されます。
- 「次の月」をクリックすると、次の月のカレンダーが表示されます。  
(順に、次の月へ、次の月へ・・・)
- 「前の月」をクリックすると、前の月のカレンダーが表示されます。  
(順に、前の月へ、前の月へ・・・)
- 「今月」をクリックすると、今月のカレンダーが表示されます。

# カレンダー

ヒント：PHPのDateTimeクラスを使いましょう。

<https://www.php.net/manual/ja/book.datetime.php>

現在の日付を取得するには、DateTimeクラスのインスタンスを作ります。

```
$date = new DateTime();
```

「\$date」の中には、現在時刻の「DateTimeクラス」のオブジェクトが入っています。  
「yyyy/mm/dd」の形式で出力するには

```
echo $date->format('Y/m/d');
```

とします。

DateTimeクラスのインスタンスを作るときに、「日付形式」の文字列を引数に指定すると、その日時にインスタンスを作ることができます。「2019年8月1日」の日付にしたいときは、

```
$date = new DateTime('2018/08/01');
```

とします。

```
$weekDay = $date->format('w');
```

とすれば、「ついたち」が何曜日かも分かります。

# カレンダー

## ヒント：日付のフォーマット

こちらのページを参照してください。

<https://www.php.net/manual/ja/function.date.php>

例)

```
$date = new DateTime();
```

```
// 年
```

```
echo $date->format('Y');
```

```
// 月
```

```
echo $date->format('m');
```

```
// 日
```

```
echo $date->format('d');
```

```
// 日付形式
```

```
echo $date->format('Y/m/d');
```

```
// 当日の曜日 (0 (日曜)から 6 (土曜))
```

```
echo $date->format('w');
```

# カレンダー

日付の計算

```
// 2019年8月1日のDateTimeオブジェクトを作成します。
$date = new DateTime('2019/08/01');

// 2019年8月1日のDateTimeオブジェクトに1か月足します。
// 引数にはDateIntervalオブジェクトを指定します。
$date->add(new DateInterval('P1M'));

// 2019年9月1日と表示されます。DateTimeオブジェクトの日付は2019年9月1日になっています。
echo $date->format('Y/m/d');

// 2019年9月1日になったDateTimeオブジェクトから1か月引きます。
// 引数にはDateIntervalオブジェクトを指定します。
$date->sub(new DateInterval('P1M'));

// 2019年9月1日と表示されます。DateTimeオブジェクトの日付は2019年9月1日に戻っています。
echo $date->format('Y/m/d');
```

# カレンダー

DateIntervalクラス

<https://www.php.net/manual/ja/class.dateinterval.php>

DateIntervalクラスは、日時の期間を作成するクラスで、インスタンスを作るとき、引数に「期間」を示す文字列をいれます。

「期間の文字列」は必ず「P」から始まり、

1か月=> 'P1M'

1日=> 'P1D'

と指定します。

// 1か月の期間のDateIntervalオブジェクト

```
$interval = new DateInterval('P1M');
```

// 1日の期間のDateIntervalオブジェクト

```
$interval = new DateInterval('P1D');
```

1週間 = 7日分ループ



当月の初め（ついたち）が  
何曜日から始まるか？

当月の日数が何日あるか  
=>どこで日にちの表示が終わるか？



# カレンダーの考え方 (2)

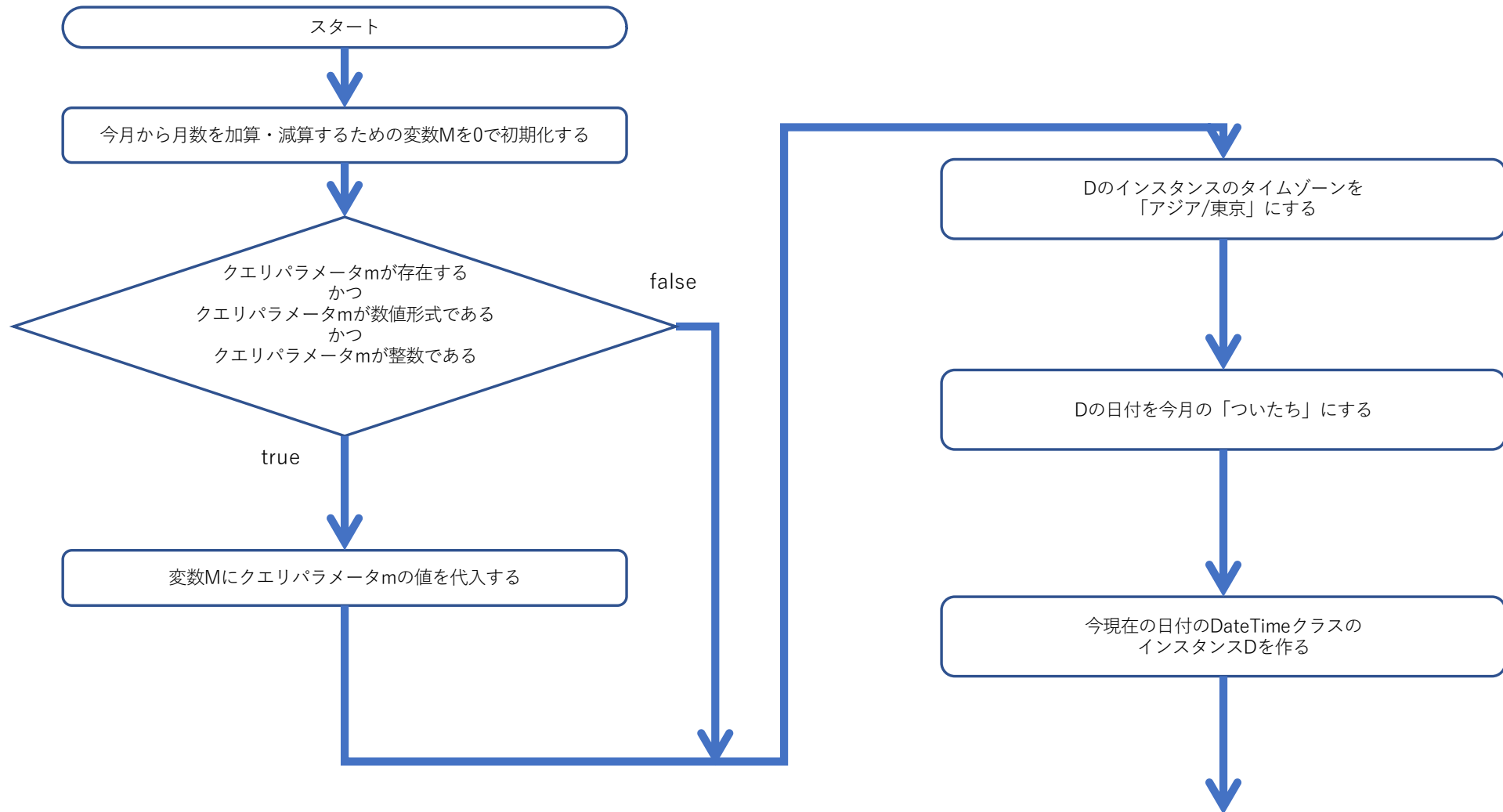
		②							
		日	月	火	水	木	金	土	
		0	1	2	3	4	5	6	
①	0	<tr>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	</tr>
	1	<tr>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	</tr>
	2	<tr>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	</tr>
	3	<tr>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	</tr>
	4	<tr>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	</tr>
	5	<tr>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	<td></td>	</tr>

① 当月にある週分繰り返し

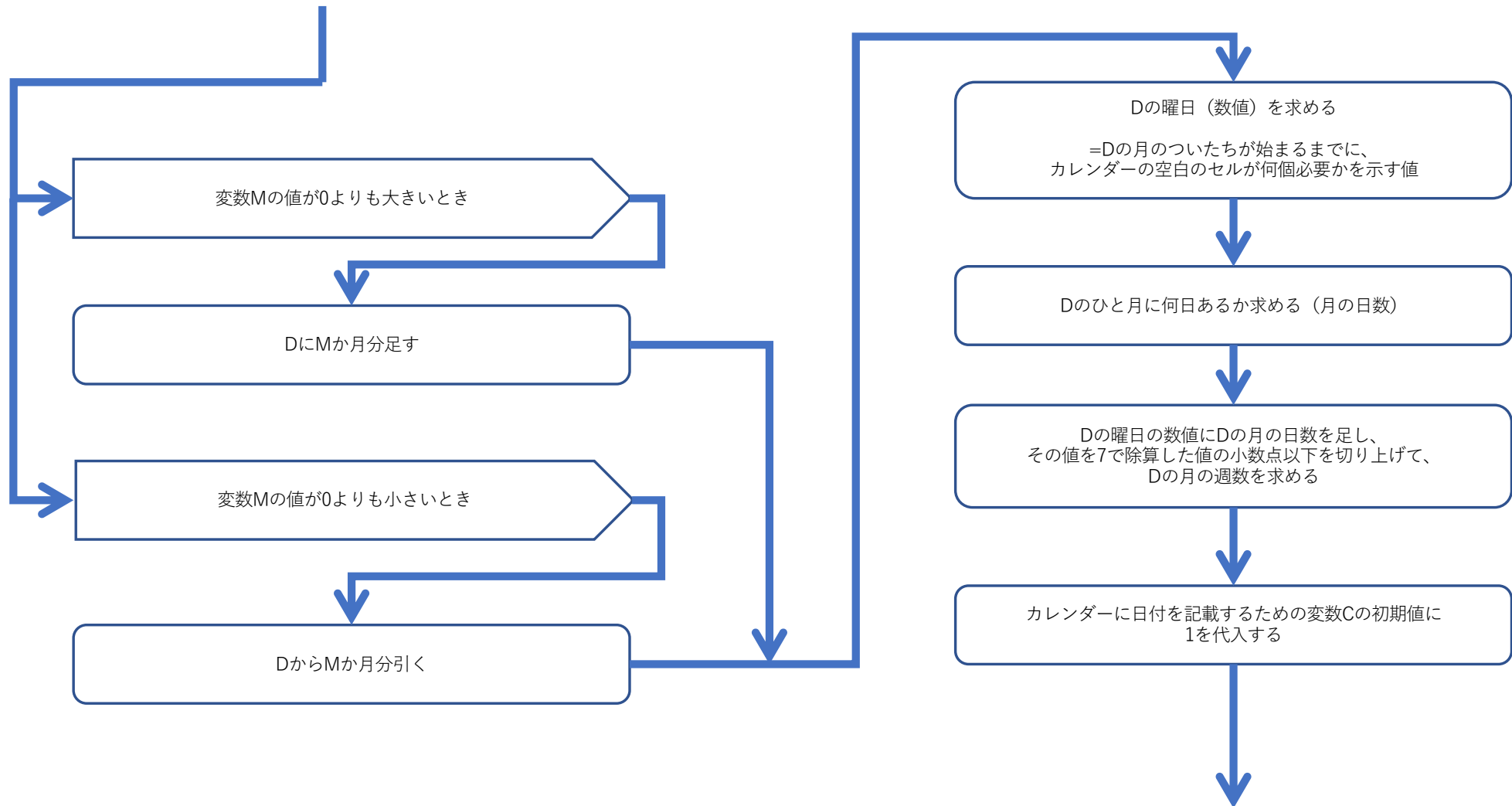
② 一週間（7日分）繰り返し



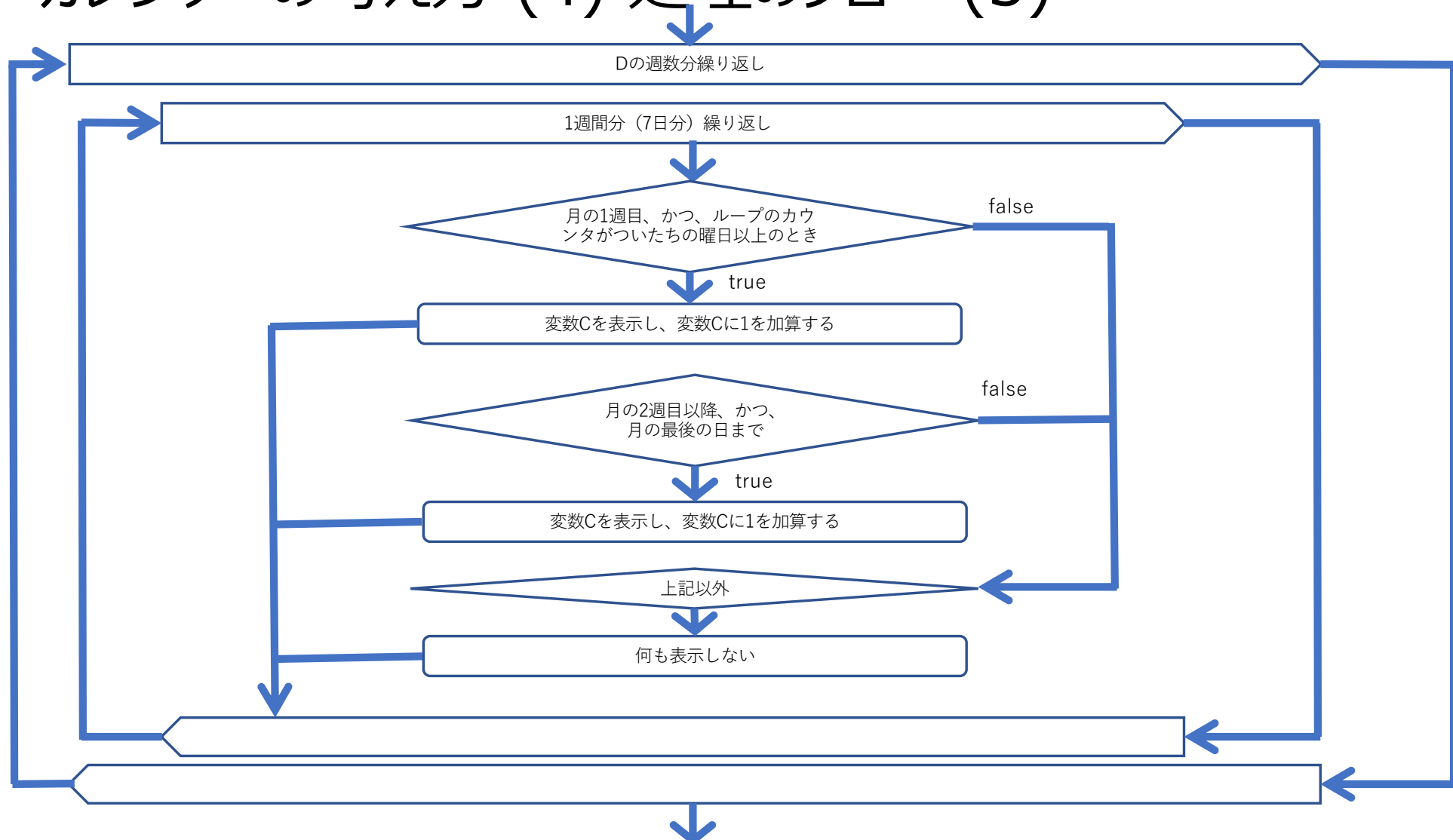
# カレンダーの考え方 (4) 処理のフロー (1)



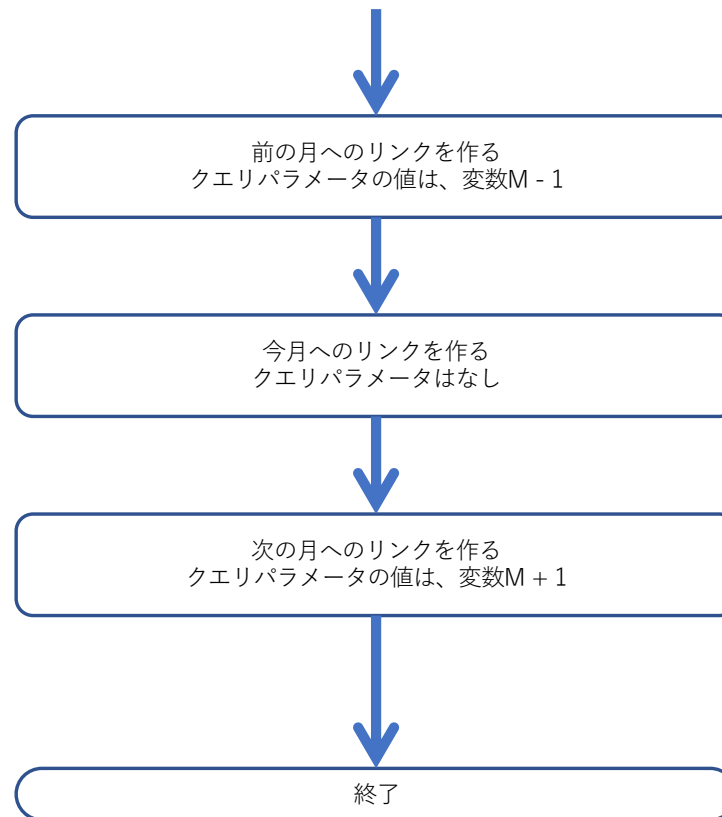
## カレンダーの考え方 (4) 処理のフロー (2)



## カレンダーの考え方 (4) 処理のフロー (3)



## カレンダーの考え方 (4) 処理のフロー (4)



# カレンダーのその他のヒント

表示がおかしいときは、ブラウザに表示されているHTMLのソースを見てみましょう。

```
1 <!DOCTYPE html>
2 <html lang="jp">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>カレンダー</title>
9   <link rel="stylesheet" href="/css/normalize.css">
10  <link rel="stylesheet" href="/css/main.css">
11 </head>
12
13 <body>
14   <div class="container">
15
16     <h1>カレンダー</h1>
17
18     <table>
19       <caption>2020年1月</caption>
20       <tr>
21         <th>日</th>
22         <th>月</th>
23         <th>火</th>
24         <th>水</th>
25         <th>木</th>
26         <th>金</th>
27         <th>土</th>
28       </tr>
29       <!-- 当月にある週数分繰り返し -->
30       <tr>
31         <!-- 一週間の日数分 (7日分) 繰り返し -->
32         <td>
33           </td>
34         <td>
35           </td>
36         <td>
37           </td>
38         <td>
39           1
40         <td>
41           2
42         <td>
43           3
44         <td>
45           4
46       </tr>
47       <tr>
48         <!-- 一週間の日数分 (7日分) 繰り返し -->
```

# カレンダーのその他のヒント

クエリストリング（クエリパラメータ）

<http://localhost/calendar/?month=1>

?month=1

の部分のことを「クエリストリング」とか「クエリパラメータ」と言います。

\$\_GETというグローバル変数でクエリストリングの値を取得します。

```
echo $_GET['month'];           // 「1」と表示される
```

# カレンダーのその他のヒント

変数が定義されているかどうか調べるには **isset()** を使います。

<https://www.php.net/manual/ja/function.isset.php>

```
if (isset($var)) {  
    // $varという変数が定義されているときの処理  
} else {  
    // $varという変数が定義されていないときの処理  
}
```

配列の要素が定義されているかどうか調べることもできます。

```
if (isset($a['key'])) {  
    // 配列$aにkeyという要素があるときの処理  
} else {  
    //配列$aにkeyという要素がないときの処理  
}
```

※\$\_GETも\$\_POSTも配列です！

# カレンダーのその他のヒント

Math関数

小数点切り上げ→ ceil()

<https://www.php.net/manual/ja/function.ceil.php>

小数点切り下げ→ floor()

<https://www.php.net/manual/ja/function.floor.php>

四捨五入→ round()

<https://www.php.net/manual/ja/function.round.php>