

## timeit&&cProfile

若输入语句时有多句话，请用';'隔开

```
--timeit.timeit(stmt='pass',setup='pass',timer=<default timer>,number = 1000000)
```

创建一个Timer实例，stmt是待测试的函数或语句，setup初始化或构建环境的导入语句，timer计时函数，number每一次测量中语句被执行的次数

```
--timeit.repeat(stmt='pass',setup='pass',timer=<default timer>,repeat=3,number = 1000000)
```

创建一个Timer实例，stmt是待测试的函数或语句，setup初始化或构建环境的导入语句，timer计时函数，number每一次测量中语句被执行的次数。

repeat是重复测量次数

```
--time.default_timer()
```

默认的计时器，能在不同平台提供最高精度的计时

```
--class timeit.Timer(stmt='pass',setup='pass', timer=<timer function>)
```

timeit中的唯一一个Timer实例

FE:

(使用timeit)

```
>>> import timeit
```

```
>>> timeit.timeit('char in text', setup='text = "I love FishC.com!"; char = "o"')
```

```
0.41440500499993504
```

```
>>> timeit.timeit('text.find(char)', setup='text = "I love FishC.com!"; char = "o"')
```

```
1.7246671520006203
```

(使用Timer实例)

```
>>> import timeit
```

```
>>> t = timeit.Timer('char in text', setup='text = "I love FishC.com!"; char = "o"')
```

```
>>> t.timeit()
```

```
0.3955516149999312
```

```
>>> t.repeat()
```

```
[0.40193588800002544, 0.3960157959998014, 0.39594301399984033]
```

(测试函数)

```
def test():
```

```
    """Stupid test function"""
```

```
    L = [i for i in range(100)]
```

```
if __name__ == '__main__':
```

```
    import timeit
```

```
    print(timeit.timeit("test()", setup="from __main__ import test"))
```

=====2017/3/12==cProfile=====

```
--cProfile.run('运行的函数'[,filename='...'])
```

filename是保存的运行信息