



链滴

# python 时间格式化

作者: [hiquanta](#)

原文链接: <https://ld246.com/article/1509005238480>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
import time
print time.time()
```

输出的结果是：

```
1279578704.6725271
```

但是这样是一连串的数字不是我们想要的结果，我们可以利用time模块的格式化时间的方法来处理：

```
time.localtime(time.time())
```

用time.localtime()方法，作用是格式化时间戳为本地的时间。

输出的结果是：

```
time.struct_time(tm_year=2010, tm_mon=7, tm_mday=19, tm_hour=22, tm_min=33, tm_sec=9, tm_wday=0, tm_yday=200, tm_isdst=0)
```

现在看起来更有希望格式成我们想要的的时间了。

```
time.strftime('%Y-%m-%d',time.localtime(time.time()))
```

最后用time.strftime()方法，把刚才的一大串信息格式化成我们想要的东西，现在的结果是：

```
2010-07-19
```

time.strftime里面有很多参数，可以让你能够更随意的输出自己想要的东西：

下面是time.strftime的参数：

```
strftime(format[, tuple]) -> string
```

将指定的struct\_time(默认为当前时间)，根据指定的格式化字符串输出

python中时间日期格式化符号：

%y 两位数的年份表示 (00-99)

%Y 四位数的年份表示 (000-9999)

%m 月份 (01-12)

%d 月内中的一天 (0-31)

%H 24小时制小时数 (0-23)

%I 12小时制小时数 (01-12)

%M 分钟数 (00=59)

%S 秒 (00-59)

%a 本地简化星期名称

%A 本地完整星期名称

%b 本地简化的月份名称

%B 本地完整的月份名称

%c 本地相应的日期表示和时间表示

%j 年内的一天 (001-366)

%p 本地A.M.或P.M.的等价符

%U 一年中的星期数 (00-53) 星期天为星期的开始

%w 星期 (0-6) , 星期天为星期的开始  
%W 一年中的星期数 (00-53) 星期一为星期的开始  
%x 本地相应的日期表示  
%X 本地相应的时间表示  
%Z 当前时区的名称  
%% %号本身