重要知識點

isalnum()

判別字串是否為特定字元 >

isupper() / islower()



Python MySQL MySQL Get Started MySQL Create Database MySQL Create Table MySQL Insert **Python String Methods** MySQL Select MySQL Where MySQL Order By Next > Previous MySQL Delete MySQL Drop Table MySQL Update Python has a set of built-in methods that you can use on strings. MySQL Limit MySQL Join Python MongoDB Note: All string methods returns new values. They do not change the original string. MongoDB Get Started MongoDB Create Database MongoDB Create Collection Description MongoDB Insert Converts the first character to upper case MongoDB Find MongoDB Query Converts string into lower case MongoDB Sort center() Returns a centered string MongoDB Delete MongoDB Drop Collection Returns the number of times a specified value occurs in a string count() MongoDB Update MongoDB Limit Returns an encoded version of the string encode() 中文整理表格 RUNOOB.COM 搜索.... Ⅲ 分类导航 ◆ Python Number(数字) Python 基础教程 HTML / CSS Python 字符串 Python 简介 Python 环境搭建 字符串是 Python 中最常用的数据类型。我们可以使用引号('或')来创建字符串。 Python 中文编码 创建字符串很简单,只要为变量分配一个值即可。例如:

上方我們列出了常見的字串判別函數,然而 python 支援更多的字串操作,學員們可以參考下方兩個網

THE WORLD'S LARGEST WEB DEVELOPER SITE

CERTIFICATES 0 Q

• 英文整理表格

ш3schools.com

↑ HTML CSS JAVASCRIPT SQL PYTHON PHP MORE ▼

Python 基础语法 数据库 Python 变量类型 var1 = 'Hello World!' Python 运算符 Python 条件语句 Python 访问字符串中的值 Python 循环语句 ASP.NET Python 不支持单字符类型,单字符在 Python 中也是作为一个字符串使用。 Web Service Python 访问子字符串,可以使用方括号来截取字符串,如下实例: Python for 循环语句 Python 循环嵌套 Python break 语句 Python continue 语句 var1 = 'Hello World!' var2 = "Python Runoob" **String Formatting** 常見格式化符號 Description Example Escape Decimal integers (not floating point) "%d" % 45 == '45' "%i" % 45 == '45' %i Same as %d "%o" % 1000 == '1750' %0 Octal number

Unsigned decimal

'{1} {0} {2}'.format('Love','I','Python')

給定變數名稱

給定順序取得字串

給定變數名稱query字串

%u

"%x" % 1000 == '3e8' %X Hexadecimal lowercase "%X" % 1000 == '3E8' %X Hexadecimal uppercase "%e" % 1000 == '1.000000e+03' %e Exponential notation, lowercase "e" "%E" % 1000 == '1.000000E+03' %E Exponential notation, uppercase "E" %f Floating point real number "%f" % 10.34 == '10.340000' %F "%F" % 10.34 == '10.340000' Same as %f "%g" % 10.34 == '10.34' %g Either %f or %e, whichever is shorter "%G" % 10.34 == '10.34' %G Same as %g but uppercase "%c" % 34 == '"' Character format %C "%r" % int == "<type 'int'>" %r Repr format (debugging format) "%s there" % 'hi' == 'hi there' %5 String format 表格資料來源:<u>stackoverflow</u> String.format() 以可讀性更高的語法做到 string formatting 基本语法是通过 {} 和:来代替以前的% '{} {} {}'.format('I','Love','Python') * output: 'I Love Python' 不指定位置,按照順序排列

* output: 'I Love Python'

* output: 'I Love Python'

'{name} {verb} {language}'.format(verb = 'Love', name = 'I', language = 'Python')

"%u" % -1000 == '-1000'

不指定位置,按照順序排列:基本语法是通过 {} 和:来代替以前的 %

In [23]: '{} {} '.format('I','Love','Python')

Out[23]: 'I Love Python'

給定順序

In [24]: '{1} {0} {2}'.format('Love','I','Python')

Out[24]: 'I Love Python'

In [25]: '{name} {verb} {language}'.format(verb = 'Love', name = 'I', language = 'Python')
Out[25]: 'I Love Python'

1 '{} {} {}'.format('I', 'Love', 'Python')
2 '{1} {0} {2}'.format('Love', 'I', 'Python')
3 '{name} {verb} {language}'.format(verb = 'Love', name = 'I', language = 'Python')
4

支援多種方式query輸入字元

list_ = ['Love', 'I', 'Python']
'{0[1]} {0[0]} {0[2]}'.format(list_)

其中0代表輸入list ** output: 'I Love Python'

dic_ = {'verb': 'Love', 'name': 'I', 'language': 'Python'}
'{name} {verb} {language}'.format(**dic_)

輸入字典取得value值 ** output: 'I Love Python'

* output: '4.356'

'{:.2f}'.format(4.356) == '%.2f' % (4.356)

* output: '4.36'

'{:.2%}'.format(4.356)

* output: '435.60%'

補齊數字字串長度

補齊數字字串長度

In [46]: ## < 補右邊, 補0到10位數
'{:0<10d}'.format(5)

Out[46]: '50000000000'

In [32]: ## > 補左邊, 補1到10位數

Out[32]: '1111111115'

參考資料

'{:1>10d}'.format(5)

更彈性的輸入,不一定要指定escape

'{}'.format(4.356) == '%.3f' % (4.356)

In [33]: ## > 補左邊, 補空格,可用來對齊
'{:>10d}'.format(5)

Out[33]: ' 5'

1 ## > 補右邊, 補0到10位數
2 '[:0>10d}'.format(5)
3 ## > 補左邊, 補1到10位數
4 '[:1>10d}'.format(5)
5 ## > 補左邊, 補空格,可用來對齊
6 '[:>10d}'.format(5)

String Formatting