



# 字符串

车万翔

哈尔滨工业大学



# 字符串的应用



```
9 import pygame, sys
10 from pygame.locals import *
11
12 pygame.init()
13
14 FPS = 30 # frames per second setting
15 fpsClock = pygame.time.Clock()
16
17 # set up the window
18 DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
19 pygame.display.set_caption('Animation')
20
21 WHITE = (255, 255, 255)
22 catImg = pygame.image.load('cat.png')
23 catx = 10
24 caty = 10
25 direction = 'right'
26
27 while True: # the main game loop
28     DISPLAYSURF.fill(WHITE)
29
30     if direction == 'right':
31         catx += 5
32         if catx == 280:
33             direction = 'down'
34     elif direction == 'down':
35         caty += 5
36         if caty == 220:
37             direction = 'left'
38     elif direction == 'left':
39         catx -= 5
40         if catx == 10:
41             direction = 'up'
42     elif direction == 'up':
43         caty -= 5
44         if caty == 10:
45             direction = 'right'
46
```

cat

Replace with: dog

☒ Replace/find ☐ Replace all

字符串查找和替换



# 字符串的应用



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

Google

哈尔滨工业大学



[网页](#) [图片](#) [地图](#) [新闻](#) [更多](#) [搜索工具](#)

找到约 8,800,000 条结果 (用时 0.32 秒)

**哈尔滨工业大学**

[www.hit.edu.cn/](http://www.hit.edu.cn/)

包括学校总体介绍, 招生就业, 院系及教学科研简介, 学生管理等。

评分: **26 / 30 · 26 条 Google 评论** [撰写评论](#)

黑龙江省哈尔滨市南岗区西大直街92号 邮政编码: 150001  
0451 8641 2114

[哈工大电子邮件系统](#) - [院系部门](#) - [学校简介](#)

**哈尔滨工业大学\_百度百科**

[baike.baidu.com/view/13897.htm](http://baike.baidu.com/view/13897.htm)

**哈尔滨工业大学**, 隶属于国家工业和信息化部, 是由工信部、教育部、黑龙江省共建的国家重点大学, 其前身是创建于1920年的哈尔滨中俄工业学校, 是中国近代培养 ...

[学校简介](#) - [师资力量](#) - [院系设置](#) - [教学情况](#)

**哈尔滨工业大学\_分数线\_专业设置\_新浪院校库\_新浪教育\_新浪网**

[kaoshi.edu.sina.com.cn](http://kaoshi.edu.sina.com.cn) > [所有院校](#)

**哈尔滨工业大学**隶属于工业和信息化部, 是由工信部、教育部、黑龙江省共建的国家重点大学, 是首批进入国家“211工程”和“985工程”建设的若干所大学之一。

**2013哈尔滨工业大学录取分数线\_哈尔滨工业大学各省\_高考院校库**

[college.gaokao.com](http://college.gaokao.com) > [全部高校](#)

50+ 项 - 高考院校库为广大高中生提供**哈尔滨工业大学**录取分数线、哈尔滨工业 ...

专业名称	专业小类	平均分	最高分
城市规划	土建类	573	630
软件工程	电气信息类	574	609

**哈尔滨工业大学的新闻搜索结果**



中国教育在线

**哈工大研究生院副院长马广富分享研究生培养问题**

**中国教育在线** - 22 小时前

**哈工大研究生院副院长马广富分享研究生培养问题**, 6月27日, 由中国教育在线举办的。



## 哈尔滨工业大学

[路线](#)

哈尔滨工业大学, 简称哈工大, 创建于1920年, 是一所以理工为主, 理、工、文、管相结合, 多学科、开放式、研究型国家重点大学, 为C9联盟成员之一。1996年首批进入中国“211工程”重点建设的院校之一。 [维基百科](#)

**地址:** 黑龙江省哈尔滨市南岗区西大直街92号 邮政编码: 150001

**电话号码:** 0451 8641 2114

**省份:** 黑龙江省

**成立时间:** 1920 年

用户还搜索了



**哈尔滨工程大学**  
哈尔滨市



**清华大学**  
北京市



**哈尔滨理工大学**  
哈尔滨市



**北京大学**  
北京市



**大连理工大学**  
大连市

[反馈/详情](#)

# 搜索引擎



# 字符串定义



- ❖ 字符串 ( String ) 是一个字符的序列
- ❖ 使用成对的单引号或双引号括起来

```
two_quote_str = "hello world"  
one_quote_str = 'hello world'
```

- ❖ 或者三引号 ( """"或''' )
  - 保留字符串中的全部格式信息

```
''' this is  
a test  
today '''
```



# 基本的字符串运算



## ❖ 长度 ( `len()` 函数 )

```
>>> first_name = 'Michael'
>>> len(first_name)
7
```

## ❖ 拼接 ( `+` )

```
>>> name = first_name + ' Jordan '
>>> print name
Michael Jordan
```

## ❖ 重复 ( `*` )

```
>>> name * 3
'Michael Jordan Michael Jordan Michael Jordan '
```



# 成员运算符 ( **in** )



## ❖ 判断一个字符串是否是另一个字符串的子串

- 返回值：**True** 或者 **False**

```
>>> name = 'Michael Jordan'
>>> 'a' in name
True
>>> 'Jordan' in name
True
>>> 'jordan' in name
False
>>> 'x' in name
False
```



## ❖ 枚举字符串的每个字符

```
>>> my_str = 'hello world'
>>> for char in my_str:
...     print char
...
h
e
l
l
o

w
o
r
l
d
```



# 示例



- ❖ 编写 `vowels_count` 函数，计算一个字符串中元音字母（`aeiou`或`AEIOU`）的数目

```
def vowels_count(string):  
    ret = 0  
    for c in string:  
        if c in 'aeiouAEIOU':  
            ret += 1  
  
    return ret
```





# 字符串索引 ( index )



- ❖ 字符串中每个字符都有一个索引值 ( 下标 )
- ❖ 索引从0 ( 前向 ) 或-1 ( 后向 ) 开始

forward index	0	1	2	3	4	5	6	7	8	9	10
	h	e	l	l	o		w	o	r	l	d
backward index	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

## ❖ 索引运算符 [ ]

```
>>> my_str = 'hello world'
>>> my_str[2]
'l'
>>> my_str[-1]
'd'
>>> my_str[11]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range
```



# 切片 ( Slicing )



## ❖ 选择字符串的子序列

## ❖ 语法 [start : finish]

- start : 子序列开始位置的索引值
- finish : 子序列结束位置的下一个字符的索引值

h	e	l	l	o		w	o	r	l	d
0	1	2	3	4	5	6	7	8	9	10

## ❖ 如果不提供 start 或者 finish , 默认 start 为第一个字符开始 , finish 为最后一个字符

```
>>> my_str = 'hello world'
>>> my_str[6:10]
'world'
>>> my_str[6:]
'world'
>>> my_str[:6]
'hello '
```



# 计数参数 (Count by Argument)



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

## ❖ 接收三个参数

- [start : finish : countBy]
- 默认 countBy 为 1

```
>>> my_str = 'hello world'
>>> my_str[0:11:2]
'hlowrd'
```

## ❖ 获得逆字符串

```
>>> my_str = 'spam'
>>> reverse_str = my_str[::-1]
>>> print reverse_str
maps
```



# 字符串是不可变的 (Immutable)



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

## ❖ 一旦生成，则内容不能改变

```
>>> my_str = 'hello world'
>>> my_str[1] = 'a'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item
                        assignment
```

## ❖ 通过切片等操作，生成一个新的字符串

```
>>> my_str = my_str[:1] + 'a' + my_str[2:]
>>> print my_str
hallo world
```



# 字符串方法 ( Methods )



## ❖ 方法

- 对象提供的函数
- 如：my\_str.replace(old, new) 生成一个新的字符串，其中使用 new 替换 old 子串

```
>>> my_str = 'hello world'
>>> my_str = my_str.replace('e', 'a')
>>> print my_str
hallo world
```

## ❖ 注意

- replace 方法返回一个新的字符串，原字符串内容不变
- 新字符串重新赋值给原来的变量



# 更多字符串方法



## ❖ find

```
>>> my_str = 'hello world'
>>> my_str.find('l') # find index of 'l' in my_str
2
```

## ❖ split

```
>>> my_str = 'hello world'
>>> my_str.split()
['hello', 'world']
```

## ❖ 其它方法

- `dir(str)`
- <http://docs.python.org/lib/string-methods.html>



# 示例：人名游戏



## ❖ 读取人名列表文件 names.txt，将每个人名转换为首字母大写，其它字母小写的格式

names.txt

AA  
AAB  
AABERG  
AABY  
AADLAND  
AAFEDT  
AAGAARD  
AAGARD  
AAGESEN  
AAKER  
AAKHUS  
AAKRE  
AALAND  
AALBERS  
AALDERINK  
AALFS  
AALGAARD  
AALTO  
AAMODT  
AAMOLD  
AAMOT  
AAMOTH  
AANDERUD  
AANENSON  
AANERUD  
AANESTAD  
AANONSEN  
AARDAL  
AARDEMA  
AARDSMA

- 文件操作
- 打开文件
  - `f = open(filename, mode)`
    - mode 有 r (读, 默认), w (写) 等
- 按行读取文件内容
  - `for line in f:`
    - `pass`
- 关闭文件
  - `f.close()`
- 写文件
  - `f.write(str)`
- 更多说明
  - <https://docs.python.org/2/tutorial/inputoutput.html#reading-and-writing-files>



# 示例：人名游戏



- ❖ 读取人名列表文件 names.txt，将每个人名转换为首字母大写，其它字母小写的格式

names.txt

AA  
AAB  
AABERG  
AABY  
AADLAND  
AAFEDT  
AAGAARD  
AAGARD  
AAGESEN  
AAKER  
AAKHUS  
AAKRE  
AALAND  
AALBERS  
AALDERINK  
AALFS  
AALGAARD  
AALTO  
AAMODT  
AAMOLD  
AAMOT  
AAMOTH  
AANDERUD  
AANENSON  
AANERUD  
AANESTAD  
AANONSEN  
AARDAL  
AARDEMA  
AARDSMA

```
f = open('names.txt')  
  
for line in f:  
    name = line.strip()  
    print name.title()  
  
f.close()
```





# 示例：人名游戏



- ❖ 编写一个名为 `is_palindrome` 的函数，判断一个人名是否为回文，如 "BOB" 是回文

```
def is_palindrome(name):  
    low = 0  
    up = len(name) - 1  
    while low < up:  
        if name[low] != name[up]:  
            return False  
        low += 1  
        up -= 1  
    return True
```

- ❖ 递归实现

```
def is_palindrome(name):  
    if len(name) < 2:  
        return True  
    if name[0] == name[-1]:  
        return is_palindrome(name[1:-1])  
    else:  
        return False
```



# 字符串比较



## ❖ 任何一个字符都对应一个数字

- ASCII (American Standard Code for Information Interchange)

## ❖ 直接比较对应数字的大小

ASCII CONTROL CODE CHART

BITS					0		0		1		0		1		1		1												
					0		1		0		1		0		1		1												
					CONTROL				SYMBOLS NUMBERS				UPPER CASE				LOWER CASE												
b4	b3	b2	b1																										
0	0	0	0	0	NUL		16	DLE		32	SP		48	0		64	P		80	‘		96	p		112				
0	0	0	0	0	0	10	0	20	20	40	30	60	60	40	100	50	120	60	140	70	160	60	140	70	160	60	140	70	160
0	0	0	1	1	SOH		17	DC1		33	!		49	1		65	A		81	Q		97	a		113	q			
0	0	1	0	1	1	11	21	21	41	31	61	41	61	101	51	121	61	141	71	161	61	141	71	161	61	141	71	161	
0	0	1	0	2	STX		18	DC2		34	"		50	2		66	B		82	R		98	b		114	r			
0	0	1	0	2	2	12	22	22	42	32	62	42	62	102	52	122	62	142	72	162	62	142	72	162	62	142	72	162	
0	0	1	1	3	ETX		19	DC3		35	#		51	3		67	C		83	S		99	c		115	s			
0	0	1	1	3	3	13	23	23	43	33	63	43	63	103	53	123	63	143	73	163	63	143	73	163	63	143	73	163	
0	1	0	0	4	EOT		20	DC4		36	\$		52	4		68	D		84	T		100	d		116	t			
0	1	0	0	4	4	14	24	24	44	34	64	44	64	104	54	124	64	144	74	164	64	144	74	164	64	144	74	164	
0	1	0	1	5	ENQ		21	NAK		37	%		53	5		69	E		85	U		101	e		117	u			
0	1	0	1	5	5	15	25	25	45	35	65	45	65	105	55	125	65	145	75	165	65	145	75	165	65	145	75	165	
0	1	1	0	6	ACK		22	SYN		38	&		54	6		70	F		86	V		102	f		118	v			
0	1	1	0	6	6	16	26	26	46	36	66	46	66	106	56	126	66	146	76	166	66	146	76	166	66	146	76	166	



# 字符串比较



## ❖ 字典序 ( Dictionary order )

- 首先比较两个字符串的第一个字符
- 如果相同，则比较下一个字符
- 如果不同，则字符串的大小关系由这两个字符的关系决定
- 如果其中一个字符为空（较短），则其更小



## ❖ 举例

```
>>> 'a' < 'b'
True
>>> 'aaab' < 'aaac'
True
>>> 'aa' < 'aab'
True
>>> 'abc' > 'bcd'
False
```



# 示例：人名游戏



- ❖ 编写函数 `is_ascending`，判断一个人名的字母是否为升序排列（允许重复字母）

```
def is_ascending(name):  
    p = name[0]  
    for c in name:  
        if p > c:  
            return False  
        else:  
            p = c  
    return True
```



# 字符串格式化 ( Formatting )



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 输出更规格的结果
- ❖ format 方法，如：

```
>>> print "Hello {} good {}".format(5, 'DAY')
Hello 5 good DAY.
```

- ❖ 括号的格式
  - {field name:align width.precision type}

```
>>> print math.pi
3.14159265359
>>> print 'PI is {:.4f}'.format(math.pi)
PI is 3.1416
>>> print 'PI is {:9.4f}'.format(math.pi)
PI is      3.1416
>>> print 'PI is {:e}'.format(math.pi)
PI is 3.141593e+00
```

- ❖ 参考

- <http://docs.python.org/2/library/string.html#format-string-syntax>



# 正则表达式 ( Regular Expressions )



哈尔滨工业大学  
HARBIN INSTITUTE OF TECHNOLOGY

## ❖ 判断一个人名 ( name ) 是否满足下列模式

- Michael : name == 'Michael'
- 以 Mi 开始 : name[:2] == 'Mi'
- 包含 cha 子串 : 'cha' in name
- 包含 c?a 子串 : ?
- 包含 c\*e 子串 : ?

## ❖ 正则表达式用来描述字符串的模式

- . 表示任意字符
- \d+ 表示一系列数字
- [a-z]表示一个小写字母
- .....

## ❖ 参考

- <https://docs.python.org/2/library/re.html>



# 示例：人名游戏



## ❖ 判断一个人名是否含有 C.A 模式

```
import re

f = open('names.txt')

pattern = '(C.A)'

for line in f:
    name = line.strip()
    result = re.search(pattern, name)
    if result:
        print 'Find in {}'.format(name)

f.close()
```