

# H1 8 字符串转换整数

## H2 题目描述

请你来实现一个 `atoi` 函数，使其能将字符串转换成整数。

首先，该函数会根据需要丢弃无用的开头空格字符，直到寻找到第一个非空格的字符为止。

当我们寻找到的第一个非空字符为正或者负号时，则将该符号与之后面尽可能多的连续数字组合起来，作为该整数的正负号；假如第一个非空字符是数字，则直接将其与之后连续的数字字符组合起来，形成整数。

该字符串除了有效的整数部分之后也可能会存在多余的字符，这些字符可以被忽略，它们对于函数不应该造成影响。

注意：假如该字符串中的第一个非空格字符不是一个有效整数字符、字符串为空或字符串仅包含空白字符时，则你的函数不需要进行转换。

在任何情况下，若函数不能进行有效的转换时，请返回 0。

说明：

假设我们的环境只能存储 32 位大小的有符号整数，那么其数值范围为  $[-2^{31}, 2^{31} - 1]$ 。如果数值超过这个范围，qing返回 `INT_MAX` ( $2^{31} - 1$ ) 或 `INT_MIN` ( $-2^{31}$ )。

示例 1:

输入: "42"  
输出: 42

示例 2:

输入: " -42"  
输出: -42  
解释: 第一个非空白字符为 '-', 它是一个负号。  
我们尽可能将负号与后面所有连续出现的数字组合起来，最后得到 -42 。

示例 3:

输入: "4193 with words"  
输出: 4193  
解释: 转换截止于数字 '3' , 因为它的下一个字符不为数字。

示例 4:

输入: "words and 987"  
输出: 0  
解释: 第一个非空字符是 'w', 但它不是数字或正、负号。  
因此无法执行有效的转换。

示例 5:

输入: "-91283472332"

输出: -2147483648

解释: 数字 "-91283472332" 超过 32 位有符号整数范围。  
因此返回 INT\_MIN (-231) 。

## H2 代码

```
class Solution:
    def myAtoi(self, str: str) -> int:
        str = str.strip()

        negative = False
        if str and str[0] == '-':
            negative = True
        if str and (str[0] == '+' or str[0] == '-'):
            str = str[1:]
        if not str:
            return 0

        digits = {i for i in '0123456789'}
        result = 0
        for c in str:
            if c not in digits:
                break
            result = result*10 + int(c)

        if negative:
            result = -result

        result = max(min(result, 2 ** 31 - 1), -2 ** 31)
        return result
```

成功 [显示详情](#) >

执行用时 : **56 ms**, 在String to Integer (atoi)的Python3提交中击败了**93.42%** 的用户

内存消耗 : **13 MB**, 在String to Integer (atoi)的Python3提交中击败了**98.07%** 的用户

进行下一个挑战 :

[有效数字](#)

炫耀一下:    

提交时间	状态	执行用时	内存消耗	语言
几秒前	通过	56 ms	13 MB	python3
12 分钟前	解答错误	N/A	N/A	python3
14 天前	执行出错	N/A	N/A	python3