My Research Folder

Python进制转换的完全实现——2/10/16进制原/补码互转

一.需要转换的几种情况

不带符号位的整数	2进制	10进制	16进制
2进制	–	bin2dec()	bin2hex()
10进制	dec2bin()	–	dec2hex()
16进制	hex2bin()	hex2dec()	–

带符号位，即2进制用补码表示。

带符号位的整数	2进制	10进制	16进制
2进制	–	signed_bin2dec()	signed_bin2hex()
10进制	signed_dec2bin()	–	signed_dec2hex()
16进制	signed_hex2bin()	signed_hex2dec()	–

说明：这里忽略了八进制，因为很少用到。

部分函数的实现效果（myBin2dec2hex为自编模块，其中未使用任何第三方库）：

```
1  import myBin2dec2hex
2
3  print(myBin2dec2hex.bin2dec('1010'))
4  print(myBin2dec2hex.hex2bin('0x45'))
5  print(myBin2dec2hex.dec2bin(12))
6  print('--------------------')
7  print(myBin2dec2hex.signed_bin2dec('1010'))
8  print(myBin2dec2hex.signed_dec2bin(12))
9  print(myBin2dec2hex.signed_dec2bin(-12, 10))
10 print(myBin2dec2hex.signed_bin2hex('101001010', 4))
11 print(myBin2dec2hex.signed_hex2bin('45F1', 18))
```

✓ 0.2s

```
10
0b1000101
0b1100
--------------------
-6
0b01100
0b1111110100
0×ff4a
0b000100010111110001
```

徐大康的博客

二.Python 自带的进制转换函数

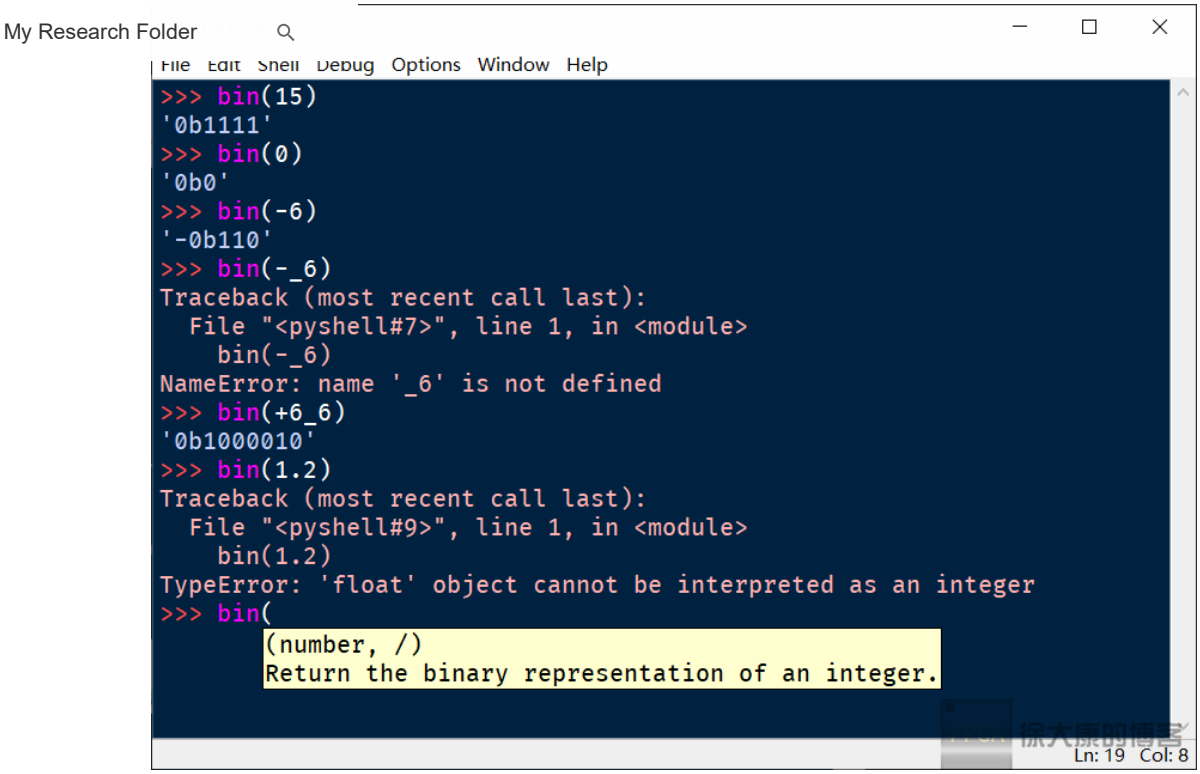
1.1 bin(整数)

输入整数，可带字符串，可在数字之间加任意下划线_。

在数字之间加下划线是Python3的新特性，注意下划线只能加在数字之间，头尾都会报错，且不能出现连续的两个下划线_，其它进制数也遵循这个规则，这也符合人们日常使用的习惯。

输出二进制字符串，且字符串前带有0b。

注意：1. bin()输入负数无法转换，只会加上一个负号；2. 输入小数会报错



1.2 int(2/10/16进制数/字符串，base = 2/10/16)

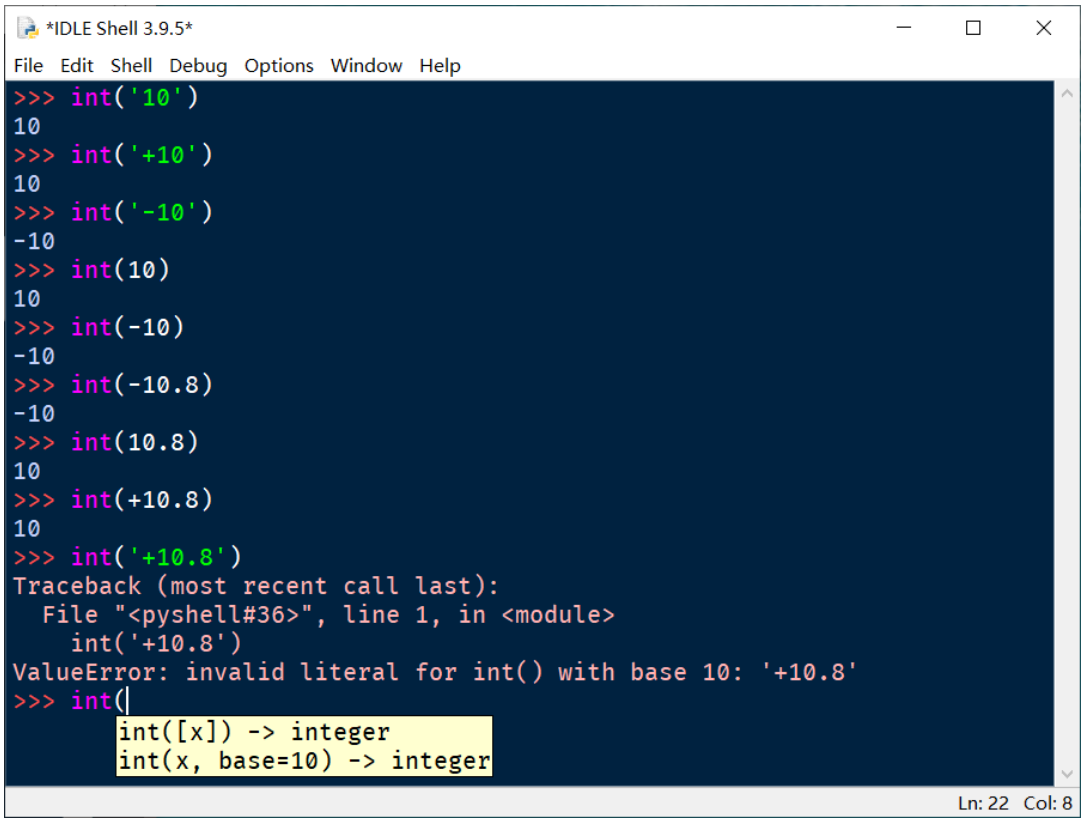
int有多种用法，如下所示。

1.2.1 int(10进制数/字符串)

默认的base = 10可省略，此用法与进制转换无关。

功能1：将10进制整数字符串（可带正负号, 可在数字之间加任意下划线_）转为整数，小数字符串是不行的。

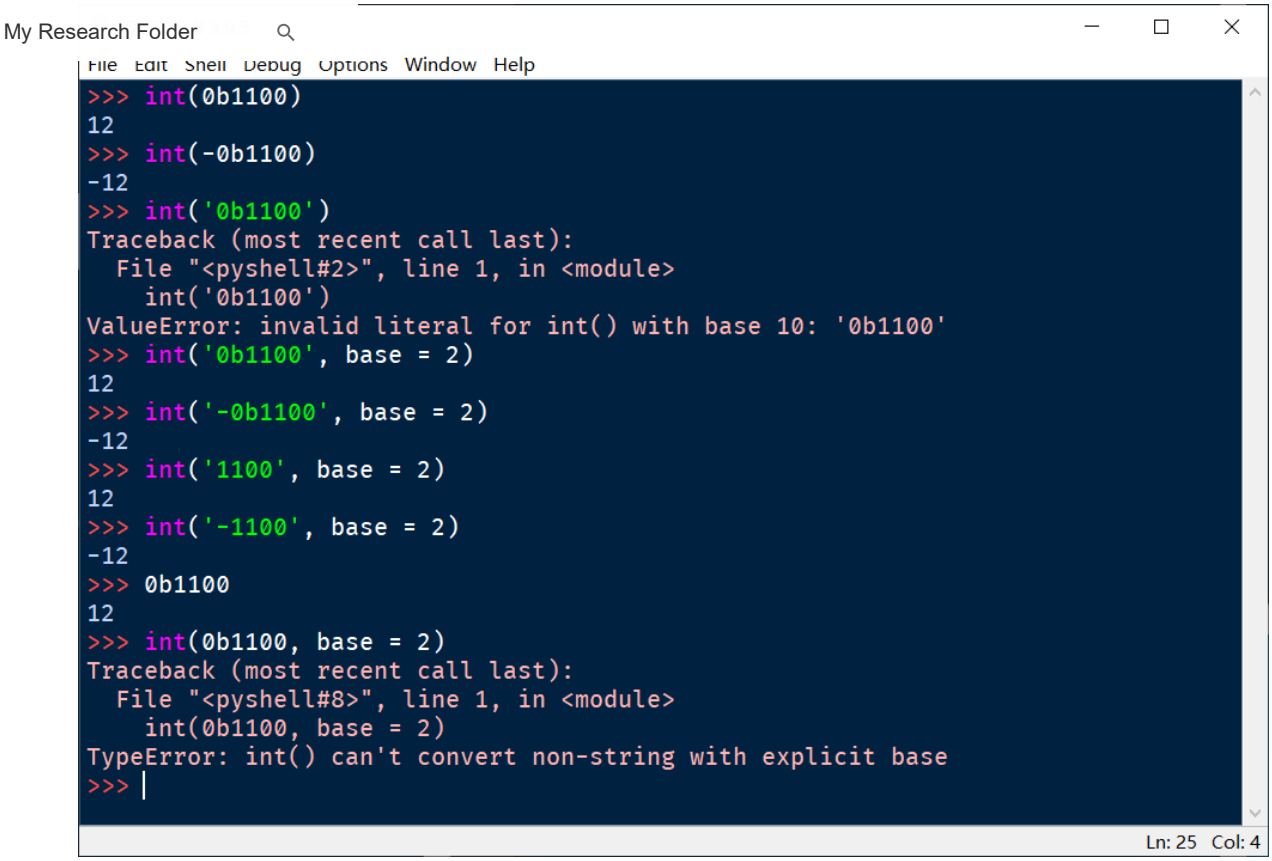
功能2：对10进制数（可带正负号）取整，小数部分会被舍弃掉。



1.2.2 int(2进制字符串, base = 2)

发现：Python将0b1100视为一个数，它和整数12是完全等价的，和12一样可以带正负号。

当加上base = 2时，第一个参数必须是字符串。即int(2进制字符串， base = 2)是固定用法。字符串带不带0b效果一样，同样可带正负号, 可在数字之间加任意下划线_。

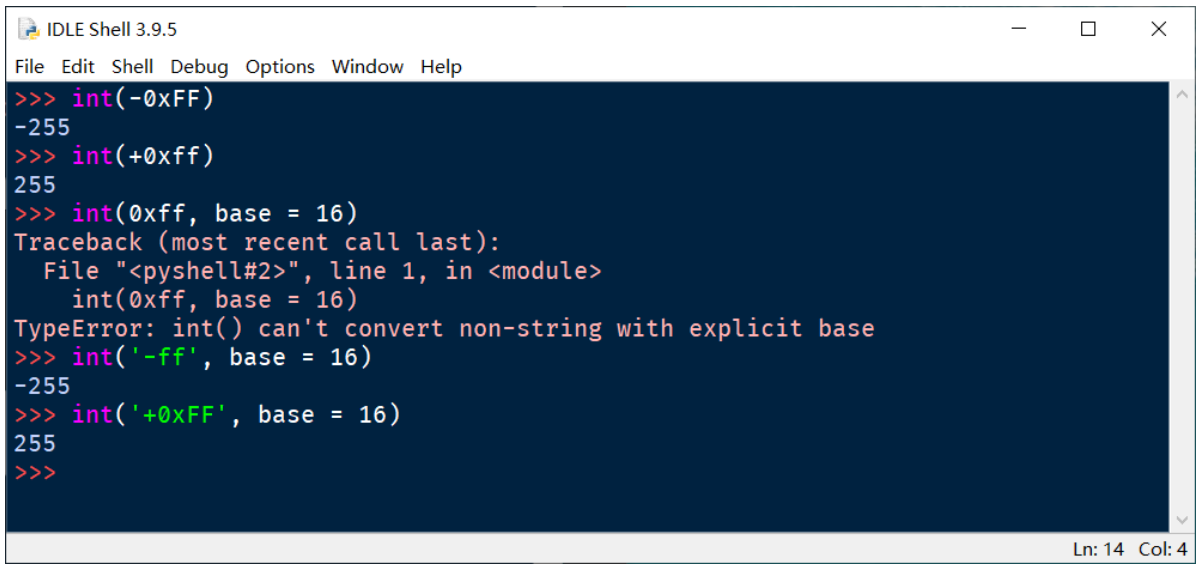


1.2.3 int(16进制字符串, base = 16)

类比1.2.2，16进制和2进制对于int()的用法是一样的。

Python将0xFF视为一个数，完全等价于255，可带正负号，且字母不区分大小写。

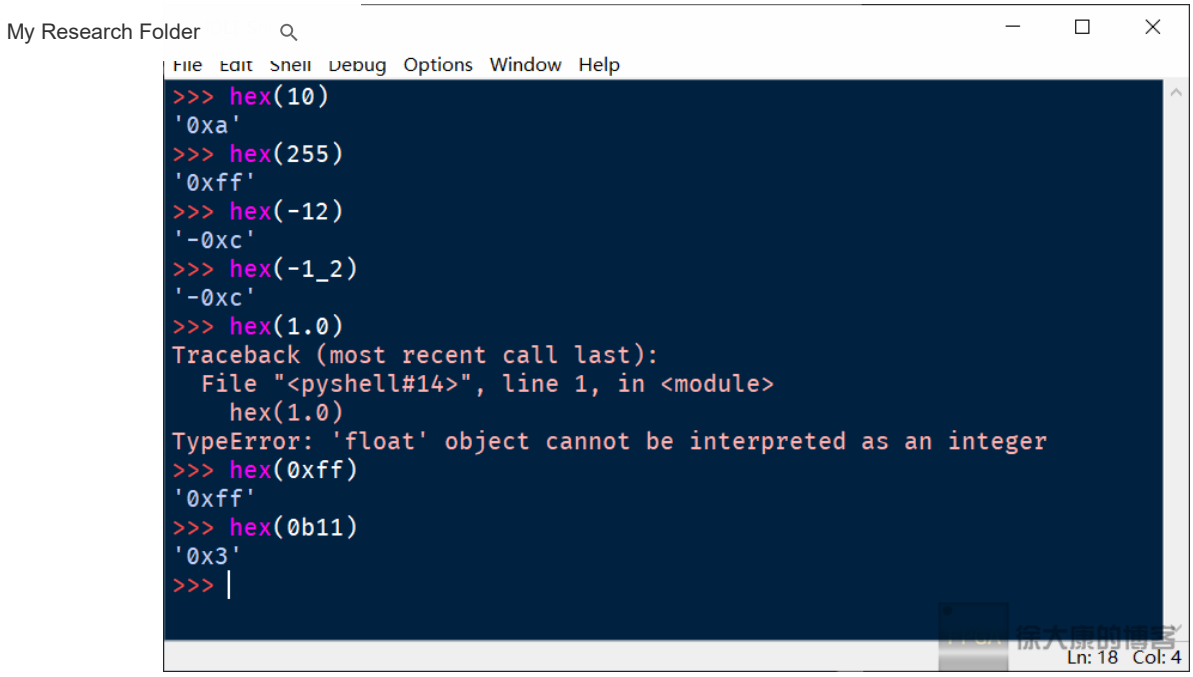
当加上base = 16后，第一个参数必须是字符串，所以int(16进制字符串, base = 16)也是固定用法。字符串带不带0x效果一样，不区分大小写，也可带正负号, 可在数字之间加任意下划线_。



1.3 hex(整数)

hex()和bin()的使用完全一样，输入整数，可带正负号，可在数字之间加任意下划线_。输出16进制字符串，带有0x的前缀。

注意：1. 不能转换负数，负号会保留。2. 不能输入小数，会报错。



三.不带符号位的自编函数实现进制转换

3.1 bin2dec ()—— 不带符号位的2进制字符串 -> 10进制整数

```
def bin2dec(bin_str: str) -> int:
    '''
    函数功能：不带符号位的2进制字符串 -> 10进制整数\n
    输入：2进制字符串，可带正负号，0b，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输出：10进制整数，只保留负号，正号不保留
    '''
    return int(bin_str.strip(), base = 2)
```

3.2 bin2hex() —— 不带符号位的2进制字符串 -> 不带符号位的16进制字符串

```
def bin2hex(bin_str: str, hex_width :int = -1) -> str:
    '''
    函数功能：不带符号位的2进制字符串 -> 不带符号位的16进制字符串\n
    输入参数1：2进制字符串，可带正负号，0b，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输入参数2：可选，16进制字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干0\n
    输出：16进制字符串，只保留负号，正号不保留
    '''
    new_bin_str = bin_str.strip()
    if (new_bin_str[0] == '+' or new_bin_str[0] == '-'): # 去除正负符号
        new_bin_str = new_bin_str[1:]
    if (new_bin_str[:2] == '0b'):
        new_bin_str = new_bin_str[2:]
    hex_str = hex(int(new_bin_str, base = 2))[2:]
    if (hex_width == -1):
        pass
    elif (hex_width < len(hex_str)): # 位宽小于实际16进制数位宽时
        print('位宽参数' + str(hex_width) + ' < 2进制' + bin_str + '输出16进制' + '0x' + hex_str
            + '实际位宽' + str(len(hex_str)) + '，请修正位宽参数')
    else:
        hex_str = '0' * (hex_width - len(hex_str)) + hex_str # 扩展位补0
    if (bin_str[0] == '-'):
        return '-' + '0x' + hex_str
    else:
        return '0x' + hex_str
```

3.3 dec2bin() —— 10进制整数/字符串 -> 不带符号位的2进制字符串

```
def dec2bin(dec_num: int, bin_width :int = -1) -> str:
    '''
    函数功能：10进制整数/字符串 -> 不带符号位的2进制字符串\n
    输入参数1：10进制整数/字符串，可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输入参数2：可选，2进制字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干0\n
    输出：16进制字符串，只保留负号，正号不保留
    '''
    input_dec_num = dec_num
    if (type(dec_num) == str):
        dec_num = int(dec_num.strip())

```

My Research Folder

```
        bin_str = old_bin_str[3:]
    else:
        bin_str = old_bin_str[2:]
    if (bin_width == -1):
        pass
    elif (bin_width < len(bin_str)):
        print('位宽参数' + str(bin_width) + ' < 10进制' + str(input_dec_num) + '输出2进制' + old_bin_str
            + '最小需要位宽' + str(len(bin_str)) + ', 请修正位宽参数')
    else:
        bin_str = '0' * (bin_width - len(bin_str)) + bin_str
    if (old_bin_str[0] == '-'):
        return '-0b' + bin_str
    else:
        return '0b' + bin_str
```

3.4 dec2hex() —— 10进制整数/字符串 -> 不带符号位的16进制字符串

```
def dec2hex(dec_num: int , hex_width: int = -1) -> str:
    '''
    函数功能：10进制整数/字符串 -> 不带符号位的16进制字符串\n
    输入参数1：10进制整数/字符串，可带正负号，前后可加任意个 \n 和 空格，数字间可加下划线\n
    输入参数2：可选，16进制字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干0\n
    输出：16进制字符串，只保留负号，正号不保留
    '''
    old_hex_str = bin2hex(dec2bin(dec_num))
    if (old_hex_str[0] == '-'):
        hex_str = old_hex_str[3:]
    else:
        hex_str = old_hex_str[2:]
    if (hex_width == -1):
        pass
    elif (hex_width < len(hex_str)):
        print('位宽参数' + str(hex_width) + ' < 10进制' + str(dec_num) + '输出16进制' + old_hex_str
            + '实际位宽' + str(len(hex_str)) + ', 请修正位宽参数')
    else:
        hex_str = '0' * (hex_width - len(hex_str)) + hex_str
    if (old_hex_str[0] == '-'):
        return '-0x' + hex_str
    else:
        return '0x' + hex_str
```

3.5 hex2dec() ——不带符号位的16进制字符串 -> 10进制整数

```
def hex2dec(hex_str: str) -> int:
    '''
    函数功能：不带符号位的16进制字符串 -> 10进制整数\n
    输入：16进制字符串，可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输出：10进制整数，只保留负号，正号不保留
    '''
    return int(hex_str.strip(), base = 16)
```

3.6 hex2bin() —— 不带符号位的16进制字符串 -> 不带符号位的2进制字符串

```
def hex2bin(hex_str: str, bin_width = -1) -> str:
    '''
    函数功能：不带符号位的16进制字符串 -> 不带符号位的2进制字符串\n
    输入：16进制字符串，可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输入参数2：可选，2进制字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干0\n
    输出：2进制字符串，只保留负号，正号不保留
    '''
    old_bin_str = dec2bin(hex2dec(hex_str))
    if (old_bin_str[0] == '-'):
        bin_str = old_bin_str[3:]
    else:
        bin_str = old_bin_str[2:]
    if (bin_width == -1):
        pass
    elif (bin_width < len(bin_str)):
        print('位宽参数' + str(bin_width) + ' < 16进制' + hex_str + '输出2进制' + old_bin_str
            + '实际位宽' + str(len(bin_str)) + ', 请修正位宽参数')
    else:
        bin_str = '0' * (bin_width - len(bin_str)) + bin_str
    if (old_bin_str[0] == '-'):
        return '-0b' + bin_str
    else:
        return '0b' + bin_str
```

My Research Folder 转换

4.1 signed_bin2dec ()—— 2进制补码字符串 -> 10进制整数

```
def signed_bin2dec(bin_str: str) -> int:
    '''
    函数功能：2进制补码字符串 -> 10进制整数\n
    输入：2进制补码字符串，不可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输出：10进制整数，只保留负号，正号不保留
    '''
    bin_str = bin_str.strip()
    if (bin_str[:2] == '0b'):
        if (bin_str[2] == '_'):
            bin_str = bin_str[3:]
        else:
            bin_str = bin_str[2:]
    if (bin_str[0] == '_'):
        int ('输入 ' + bin_str + ' 不合法，首字符不能是下划线 且 不允许出现连续两个下划线')
    elif (bin_str[0] == '0'):
        return int(bin_str, base = 2)
    elif (bin_str[0] == '1'):
        a = int(bin_str, base = 2) # 此语句可检查输入是否合法
        bin_str = bin_str.replace('_', '')
        return a - 2**len(bin_str)
    else:
        int('输入 ' + bin_str + ' 不合法，必须为2进制补码，不允许带正负号')
```

4.2 signed_bin2hex() —— 2进制补码字符串 -> 16进制补码字符串

```
def fourBin2OneHex(four_bin: str) -> str:
    '''
    函数功能：4位2进制字符串 -> 1位16进制字符串\n
    输入：4位2进制字符串，输入范围0000~1111\n
    输出：1位16进制字符串
    '''
    if (four_bin == '0000'):
        return '0'
    elif (four_bin == '0001'):
        return '1'
    elif (four_bin == '0010'):
        return '2'
    elif (four_bin == '0011'):
        return '3'
    elif (four_bin == '0100'):
        return '4'
    elif (four_bin == '0101'):
        return '5'
    elif (four_bin == '0110'):
        return '6'
    elif (four_bin == '0111'):
        return '7'
    elif (four_bin == '1000'):
        return '8'
    elif (four_bin == '1001'):
        return '9'
    elif (four_bin == '1010'):
        return 'a'
    elif (four_bin == '1011'):
        return 'b'
    elif (four_bin == '1100'):
        return 'c'
    elif (four_bin == '1101'):
        return 'd'
    elif (four_bin == '1110'):
        return 'e'
    elif (four_bin == '1111'):
        return 'f'
    else:
        int('输入2进制字符' + four_bin + '错误，2进制只能包含0或1')

def signed_bin2hex(bin_str: str, hex_width: int = -1) -> str:
    '''
    函数功能：2进制补码字符串 -> 16进制补码字符串\n
    输入参数1：2进制补码字符串，不可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输入参数2：可选，16进制补码字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干符号位\n
    输出：16进制补码字符串
    '''
    input_bin_str = bin_str

```

My Research Folder

```
        bin_str = bin_str[2:]
    elif (bin_str[0] == '0' or bin_str[0] == '1'):
        pass
    else:
        int('输入 ' + bin_str + ' 不合法，输入必须为2进制补码，不允许带正负号 且 首字符不能是下划线')
    # 检查输入是否合法，末尾字符不能是下划线 且 不能出现连续的两个下划线
    if (bin_str[-1] == '_' or '__' in bin_str):
        int('输入 ' + bin_str + ' 不合法，末尾字符不能是下划线 且 不能出现连续的两个下划线')
    else:
        bin_str = bin_str.replace('_', '') # 输入合法则去除下划线
    # 去掉2进制补码字符串前面多余的符号位，保留两位
    for i in range(len(bin_str)-1):
        if (bin_str[i+1] == bin_str[0]):
            if (i + 1 == len(bin_str)-1):
                bin_str = bin_str[i:]
            else:
                continue
        else:
            bin_str = bin_str[i:]
            break
    if (len(bin_str) % 4 > 0): # 补符号位到位宽为4的倍数
        bin_str = bin_str[0] * (4 - len(bin_str) % 4) + bin_str
    hex_str = ''
    for i in range(int(len(bin_str)/4)):
        hex_str += fourBin2OneHex(bin_str[i*4 : i*4+4])
    if (hex_width == -1):
        pass
    elif (hex_width < len(hex_str)):
        print('位宽参数' + str(hex_width) + ' < 2进制补码' + input_bin_str + '输出16进制补码'
            + '0x' + hex_str +'实际位宽' + str(len(hex_str)) + '，请修正位宽参数')
    else:
        if (hex_str[0] in ['0', '1', '2', '3', '4', '5', '6', '7']):
            hex_str = '0' * (hex_width - len(hex_str)) + hex_str
        else:
            hex_str = 'f' * (hex_width - len(hex_str)) + hex_str
    return '0x' + hex_str
```

4.3 signed_dec2bin() —— 10进制数/字符串 -> 2进制补码字符串

```
def signed_dec2bin(dec_num: int, bin_width: int = -1) -> str:
    '''
    函数功能：10进制数/字符串 -> 2进制补码字符串\n
    输入参数1：10进制数/字符串，可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输入参数2：可选，2进制补码字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干符号位\n
    输出：2进制补码字符串
    '''
    dec_num_str = str(dec_num)
    if (type(dec_num) == str):
        dec_num = int(dec_num.strip())
    if (dec_num == 0):
        bin_str = '0'
    elif (dec_num > 0):
        bin_str = '0' + bin(dec_num)[2:] # 补符号位0
    else:
        for i in range(10000):
            if (2**i + dec_num >= 0):
                bin_str = bin(2**(i+1) + dec_num)[2:] # 一个负数num的补码等于 (2**i + dec_num)
                break
    if (bin_width == -1):
        pass
    elif (bin_width < len(bin_str)):
        # 实际位宽大于设定位宽则报警告，然后按实际位宽输出
        print('位宽参数' + str(bin_width) + ' < 10进制' + dec_num_str + '输出2进制补码'
            + '0b' + bin_str + '实际位宽' + str(len(bin_str)) + '，请修正位宽参数')
    else:
        bin_str = bin_str[0] * (bin_width - len(bin_str)) + bin_str # 实际位宽小于设定位宽则补符号位
    return '0b' + bin_str
```

4.4 signed_dec2hex() —— 10进制数/字符串 -> 16进制补码字符串

```
def signed_dec2hex(dec_num: int, hex_width = -1) -> str:
    '''
    函数功能：10进制数/字符串 -> 16进制补码字符串\n
    输入参数1：10进制数/字符串，可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线_\n
    输入参数2：可选，16进制补码字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干符号位\n
    输出：16进制补码字符串
```

My Research Folder

```
gned_dec2bin(dec_num))[2:]
    if (hex_width == -1):
        pass
    elif (hex_width < len(hex_str)):
        print('位宽参数' + str(hex_width) + ' < 10进制' + str(dec_num) + '输出16进制补码' + '0x' +
            hex_str + '实际位宽' + str(len(hex_str)) + ', 请修正位宽参数')
    else:
        if (hex_str[0] in ['0', '1', '2', '3', '4', '5', '6', '7']):
            hex_str = '0' * (hex_width - len(hex_str)) + hex_str
        else:
            hex_str = 'f' * (hex_width - len(hex_str)) + hex_str
    return '0x' + hex_str
```

4.5 signed_hex2bin() —— 16进制补码字符串 -> 2进制补码字符串

```
def oneHex2fourBin(one_hex: str) -> str:
    '''
    函数功能：1位16进制字符串 -> 4位2进制字符串\n
    输入：1位16进制字符串，输入范围0~9，a~f或A~F\n
    输出：4位2进制字符串
    '''
    if (one_hex == '0'):
        return '0000'
    elif (one_hex == '1'):
        return '0001'
    elif (one_hex == '2'):
        return '0010'
    elif (one_hex == '3'):
        return '0011'
    elif (one_hex == '4'):
        return '0100'
    elif (one_hex == '5'):
        return '0101'
    elif (one_hex == '6'):
        return '0110'
    elif (one_hex == '7'):
        return '0111'
    elif (one_hex == '8'):
        return '1000'
    elif (one_hex == '9'):
        return '1001'
    elif (one_hex == 'a' or one_hex == 'A'):
        return '1010'
    elif (one_hex == 'b' or one_hex == 'B'):
        return '1011'
    elif (one_hex == 'c' or one_hex == 'C'):
        return '1100'
    elif (one_hex == 'd' or one_hex == 'D'):
        return '1101'
    elif (one_hex == 'e' or one_hex == 'E'):
        return '1110'
    elif (one_hex == 'f' or one_hex == 'F'):
        return '1111'
    else:
        int('输入16进制字符' + one_hex + '错误，16进制只能包含0~9，a~f或A~F')

def signed_hex2bin(hex_str: str, bin_width: int = -1) -> str:
    '''
    函数功能：16进制补码字符串 -> 2进制补码字符串\n
    输入参数1：16进制补码字符串，不可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输入参数2：可选，2进制补码字符串宽度，若实际输出宽度>此参数，警告并原样输出；若实际输出宽度<=此参数，高位补若干符号位\n
    输出：2进制补码字符串
    '''
    input_hex_str = hex_str
    hex_str = hex_str.strip()
    # 检查输入是否合法，不允许带正负号，首尾不能是下划线，不能出现连续两个下划线
    if (hex_str[0] in ['+', '-', '_'] or hex_str[-1] == '_' or '__' in hex_str):
        int('输入' + input_hex_str + '不合法，必须为16进制补码，不允许带正负号，'
            + '首尾不能是下划线，不能出现连续两个下划线')
    elif (hex_str[:2] == '0x'):
        hex_str = hex_str[2:]
    hex_str = hex_str.replace('_', '') # 输入合法则去除下划线
    bin_str = ''
    for i in hex_str:
        bin_str += oneHex2fourBin(i)
    # 去掉2进制补码字符串前面多余的符号位，保留两位
    for i in range(len(bin_str)-1):
        if (bin_str[i+1] == bin_str[0]):
```

My Research Folder

```
            else:
                continue
        else:
            bin_str = bin_str[i:]
            break
    if (bin_str == '00'):
        bin_str = '0'
    if (bin_width == -1):
        pass
    elif (bin_width < len(bin_str)):
        # 实际位宽大于设定位宽则报警告，然后按实际位宽输出
        print('位宽参数' + str(bin_width) + ' < 16进制补码' + input_hex_str + '输出2进制补码'
            + '0b' + bin_str + '实际位宽' + str(len(bin_str)) + '，请修正位宽参数')
    else:
        bin_str = bin_str[0] * (bin_width - len(bin_str)) + bin_str # 实际位宽小于设定位宽则补符号位
    return '0b' + bin_str
```

4.6 signed_hex2dec() —— 16进制补码字符串 -> 10进制整数

```
def signed_hex2dec(hex_str: str) -> int:
    '''
    函数功能：16进制补码字符串 -> 10进制整数\n
    输入：16进制补码字符串，不可带正负号，前后可加任意个 \\n 和 空格，数字间可加下划线\n
    输出：10进制整数，只保留负号，正号不保留
    '''
    return signed_bin2dec(signed_hex2bin(hex_str))
```

四. import自编模块

见我的另一篇博客，Python如何导入自己编写的py文件

五. 进制转换模块及其测试jupter notebook文档分享

我将上述所有转换函数放入bin2dec2hex.py文件中，还有测试用的文档，一起放入了我的码云开源仓库中，需要的可以去以下链接自取：

https://gitee.com/xuxiaokang/python-self-compiled-module.git