Documentation Contrall e) Code coverage

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 libraire :

```
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 100% 11 Total 0 100% 11

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%

Documentation Contrall e) Code coverage

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

```
Coverage for program/app.py: 100%
   11 statements 11 run 0 missing 0 excluded
  import argparse
2 import urllib2
5 def parse_args(args=None):
      parser = argparse.ArgumentParser(description="Ecriture dans un fichier")
       parser.add_argument("-d", "--dest", help="Destination file. If absent, will print to stdout")
7
8 return parser.parse_args(args)
10 def main(command_line_arguments=None):
11
       args = parse_args(command_line_arguments)
12
13
       result = "Hello world"
14
15
       with open(args.dest, 'w') as out_f:
16
               out_f.write(result)
« index coverage.py v5.4, created at 2021-02-19 15:12
```

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 faîte, le format XML nous permettra de créer notre propre interface de visualisation des rapports de code coverage.