

## e) Code coverage

### Utilisation

“**coverage.py**” est un module de couverture de tests qui fonctionne avec le module `pytest` précédemment utilisé.

Celui-ci va nous indiquer le pourcentage de notre code qui est testé.

Une bonne couverture de test, supérieure à 80%, est signe d'un projet bien testé.

Il n'a pas besoin d'être testé à 100% dès lors que les tests ont été intelligemment mis en place.

Pour cela, on commencera par installer la librairie :

```
pip install coverage
pip install pytest-cov
```

Puis, on exécutera la commande suivante:

```
pytest --cov=program --cov-report html test_*.py
```

Cela signifie : “teste les fichiers contenus dans le dossier 'program', crée un rapport en html et utilise les tests qui sont ici-même et qui sont de la forme 'test\_[caractères].py”



Les rapports d'erreurs de “coverage” peuvent être de plusieurs types:

- HTML
- XML

### Couverture de tests

Une fois la commande terminée, un dossier se créera à l'endroit où la commande a été lancée et contiendra tout le rapport.

En cliquant sur `index.html`, on aura la page suivante :

#### Coverage report: 100%

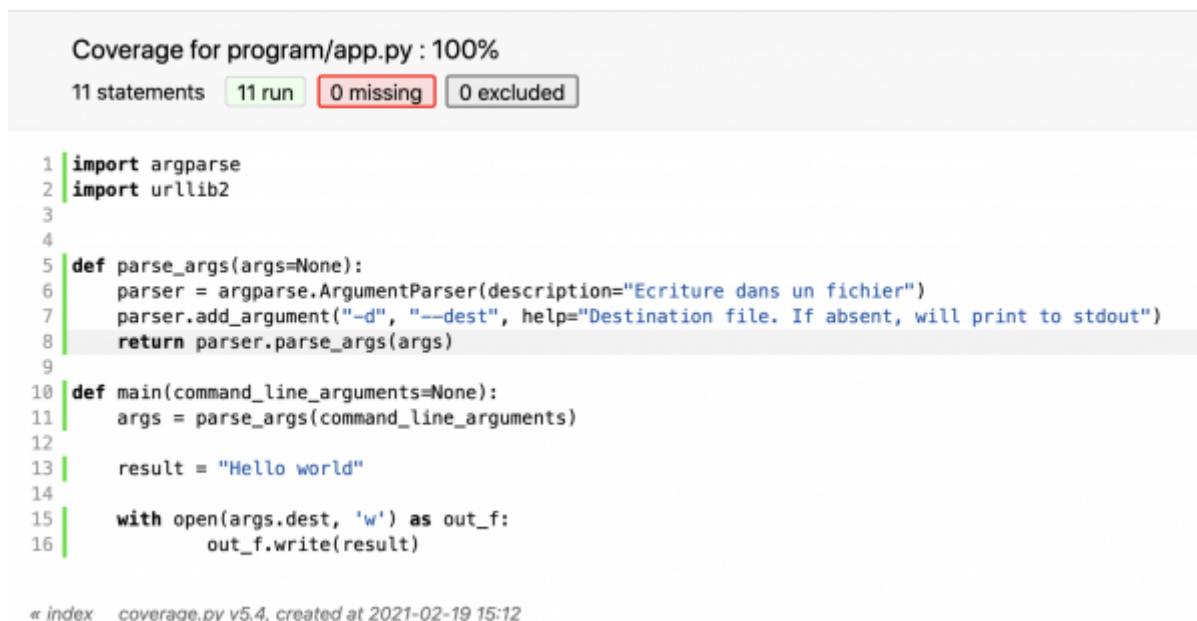
Module ↑	statements	missing	excluded	coverage
program/__init__.py	0	0	0	100%
program/app.py	11	0	0	100%
<b>Total</b>	<b>11</b>	<b>0</b>	<b>0</b>	<b>100%</b>

*coverage.py v5.4, created at 2021-02-19 15:12*

Sur la page, on a plusieurs infos:

- Coverage report : la couverture de l'ensemble des tests (il n'y en a que 1 dans notre cas)
- le détail pour chaque fichier testé : `app.py` est testé à 100%

On peut même cliquer sur `app.py` pour avoir plus d'informations :



Ainsi, on a :

- en vert, les parties du programme qui ont été testées
- en rouge, les parties du programme qui n'ont pas été testées (par chance, on en a pas !)

Si le format HTML offre une page de rapport toute faite, le format XML nous permettra de créer notre propre interface de visualisation des rapports de code coverage.