

8. String to Integer (atoi)

Implement `atoi` which converts a string to an integer.

The function first discards as many whitespace characters as necessary until the first non-whitespace character is found. Then, starting from this character, takes an optional initial `plus` or `minus` sign followed by as many numerical digits as possible, and interprets them as a numerical value.

The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in `str` is not a valid integral number, or if no such sequence exists because either `str` is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned.

Example 1:

```
Input: "42"  
Output: 42
```

Example 2:

```
Input: "   -42"  
Output: -42  
Explanation: The first non-whitespace character is '-', which is the minus sign.  
              Then take as many numerical digits as possible, which gets 42.
```

Example 3:

```
Input: "4193 with words"  
Output: 4193  
Explanation: Conversion stops at digit '3' as the next character is not a numerical digit.
```

- 先判斷是否有空白, '-' or '+', 其他除了數字以外的都是無效字元 · 返回 0
- 每次累加確定是否已經溢位
- 負值取2的補數

```
1  int myAtoi(char *str)
2  {
3      bool neg = 0;
4      long long int res = 0;
5
6      while (*str == ' ')
7          str += 1;
8
9      if (*str == '-') {
10         neg = 1;
11         str += 1;
12     } else if (*str == '+') {
13         str += 1;
14     }
15
16     while (*str != 0) {
17         if ((*str > '9') || (*str < '0'))
18             break;
19
20         if ((res = ((res * 10) + (*str++ - '0'))) > INT_MAX)
21             return (neg) ? INT_MIN : INT_MAX;
22     }
23
24     return (neg) ? (~res + 1) : res;
25 }
```