

python~字符串操作

4.2 字符串的常用操作

- 在 `ipython3` 中定义一个 字符串，例如：`hello_str = ""`
- 输入 `hello_str.` 按下 `TAB` 键，`ipython` 会提示 字符串 能够使用的 方法 如下：

```
In [1]: hello_str.  
hello_str.capitalize    hello_str.isidentifier  hello_str.rindex  
hello_str.casefold      hello_str.islower       hello_str.rjust  
hello_str.center        hello_str.isnumeric     hello_str.rpartition  
hello_str.count         hello_str.isprintable   hello_str.rsplit  
hello_str.encode        hello_str.isspace       hello_str.rstrip  
hello_str.endswith     hello_str.istitle       hello_str.split  
hello_str.expandtabs    hello_str.isupper       hello_str.splitlines  
hello_str.find          hello_str.join           hello_str.startswith  
hello_str.format        hello_str.ljust         hello_str.strip  
hello_str.format_map    hello_str.lower         hello_str.swapcase  
hello_str.index         hello_str.lstrip        hello_str.title  
hello_str.isalnum       hello_str.maketrans     hello_str.translate  
hello_str.isalpha       hello_str.partition     hello_str.upper  
hello_str.isdecimal     hello_str.replace       hello_str.zfill  
hello_str.isdigit      hello_str.rfind
```

提示：正是因为 python 内置提供的方法足够多，才使得在开发时，能够针对字符串进行更加灵活的操作！应对更多的开发需求！

1) 判断类型 – 9

方法	说明
<code>string.isspace()</code>	如果 <code>string</code> 中只包含空格，则返回 <code>True</code>
<code>string.isalnum()</code>	如果 <code>string</code> 至少有一个字符并且所有字符都是字母或数字则返回 <code>True</code>
<code>string.isalpha()</code>	如果 <code>string</code> 至少有一个字符并且所有字符都是字母则返回 <code>True</code>
<code>string.isdecimal()</code>	如果 <code>string</code> 只包含数字则返回 <code>True</code> ，全角数字
<code>string.isdigit()</code>	如果 <code>string</code> 只包含数字则返回 <code>True</code> ，全角数字、(1)、\u00b2
<code>string.isnumeric()</code>	如果 <code>string</code> 只包含数字则返回 <code>True</code> ，全角数字，汉字数字

<code>string.istitle()</code>	如果 <code>string</code> 是标题化的 (每个单词的首字母大写) 则返回 <code>True</code>
<code>string.islower()</code>	如果 <code>string</code> 中包含至少一个区分大小写的字符, 并且所有这些 (区分大小写的) 字符都是小写, 则返回 <code>True</code>
<code>string.isupper()</code>	如果 <code>string</code> 中包含至少一个区分大小写的字符, 并且所有这些 (区分大小写的) 字符都是大写, 则返回 <code>True</code>

2) 查找和替换 – 7

方法	说明
<code>string.startswith(str)</code>	检查字符串是否是以 <code>str</code> 开头, 是则返回 <code>True</code>
<code>string.endswith(str)</code>	检查字符串是否是以 <code>str</code> 结束, 是则返回 <code>True</code>
<code>string.find(str, start=0, end=len(string))</code>	检测 <code>str</code> 是否包含在 <code>string</code> 中, 如果 <code>start</code> 和 <code>end</code> 指定范围, 则检查是否包含在指定范围内, 如果是返回开始的索引值, 否则返回 <code>-1</code>
<code>string.rfind(str, start=0, end=len(string))</code>	类似于 <code>find()</code> , 不过是从右边开始查找
<code>string.index(str, start=0, end=len(string))</code>	跟 <code>find()</code> 方法类似, 不过如果 <code>str</code> 不在 <code>string</code> 会报错
<code>string.rindex(str, start=0, end=len(string))</code>	类似于 <code>index()</code> , 不过是从右边开始
<code>string.replace(old_str, new_str, num=string.count(old))</code>	把 <code>string</code> 中的 <code>old_str</code> 替换成 <code>new_str</code> , 如果 <code>num</code> 指定, 则替换不超过 <code>num</code> 次

3) 大小写转换 – 5

方法	说明
<code>string.capitalize()</code>	把字符串的第一个字符大写
<code>string.title()</code>	把字符串的每个单词首字母大写
<code>string.lower()</code>	转换 <code>string</code> 中所有大写字符为小写
<code>string.upper()</code>	转换 <code>string</code> 中的小写字母为大写
<code>string.swapcase()</code>	翻转 <code>string</code> 中的大小写

4) 文本对齐 – 3

方法	说明
<code>string.ljust(width)</code>	返回一个原字符串左对齐，并使用空格填充至长度 <code>width</code> 的新字符串
<code>string.rjust(width)</code>	返回一个原字符串右对齐，并使用空格填充至长度 <code>width</code> 的新字符串
<code>string.center(width)</code>	返回一个原字符串居中，并使用空格填充至长度 <code>width</code> 的新字符串

5) 去除空白字符 – 3

方法	说明
<code>string.lstrip()</code>	截掉 <code>string</code> 左边（开始）的空白字符
<code>string.rstrip()</code>	截掉 <code>string</code> 右边（末尾）的空白字符
<code>string.strip()</code>	截掉 <code>string</code> 左右两边的空白字符

6) 拆分和连接 – 5

方法	说明
<code>string.partition(str)</code>	把字符串 <code>string</code> 分成一个 3 元素的元组 (<code>str</code> 前面, <code>str</code> , <code>str</code> 后面)
<code>string.rpartition(str)</code>	类似于 <code>partition()</code> 方法，不过是从右边开始查找
<code>string.split(str="", num)</code>	以 <code>str</code> 为分隔符拆分 <code>string</code> ，如果 <code>num</code> 有指定值，则仅分隔 <code>num + 1</code> 个子字符串， <code>str</code> 默认包含 <code>'r'</code> , <code>'t'</code> , <code>'n'</code> 和空格
<code>string.splitlines()</code>	按照行 (<code>'r'</code> , <code>'n'</code> , <code>'rn'</code>) 分隔，返回一个包含各行作为元素的列表
<code>string.join(seq)</code>	以 <code>string</code> 作为分隔符，将 <code>seq</code> 中所有的元素（的字符串表示）合并为一个新的字符串