

## 一周中的第几天

给你一个日期，请你设计一个算法来判断它是对应一周中的哪一天。

输入为三个整数：day、month 和 year，分别表示日、月、年。

您返回的结果必须是这几个值中的一个 {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}。

示例 1:

输入: day = 31, month = 8, year = 2019

输出: "Saturday"

示例 2:

输入: day = 18, month = 7, year = 1999

输出: "Sunday"

示例 3:

输入: day = 15, month = 8, year = 1993

输出: "Sunday"

**方法一：导入 datetime python 接口函数调用直接解决**

```
import datetime
class Solution:
    def dayOfTheWeek(self, day: int, month: int, year: int) -> str:
        #根据 python 有关时间 API 进行转化
        #控制输入时间格式 返回一周中第几天 即可得到周几
        w_list = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"] # 星期列表
        d = str(year) + str("{:0>2d}".format(month)) + str("{:0>2d}".format(day)) # 格式化日期，月和日必须是两位的
        return w_list[datetime.datetime.strptime(d, "%Y%m%d").weekday()]
```

**方法二：根据当天日期星期几 计算出与给定日期的天数差 对7取余即可得到星期几**

```
#根据当天日期星期几 计算出与给定日期的偏差
#已知今天 2022 年 1 月 9 日 是星期天，只需要知道输入的日期与今天的天数差，再对七取余就可以算出该日期是星期几。
ANS = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]
```

```

DAYS = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
class Solution:
    def dayOfTheWeek(self, day: int, month: int, year: int) -> str:
        d=day
        y=year
        m=month
        # 2022.1.9 星期天
        ans = 0
        # 天数的偏移量
        ans = (ans + d - 9) % 7
        # 计算年的偏差量
        if y < 2022:
            for i in range(y, 2022):
                ans = (ans - (366 if not i % 4 and (i % 100 or not i % 400)
else 365)) % 7
        else:
            for i in range(2022, y):
                ans = (ans + (366 if not i % 4 and (i % 100 or not i % 400)
else 365)) % 7
        # 计算月的偏差量
        for i in range(m - 1):
            ans = (ans + DAYS[i]) % 7
            if i == 1 and not y % 4 and (y % 100 or not y % 400):#判断闰年 2
月份天数 29 天
                ans = (ans + 1) % 7
        return ANS[ans]

```