

之前的文章中，我们提到了shell中常用的字符串处理操作，我们来回顾一下

- 一、获取变量的长度（字符串长度）。
- 二、对变量值进行截取（截取字符串）。
- 三、替换变量值中的某段字符。
- 四、删除变量值中的某段字符。
- 五、对变量值进行大小写转换。
- 六、当变量值为空时，对变量进行一些操作。

前两篇文章中，我们已经总结了上述操作中的前四种操作，今天，我们来介绍一下上述操作的最后两种操作，并将之前的所有总结进行一个汇总，以便以后进行回顾

字符串大小写转换

首先，我们准备一个变量，变量值为字母a到z，只不过a到k为小写，l到Z为大写，如下图所示

如果我们想要将变量testvar1中的所有小写英文字母都转换成大写，该怎么办呢？我们可以使用如下方法。

如上图所示，只要在输出变量值时加上"^"即可，"^"为数字键6对应的符号。

如果想要将变量testvar1中的所有大写英文字母都转换成小写，可以使用如下方法。

如上图所示，只要在输出变量值时加上",,"即可将字母中的大写字符转换成小写，很简单吧~

当变量值为空或者非空时对变量进行操作

在写shell脚本时，我们可能会遇到如下场景。

我们需要判断某个变量是否为空，如果变量值不为空，则使用变量当前的值，如果为空，则赋予变量一个默认值，以便之后进行处理。

那么，在脚本中可能会出现类似如下代码

```
if [ -z "$testvar" ];then
testvar="test"
fi
```

如上述代码所示，我们使用 if 结构，判断了变量testvar是否为空，如果为空，则将变量赋值为test，当然，如果变量值不为空，那么则不会被赋值，即保持变量原值

其实，我们可以使用更简单的方法，我们只需要一行代码，就可以代替上例中的三行代码，示例如下

```
${testvar:= "test"}
```

没错，就是这么简单，上述代码表示，如果变量testvar的值为空，则将testvar赋值为test，如果不为空，则testvar保持原值不变。我们在命令行中试试上述代码的将变量testvar的值设置为空值，然后使用上述语法，如下。

可以看到，当变量为空时，上述语法不仅会将变量的值设置为指定值，同时还会返回对应的值，我们可以直接使用echo命令进行输出，经过上述语法处理后，再次转已经被赋值。

当变量值不为空时，经过上述语法处理后，只会返回变量的原值，并不会改变变量的值，示例如下

可以发现，当变量为空时，使用上述语法会返回我们指定的值，同时还会将变量设定为我们指定的值，其实，还有另外一种方法，能够在变量为空时返回我们指定的变量进行赋值，示例如下。

```
${testvar:- "test"}
```

细心如你一定发现了，与之前的语法相比，只是语法中的"="替换成了"-"

上述语法表示，如果testvar变量值为空，则返回指定的文本"test"，如果testvar变量值不为空，则返回testvar变量本身的值，但是，当testvar变量值为空时，使用会返回指定的值，并不会对变量进行赋值操作，我们在命令行中看一下效果，示例如下。

可以看到，当变量为空时，上述语法并不会对变量赋值，当变量值不为空时，示例如下。

之前的两个示例，都是在变量为空时，返回我们指定的值，那么能不能在变量不为空时，返回我们指定的值呢？必须能的，使用如下语法即可。

```
${testvar:+ "test"}
```

上述语法表示，当testvar变量值不为空时，返回指定的字符串"test"，如果变量为空，则返回变量本身的值（即空值），示例如下

如上图所示，当变量值不为空时，使用上述语法会返回我们指定的值，但是并不会改变变量原来的值，如果变量值为空，使用上述语法，仍然返回空值，同时也不会值。

我们还可以在变量为空时，输出bash风格的error_info，示例如下

```
${testvar:? "error_info"}
```

上述语法表示testvar变量为空时，输出bash错误，错误信息为我们指定的字符串error_info

示例如下

总结

为了方便以后回顾，我们将之前所有与字符串处理、变量处理的相关操作进行汇总，并进行一个总结。

获取字符串长度

```
1 # testvar="12345"
2 # echo ${#testvar}
3 5
```

对变量值进行大小写转换

将变量值中的小写字母转换为大写

语法：echo \${var^^}

```
1 #示例
2 # testvar="case conversion"
3 # echo ${testvar}
4 case conversion
5 # echo ${testvar^^}
6 CASE CONVERSION
```

将变量值中的小写字母转换为大写

语法：echo \${var^^}

```
1 #示例
2 # testvar="CASE CONVERSION"
3 # echo ${testvar}
4 CASE CONVERSION
5 # echo ${testvar,,}
6 case conversion
```

当变量值为空或者非空时操作变量

```
${var:=value}
```

上述语法表示如果var为空，则返回value，并将value赋值给var，如果var不为空，则返回var本身的值，var不为空时，var值不会被改变，var为空时，var的值会被值。

```
${var:-value}
```

上述语法表示如果var为空，则返回value，如果var不为空，则返回var的值，无论var是否为空，var本身的值不会改变。

```
${var:+value}
```

上述语法表示如果var不为空，则返回value，如果var为空，则返回空值，无论var是否为空，var本身的值不会改变。

```
${var:?error_info}
```

上述语法表示如果var为空，那么在当前终端打印error_info，如果var的值不为空，则返回var的值，无论var是否为空，var本身的值都不会改变。

从指定位置截取字符串，截取到字符串的末尾

#下例表示从正数第4个字符以后开始截取，直到字符串的末尾。

```
1 # website="www.zsythink.net"
2 # echo ${website:4}
3 zsythink.net
```

#下例表示从倒数第4个字符开始截取，直到字符串的末尾

#注意：下例中的"负号"与"冒号"之间必须存在任意字符（通常使用0或空格占位），否则无法起到截取字符串的作用。

```
1 # website="www.zsythink.net"
2 # echo ${website:0-4}
3 .net
4 # echo ${website:-4}
5 .net
6 # echo ${website:a-4}
7 .net
```

从指定位置截取字符串，并且截取指定的长度

#下例表示从正数第4个字符以后开始截取，截取8个字符。

```
1 # website="www.zsythink.net"
2 # echo ${website:4:8}
3 zsythink
```

#下例表示从倒数第9个字符开始截取，截取5个字符。

```
1 # website="www.zsythink.net"
2 # echo ${website:0-9:5}
3 think
```

##centos7中，截取长度可以为负数，centos6中不行，centos7中的示例如下

#下例表示从正数第4个字符以后开始截取，截取到字符串的末尾，再将截取后的字符串的最后3个字符删除。

```
1 # website="www.zsythink.net"
2 # echo ${website:4:-3}
3 zsythink.
```

##centos7中，截取长度可以为负数，centos6中不行，centos7中的示例如下

#下例表示从倒数第4个字符开始截取，截取到字符串的末尾，再将截取后的字符串的最后1个字符删除。

```
1 # website="www.zsythink.net"
2 # echo ${website:0-4:-1}
3 .ne
```

掐头去尾截取之掐头截取

掐头截取：删除字符串中从左向右第一个遇到的char，以及其左侧的字符

##语法：`${var#*char}`

```
1  # 下例表示删除字符串中从左向右第一个遇到的 "."，以及其左侧的字符。
2  # website="www.zsythink.net"
3  # echo ${website#*.}
4  zsythink.net
5
6  # 下例表示删除字符串中从左向右第一个遇到的 "//"，以及其左侧的字符。
7  # website="https://www.zsythink.net"
8  # echo ${website#*//}
9  ww.zsythink.net
```

掐头截取：删除字符串中从左向右最后一个遇到的char，以及其左侧的字符

##语法：`${var##*char}`