

字符串:

Java String x = "abbc"

☆ 每次改变 String 里的内容的话,都是创建了一个新的 String

Java / C# / JavaScript / Go : String are immutable  
不可变的

Ruby / PHP / C : String are mutable  
可变的

## 遍历字符串

- Python:

```
for ch in "abbc":  
    print(ch)
```

- Java:

```
String x = "abbc";  
for (int i = 0; i < x.size(); ++i) {  
    char ch = x.charAt(i);  
}  
for (ch in x.toCharArray()) {  
    System.out.println(ch);  
}
```

用 i 下标来搞,然后调 x.charAt

也可以把它转换成 Char Array 之后,然后遍历里面所有值,这样就可以把个 String 里面的所有字符一一打出来

## 字符串比较

Java:

```
String x = new String("abb");
```

```
String y = new String("abb");
```

```
x == y --> false
```

```
✓ x.equals(y) --> true
```

```
✓ x.equalsIgnoreCase(y) --> true
```

忽略大小写

两个不同的变量,它指向内存中的不同地址

(虽然它们字符串的内容都一样,但是它们是不同的内存上的地址)

→ 比较它们的指针,比较它们的 reference 地址,而不是比较字符串里的内容

Q1、Q3

sign: 符号 sign = 1 : 正负号

total: 绝对值

## Atoi

字符串转整数

```
public int myAtoi(String str) {
    int index = 0, sign = 1, total = 0;
    //1. Empty string
    if(str.length() == 0) return 0;

    //2. Remove Spaces
    while(str.charAt(index) == ' ' && index < str.length())
        index++;

    //3. Handle signs
    if(str.charAt(index) == '+' || str.charAt(index) == '-'){
        sign = str.charAt(index) == '+' ? 1 : -1;
        index++;
    }

    //4. Convert number and avoid overflow
    while(index < str.length()){
        int digit = str.charAt(index) - '0';
        if(digit < 0 || digit > 9) break;

        //check if total will be overflow after 10 times and add digit
        if(Integer.MAX_VALUE/10 < total ||
           Integer.MAX_VALUE/10 == total && Integer.MAX_VALUE % 10 < digit)
            return sign == 1 ? Integer.MAX_VALUE : Integer.MIN_VALUE;

        total = 10 * total + digit;
        index++;
    }
    return total * sign;
}
```

→ 3. 处理正负号

→ 不是数字 break

是数字的话, 把它减去字符'0', 就得到这个数字所代表的整型

返回本身; 超过的话, 返回integer max 值的

不断累加

→ 最后返回这个total 乘以它的符号

字符串操作问题

Anagram 异位词问题

Palindrome 回文串问题  
回文词 or