

time

--time.clock()

返回从第一次调用time.clock()到现在的时间差(float)，较为精确

--time.gmtime()

返回.0时区的时间元组

FE:

```
>>> import time
```

```
>>> time.gmtime()
```

```
time.struct_time(tm_year=2017, tm_mon=2, tm_mday=2, tm_hour=2, tm_min=53, tm_sec=41, tm_wday=3, tm_yday=33, tm_isdst=0)
```

--time.localtime()

返回当地的时间元组

FE:

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

```
time.struct_time(tm_year=2017, tm_mon=2, tm_mday=2, tm_hour=10, tm_min=55, tm_sec=11, tm_wday=3, tm_yday=33, tm_isdst=0)
```

--创建struct_time对象

FE:

```
>>> t = time.localtime()
```

```
>>> t.tm_year
```

```
2017
```

```
>>> t.tm_mon
```

```
2
```

```
>>> t.tm_mday
```

```
2
```

```
>>> t.tm_yday
```

```
33
```

```
>>> t.tm_wday
```

```
3
```

```
>>> type(t.tm_year)
```

```
<class 'int'>
```

--time.sleep(secs)

设置线程停歇

=====2017/2/2=====

--time.strftime(format[, t])

format是要格式化的字符串，t是一个struct_time对象

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

FE:

```
>>> time.strftime("%b %d %Y %H:%M:%S", time.gmtime())
'Feb 02 2017 13:00:57'
>>> type(time.strftime("%b %d %Y %H:%M:%S", time.gmtime()))
<class 'str'>
```