

位运算

1. 按位与 (AND)
2. 按位或 (OR)
3. 按位异或 (XOR)
4. 按位非 (NOT)
5. 左移 (Left Shift)
6. 右移 (Right Shift)

在Python中，位运算是一种直接对整数的二进制表示进行操作的方法。Python支持以下几种位运算符：

1. 按位与 (AND) : `&`
2. 按位或 (OR) : `|`
3. 按位异或 (XOR) : `^`
4. 按位非 (NOT) : `~`
5. 左移 (Left Shift) : `<<`
6. 右移 (Right Shift) : `>>`

以下是每种运算符的简单说明和示例：

1. 按位与 (AND)

按位与运算符 `&` 对两个数的二进制表示进行逐位与操作。只有两个相应的位都为1时，结果才为1。

```
1  a = 12 # 二进制 1100
2  b = 5  # 二进制 0101
3  result = a & b # 结果 0100
4  print(result) # 输出 4
```

2. 按位或 (OR)

按位或运算符 `|` 对两个数的二进制表示进行逐位或操作。两个相应的位中只要有一个为1，结果就为1。

```
1  a = 12 # 二进制 1100
2  b = 5  # 二进制 0101
3  result = a | b # 结果 1101
4  print(result) # 输出 13
```

3. 按位异或 (XOR)

按位异或运算符 `^` 对两个数的二进制表示进行逐位异或操作。两个相应的位相同则结果为0，不同则结果为1。

```
1  a = 12 # 二进制 1100
2  b = 5  # 二进制 0101
3  result = a ^ b # 结果 1001
4  print(result) # 输出 9
```

4. 按位非 (NOT)

按位非运算符 `~` 对一个数的二进制表示进行逐位取反操作。0变成1, 1变成0。

```
1  a = 5 # 二进制 0101
2  result = ~a # 结果 1010
3  print(result) # 输出 -6
```

5. 左移 (Left Shift)

左移运算符 `<<` 将一个数的二进制表示向左移动指定的位数, 右边空出的位补0。

```
1  a = 5 # 二进制 0101
2  result = a << 2 # 左移2位
3  print(result) # 输出 20
```

6. 右移 (Right Shift)

右移运算符 `>>` 将一个数的二进制表示向右移动指定的位数, 左边空出的位补符号位。

```
1  a = 20 # 二进制 10100
2  result = a >> 2 # 右移2位
3  print(result) # 输出 5
```

这些位运算符在处理二进制数据、优化性能、加密算法等方面非常有用。