

The `Calendar` class is an abstract class that provides methods for converting between a specific instant in time and a set of calendar fields such as YEAR, MONTH, DAY_OF_MONTH, HOUR, and so on, and for manipulating the calendar fields, such as getting the date of the next week.

You are given a date. You just need to write the method, `getDay`, which returns the day on that date. To simplify your task, we have provided a portion of the code in the editor.

Example
`month = 8`
`day = 14`
`year = 2017`

The method should return `MONDAY` as the day on that date.

AUGUST 2017						
SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

```
6 import java.util.concurrent.*;
7 import java.util.regex.*;
8
9 class Result {
10
11     /*
12      * Complete the 'findDay' function below.
13      *
14      * The function is expected to return a STRING.
15      * The function accepts following parameters:
16      * 1. INTEGER month
17      * 2. INTEGER day
18      * 3. INTEGER year
19      */
20
21     public static String findDay(int month, int day, int year) {
22
23         Calendar cal = Calendar.getInstance();
24         cal.set(Calendar.MONTH, month-1);
25         cal.set(Calendar.DAY_OF_MONTH, day);
26         cal.set(Calendar.YEAR, year);
```

Upload Code as File




☐ Test against custom input

Run Code

Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?



Next Challenge

Problem

Submissions

Leaderboard

Discussions

Calendar Module

The calendar module allows you to output calendars and provides additional useful functions for them.

```
class calendar.TextCalendar([firstweekday])
```

This class can be used to generate plain text calendars.

Sample Code

```
>>> import calendar
>>>
>>> print calendar.TextCalendar(firstweekday=6).formatyear(2015)
2015
```

January							February							March						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2	1	2	3	4	5	6	7	1	2	3	4			
4	5	6	7	8	9	10	8	9	10	11	12	13	14	8	9	10	11	1		
11	12	13	14	15	16	17	15	16	17	18	19	20	21	15	16	17	18	1		
18	19	20	21	22	23	24	22	23	24	25	26	27	28	22	23	24	25	2		
25	26	27	28	29	30	31	29	30	31					29	30	31				

April							May							June						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2						1	2							
5	6	7	8	9	10	11	3	4	5	6	7	8	9	7	8	9	10	1		
12	13	14	15	16	17	18	10	11	12	13	14	15	16	14	15	16	17	1		
19	20	21	22	23	24	25	17	18	19	20	21	22	23	21	22	23	24	2		
26	27	28	29	30			24	25	26	27	28	29	30	28	29	30				
							31													

```
5 print ('{}'.format(calendar.day_name[date.weekday()]).upper())
6
```

Line: 6 Col: 1

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

[f](#) [t](#) [in](#)

Next Challenge