Python: coverage

> Coverage.py is a tool for measuring code coverage of Python programs. It monitors your program, noting which parts of the code have been executed, then analyzes the source to identify code that could have been executed but was not.
>
> Coverage measurement is typically used to gauge the effectiveness of tests. It can show which parts of your code are being exercised by tests, and which are not.

Instalación

Es preferible hacer la instalación en un entorno virtual

```
@tos:~/PPS/bloque_01/ejercicio1/palindromo$ python3 -m venv venv
@tos:~/PPS/bloque_01/ejercicio1/palindromo$ source
venv/bin/activate
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ pip freeze
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ pip install
coverage
Collecting coverage
  Using cached coverage-6.1.2-cp38-cp38-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.many
linux2010_x86_64.whl (216 kB)
Installing collected packages: coverage
Successfully installed coverage-6.1.2
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

Comprobamos la instalación

```
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ coverage help
Coverage.py, version 6.1.2 with C extension
Measure, collect, and report on code coverage in Python programs.

usage: coverage <command> [options] [args]
```

```
Commands:
    annotate    Annotate source files with execution information.
    combine     Combine a number of data files.
    debug       Display information about the internals of
coverage.py
    erase       Erase previously collected coverage data.
    help        Get help on using coverage.py.
    html        Create an HTML report.
    json        Create a JSON report of coverage results.
    report      Report coverage stats on modules.
    run         Run a Python program and measure code execution.
    xml         Create an XML report of coverage results.

Use "coverage help <command>" for detailed help on any command.
Full documentation is at https://coverage.readthedocs.io
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

Ejecución de tests bajo coverage

La ejecución normal de los tests con unittest sería

```
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ python -m
unittest  tests/test_palindromo.py
...........
-----------------------------------------------------------------
-----
Ran 11 tests in 0.000s

OK
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

y con coverage sustituimos `python` por `coverage run`

```
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ coverage run -
m unittest  tests/test_palindromo.py
...........
-----------------------------------------------------------------
-----
Ran 11 tests in 0.001s

OK
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

Reports

Para ver los resultados tenemos el comando report:

```
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ coverage
report
Name                           Stmts   Miss  Cover
----------------------------------------------
palindromo.py                     15      3    80%
tests/test_palindromo.py          36      0   100%
----------------------------------------------
TOTAL                             51      3    94%
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

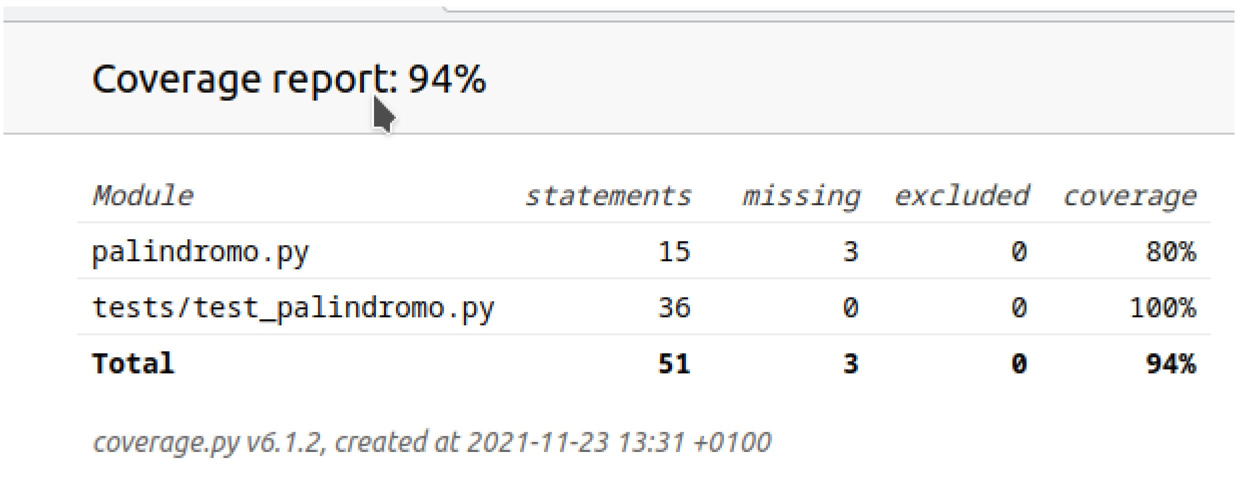
Si queremos una presentación más vistosa y acceder a algunos detalles adicionales podemos visualizar los resultados en HTML con `coverage html`. El report en el nuevo formato se almacena en el directorio htmlcov:

```
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
Wrote HTML report to htmlcov/index.html
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

Sólo nos queda invocar el navegador:

```
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$ firefox
htmlcov/index.html &
(venv) @tos:~/PPS/bloque_01/ejercicio1/palindromo$
```

Y este es el resultado:

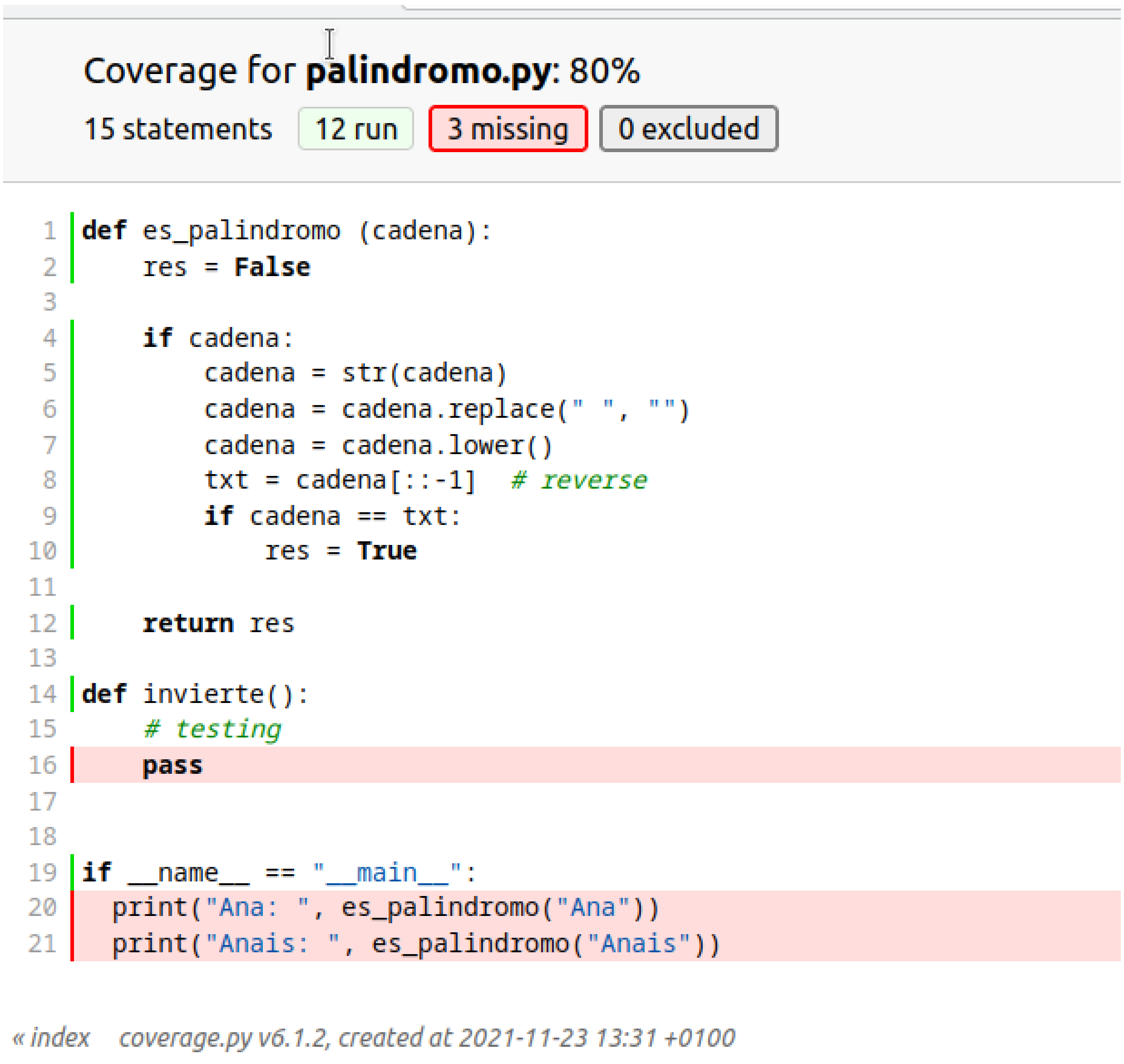


Coverage report: 94%

Module	statements	missing	excluded	coverage
palindromo.py	15	3	0	80%
tests/test_palindromo.py	36	0	0	100%
Total	**51**	**3**	**0**	**94%**

coverage.py v6.1.2, created at 2021-11-23 13:31 +0100

Si hacemos clic sobre el nombre del fichero tendremos detalles adicionales:



Coverage for **palindromo.py**: 80%

15 statements | 12 run | 3 missing | 0 excluded

```
def es_palindromo (cadena):
    res = False

    if cadena:
        cadena = str(cadena)
        cadena = cadena.replace(" ", "")
        cadena = cadena.lower()
        txt = cadena[::-1]   # reverse
        if cadena == txt:
            res = True

    return res

def invierte():
    # testing
    pass


if __name__ == "__main__":
  print("Ana: ", es_palindromo("Ana"))
  print("Anais: ", es_palindromo("Anais"))
```

« index   coverage.py v6.1.2, created at 2021-11-23 13:31 +0100

Enlaces

- https://coverage.readthedocs.io/en/6.1.2/