# 题目

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。接下来的转化规则如下：

如果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字符组合起来，形成一个有符号整数。

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，形成一个整数。

该字符串在有效的整数部分之后也可能会存在多余的字符，那么这些字符可以被忽略，它们对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，即无法进行有效转换。

在任何情况下，若函数不能进行有效的转换时，请返回 0 。

**提示：**

本题中的空白字符只包括空格字符 ' ' 。

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[−231,  231−1]。如果数值超过这个范围，请返回 INT\_MAX (231 −1) 或 INT\_MIN (−231) 。

**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) 。

# 分析

**思路：**

利用flag存储+-，利用start确定是否已经开始，若之后再出现不是数字则break。

**代码：**

class Solution {

public:

int myAtoi(string str) {

long long result(0);

int str\_size=str.size();

int flag=1;//存储+-标志位

bool start=false;//数字读取是否已经开始

//若之后再出现不是数字的字符，则break

for(int i=0;i<str\_size;i++){

if(str[i]==' '&&start==false){

continue;

}

if(str[i]=='-'&&start==false){

flag=-1;

start=true;

continue;

}

if(str[i]=='+'&&start==false){

flag=1;

start=true;

continue;

}

if(str[i]>='0'&&str[i]<='9'){

start=true;

result=result\*10+str[i]-48;

//判断溢出

if(flag\*result>INT\_MAX)

return INT\_MAX;

if(flag\*result<INT\_MIN)

return INT\_MIN;

continue;

}

else{

break;

}

}

return flag\*result;

}

};