



python

```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.color("purple")
turtle.setheading(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```

Python语言程序设计

模块2: time库的使用



嵩天
北京理工大学



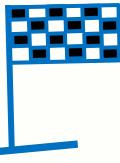
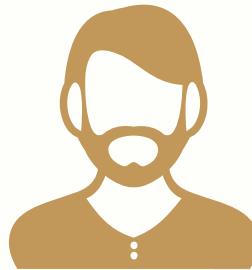


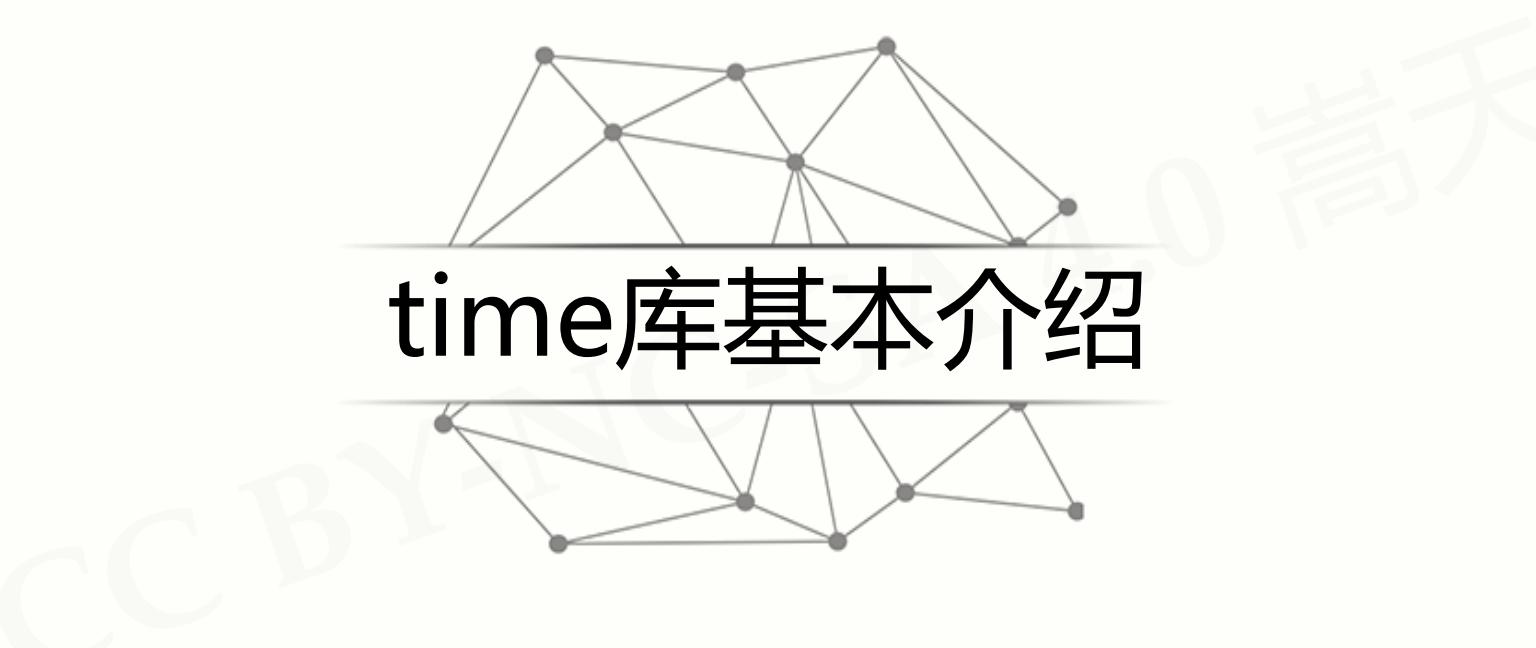
单元开篇



模块2: time库的使用

- time库基本介绍
- 时间获取
- 时间格式化
- 程序计时应用





time库基本介绍

time库概述

time库是Python中处理时间的标准库

- 计算机时间的表达

```
import time
```

- 提供获取系统时间并格式化输出功能

```
time.<b>()</b>
```

- 提供系统级精确计时功能，用于程序性能分析

time库概述

time库包括三类函数

- **时间获取** : `time()` `ctime()` `gmtime()`
- **时间格式化** : `strftime()` `strptime()`
- **程序计时** : `sleep()`, `perf_counter()`



时间获取



时间获取

函数	描述
time()	获取当前时间戳，即计算机内部时间值，浮点数 <code>>>>time.time()</code> <code>1516939876.6022282</code>
ctime()	获取当前时间并以易读方式表示，返回字符串 <code>>>>time.ctime()</code> <code>'Fri Jan 26 12:11:16 2018'</code>

时间获取

函数	描述
gmtime()	获取当前时间，表示为计算机可处理的时间格式 <code>>>>time.gmtime()</code> <code>time.struct_time(tm_year=2018, tm_mon=1, tm_mday=26, tm_hour=4, tm_min=11, tm_sec=16, tm_wday=4, tm_yday=26, tm_isdst=0)</code>



时间格式化



时间格式化

将时间以合理的方式展示出来

- 格式化：类似字符串格式化，需要有展示模板
- 展示模板由特定的格式化控制符组成
- `strftime()`方法

时间格式化

函数	描述
strftime(tp, ts)	tp是格式化模板字符串，用来定义输出效果 ts是计算机内部时间类型变量 <code>>>>t = time.gmtime()</code> <code>>>>time.strftime("%Y-%m-%d %H:%M:%S", t)</code> <code>'2018-01-26 12:55:20'</code>

格式化控制符

格式化字符串	日期/时间说明	值范围和实例
%Y	年份	0000~9999 , 例如 : 1900
%m	月份	01~12 , 例如 : 10
%B	月份名称	January~December , 例如 : April
%b	月份名称缩写	Jan~Dec , 例如 : Apr
%d	日期	01~31 , 例如 : 25
%A	星期	Monday~Sunday , 例如 : Wednesday

格式化控制符

格式化字符串	日期/时间说明	值范围和实例
%a	星期缩写	Mon~Sun , 例如 : Wed
%H	小时 (24h制)	00~23 , 例如 : 12
%h	小时 (12h制)	01~12 , 例如 : 7
%p	上/下午	AM, PM , 例如 : PM
%M	分钟	00~59 , 例如 : 26
%S	秒	00~59 , 例如 : 26

时间格式化

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

```
>>>time.strftime("%Y-%m-%d %H:%M:%S",t)
```



```
'2018-01-26 12:55:20'
```



```
>>>timeStr = '2018-01-26 12:55:20'
```

```
>>>time.strptime(timeStr, "%Y-%m-%d %H:%M:%S")
```

时间格式化

函数	描述
strptime(str, tpl)	<p>str是字符串形式的时间值</p> <p>tpl是格式化模板字符串，用来定义输入效果</p> <pre>>>>timeStr = '2018-01-26 12:55:20' >>>time.strptime(timeStr, "%Y-%m-%d %H:%M:%S") time.struct_time(tm_year=2018, tm_mon=1, tm_mday=26, tm_hour=12, tm_min=55, tm_sec=20, tm_wday=4, tm_yday=26, tm_isdst=0)</pre>

程序计时应用

程序计时

程序计时应用广泛

- 程序计时指测量起止动作所经历时间的过程
- 测量时间 : `perf_counter()`
- 产生时间 : `sleep()`

程序计时

函数	描述
perf_counter()	<p>返回一个CPU级别的精确时间计数值，单位为秒 由于这个计数值起点不确定，连续调用差值才有意义</p> <pre>>>>start = time.perf_counter() 318.66599499718114 >>>end = time.perf_counter() 341.3905185375658 >>>end - start 22.724523540384666</pre>

程序计时

函数	描述
sleep(s)	s拟休眠的时间，单位是秒，可以是浮点数 <code>>>>def wait():</code> <code> time.sleep(3.3)</code> <code>>>>wait() #程序将等待3.3秒后再退出</code>



单元小结



模块2: time库的使用

- 时间获取 : `time()` `ctime()` `gmtime()`
- 时间格式化 : `strftime()` `strptime()`
- 程序计时 : `perf_counter()` `sleep()`

