

# Reporting

Fork me on GitHub

It is possible to generate any combination of the reports for a single test run.

The available reports are terminal (with or without missing line numbers shown), HTML, XML and annotated source code.

The terminal report without line numbers (default):

```
pytest --cov-report term --cov=myproj tests/

----- coverage: platform linux2, python 2.6.4-
final-0 -----
Name                Stmts   Miss  Cover
-----
myproj/___init___      2      0   100%
myproj/myproj        257     13    94%
myproj/feature4286     94      7    92%
-----
TOTAL                 353     20    94%
```

The terminal report with line numbers:

```
pytest --cov-report term-missing --cov=myproj tests/

----- coverage: platform linux2, python 2.6.4-
final-0 -----
Name                Stmts   Miss  Cover   Missing
-----
myproj/___init___      2      0   100%
myproj/myproj        257     13    94%   24-26, 99, 149,
233-236, 297-298, 369-370
myproj/feature4286     94      7    92%   183-188, 197
-----
TOTAL                 353     20    94%
```

The terminal report with skip covered:

```
pytest --cov-report term:skip-covered --cov=myproj tests/

----- coverage: platform linux2, python 2.6.4-
final-0 -----
Name                Stmts   Miss  Cover
-----
```

```
-----  
myproj/myproj          257    13   94%  
myproj/feature4286     94     7   92%  
-----  
TOTAL                  353    20   94%
```

1 files skipped due to complete coverage.

You can use **skip-covered** with **term-missing** as well. e.g. **--cov-report term-missing:skip-covered**

These three report options output to files without showing anything on the terminal:

```
pytest --cov-report html  
       --cov-report xml  
       --cov-report annotate  
       --cov=myproj tests/
```

The output location for each of these reports can be specified. The output location for the XML report is a file. Where as the output location for the HTML and annotated source code reports are directories:

```
pytest --cov-report html:cov_html  
       --cov-report xml:cov.xml  
       --cov-report annotate:cov_annotate  
       --cov=myproj tests/
```

The final report option can also suppress printing to the terminal:

```
pytest --cov-report= --cov=myproj tests/
```

This mode can be especially useful on continuous integration servers, where a coverage file is needed for subsequent processing, but no local report needs to be viewed. For example, tests run on Travis-CI could produce a .coverage file for use with Coveralls.