

```
id=11)
tasks/test_four.py:20: AssertionError
===== 1 failed, 3 passed in 0.08 seconds =====
```

`assert` 触发测试失败之后，代码片断下方显示的是本地变量 `t_after`、`t_beforeh`、`t_expected` 详细的值。

--tb=style 选项

--tb=style

`--tb=style` 选项决定捕捉到失败时输出信息的显示方式。某个测试用例失败后，`pytest` 会列举出失败信息，包括失败出现在哪一行、是什么失败、怎么失败的，此过程我们称之为“信息回溯”。大多数情况下，信息回溯是必要的，它对找到问题很有帮助，但有时也会对多余的信息感到厌烦，这时 `--tb=style` 选项就有用武之地了。我推荐的 `style` 类型有 `short`、`line`、`no`。`short` 模式仅输出 `assert` 的一行以及系统判定内容（不显示上下文）；`line` 模式只使用一行输出显示所有的错误信息；`no` 模式则直接屏蔽全部回溯信息。

继续使用上文中对于 `test_replace()` 的修改，利用它来看看各种失败回溯信息的显示方式。

使用 `--tb=no` 屏蔽全部回溯信息。

```
$ cd /path/to/code/ch1
$ pytest --tb=no tasks
===== test session starts =====
collected 4 items

tasks/test_four.py .F
tasks/test_three.py ..
===== 1 failed, 3 passed in 0.04 seconds =====
```

使用 `--tb=line`，它可以告诉我们错误的位置，有时这已经足够了（特别是在运行大量测试用例的情况下，可以用它发现失败的共性）。

```
$ pytest --tb=line tasks
===== test session starts =====
collected 4 items
```