Initiation à la programmation Python
Une première analyse exploratoire de l'usage des tests

Université de Lille, Laboratoire CRIStAL, équipe NOCE
Mirabelle Nebut - Yvan Peter

atelier APIMU - EIAH'23

Topic

Genèse de ce travail : dpt Info de ULille

En L1 portail SESI[1], UE initiation à la programmation Python, des étudiant·es qui...

- se jettent sur le code sans réflexion préalable
- enchaînent les exercices sans avoir exécuté leur code
- demandent à l'enseignant de valider leur code

En master Informatique, des étudiant·es qui...

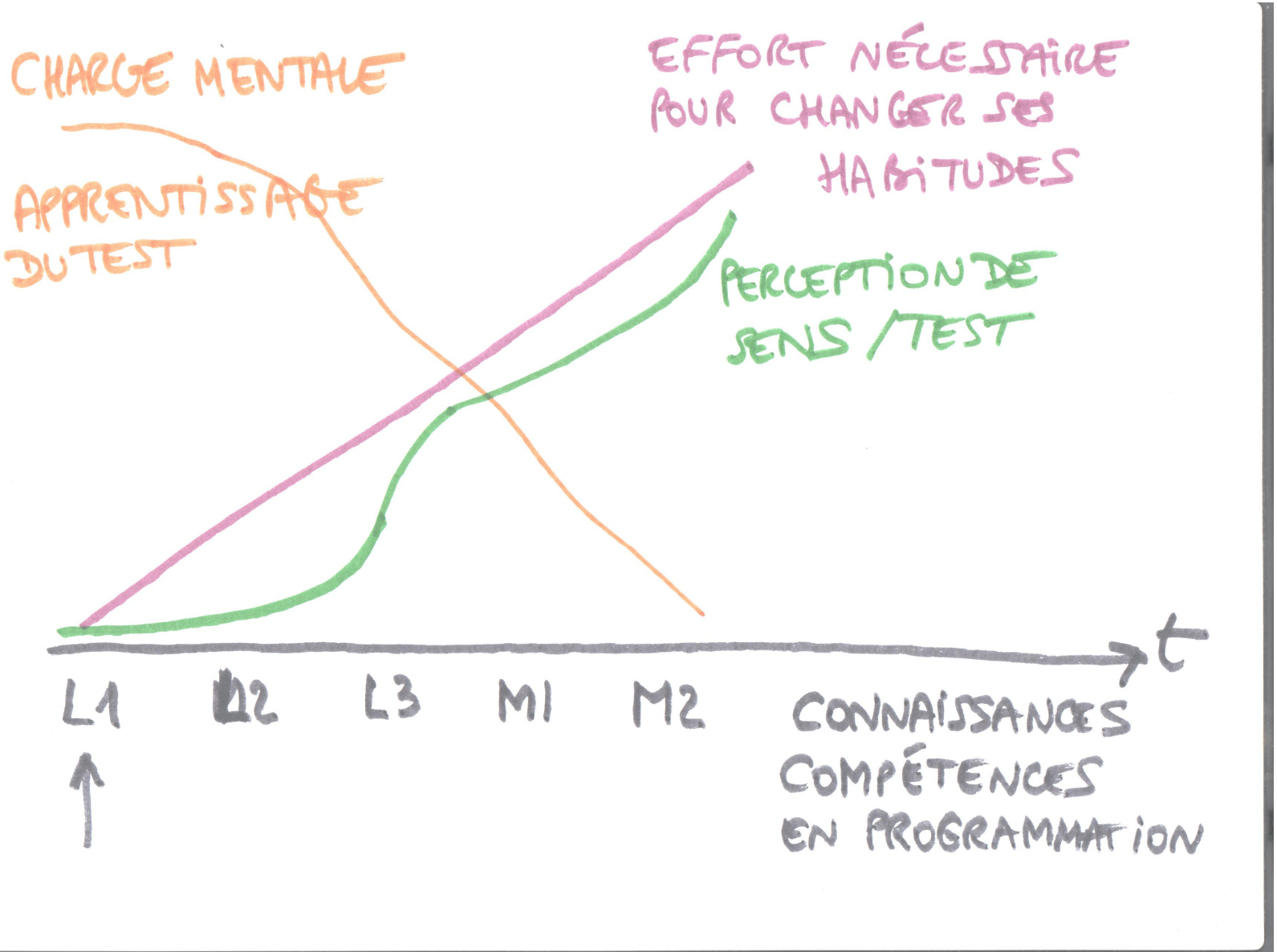
- après avoir suivi plusieurs cours traitant de test du logiciel
- spontanément n'utilisent pas les tests dans leurs projets

⇒ Prendre l'habitude de tester les programmes dès la L1 ?
⇒ Quel impact sur les comportements de programmation ?

[1] Sciences Exactes et Sciences de l'Ingénierie

Problématique(s)

Garousi et al., 2020, Scatalon et al., 2019



Contexte de l'UE d'initiation à la prog avec Python

IDE Thonny dédié à l'apprentissage :

- vues adaptées aux débutant·es, débogueur simple d'accès
- console Python : toutes les interactions avec Python se font dans Thonny
- (+ extensible par un mécanisme de greffon)

Depuis plusieurs années :

- fourniture d'exemples dans la docstring, exécutables avec `doctest`
- peu de moyens mis en œuvre, outil pas si adapté car sémantique pas toujours intuitive ⇒ succès très mitigé

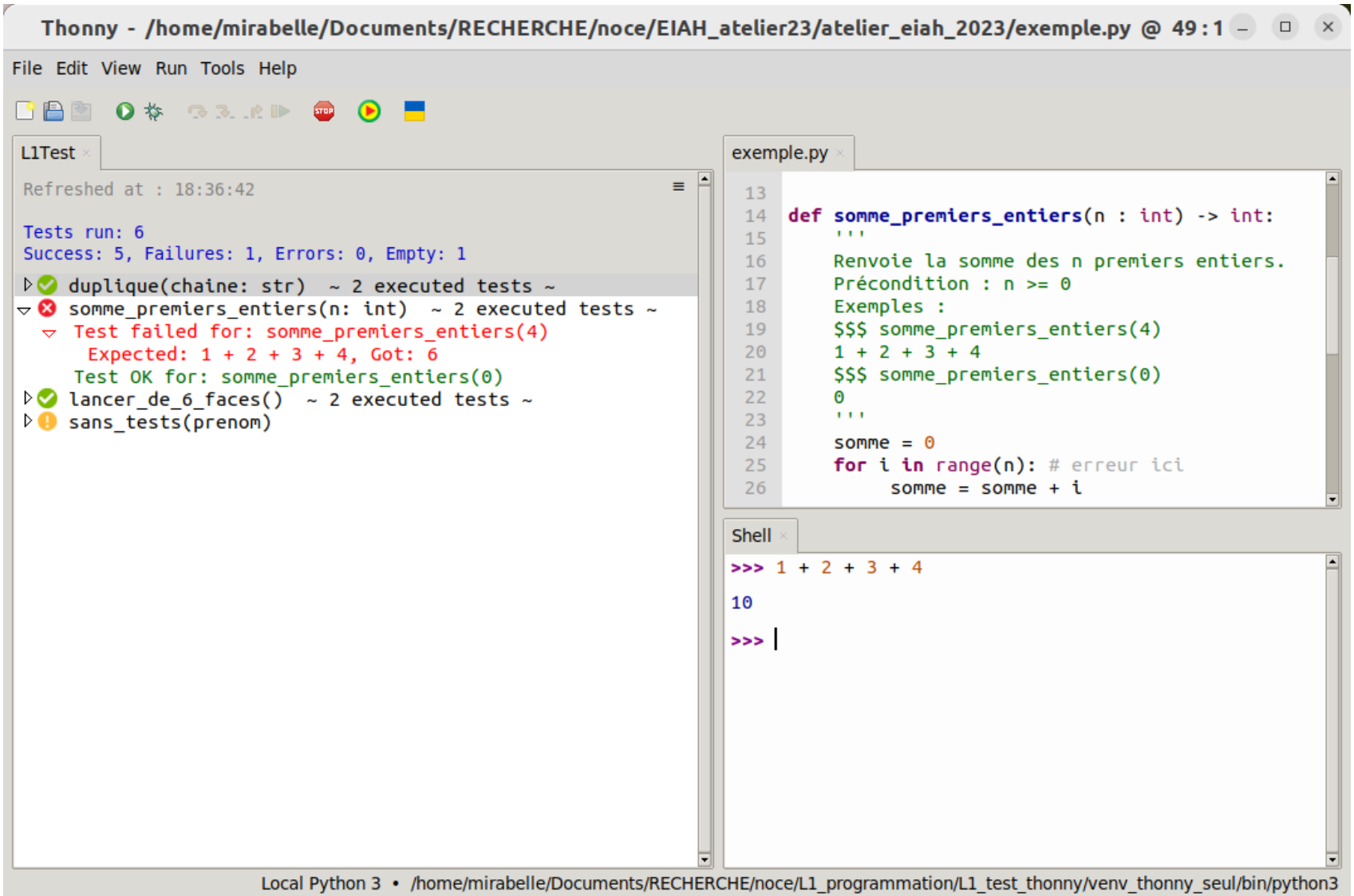
⇒ Besoin d'un autre outil
⇒ Qui intégre à Thonny l'écriture et l'exécution des tests

Outils de test : adéquation à l'initiation à la programmation

	assert	doctest	pytest	outils xUnit	L1test (dans l'idéal)
écriture des tests	++	++	−	− −	++
retour	−	− −	+	++	++
outil dédié au test	−	+−	++	++	++

Topic

Thonny + L1test



Syntaxe et sémantique des tests avec L1test

Tests = exemples exécutables très simples.

Même principe que doctest pour la syntaxe :

- tests écrits dans la docstring des fonctions
- en imitant la syntaxe de l'interpréteur Python

```
$$$ <expression_à_évaluer>
<expression_résultat_attendu>
```

À noter :

- le résultat attendu peut être décrit par une expression
- comparaison intuitive par == entre résultat attendu et résultat réel
- impossible de tester une procédure d'affichage (renvoie `None`)

Résultat de l'exécution des tests avec L1test

Nouveau bouton pour lancer les tests + fenêtre dédiée

Code couleur hérité des outils standard de test unitaire :

- vert pour un test qui passe
- rouge pour un échec / une erreur

orange pour une fonction sans tests

Topic

L1log : Principes

Événement Tkinter généré par Thonny pour chaque interaction avec l'interface graphique.

Greffon de collecte de log existant sous Thonny : traces trop fines

⇒ L1log :

- filtrage et abstraction des événements Tkinter
- envoi en temps réel dans un LRS au format xAPI[2]
- utilisation du vocabulaire de ProgSnap2[3], spécifique aux activités liées à la programmation

[2] `https://xapi.com/`

[3] Price, T. W., Hovemeyer, D., Rivers, K., Gao, G., Bart, A. C., Kazerouni, A. M., Becker, B. A., Petersen, A., Gusukuma, L., Edwards, S. H., and Babcock, D. (2020). Progsnap2 : A flexible format for programming process data. In Proceedings of the 2020 ACM *Conference on Innovation and Technology in Computer Science Education*, pages 356-–362. ACM.

Exemple de trace xAPI pour un test


```
{
 "timestamp":{"$date":"2022-10-07T11:15:44.981Z"},
 "actor":
  {
   "openid":"https://www.cristal.univ-lille.fr/users/281b59ea7d35e20",
   "objectType":"Agent"
  },
 "verb":{"id":"https://www.cristal.univ-lille.fr/verbs/Run.test"},
 "object":
 {
  "id":"https://www.cristal.univ-lille.fr/objects/Test",
  "objectType":"Activity",
  "extension":{"https://www.cristal.univ-lille.fr/objects/File/Filename":"file_-8712108631165266730",
  "https://www.cristal.univ-lille.fr/objects/Test/TestedExpression":"maximum(4, 1)",
  "https://www.cristal.univ-lille.fr/objects/Test/TestedLine":27,
  "https://www.cristal.univ-lille.fr/objects/Test/TestedFunction":"maximum"}
 },
 "result":
 {
  "success":true,
  "extension":{"https://www.cristal.univ-lille.fr/objects/Test/expectedResult":"4",
  "https://www.cristal.univ-lille.fr/objects/Test/obtainedResult":"4"}
 }
}
```

Propriétés collectées

- identifiant utilisateur (hashé)
- identifiant session
- horodotage de l'événement
- type de l'événement (Run.Program, Run.Test, Run.Command etc)
- résultat de l'action (erreur ?)
- (si besoin) contenu de l'éditeur au moment de l'action
- (si besoin) sortie produite lors de l'exécution d'un programme, des tests ou de l'utilisation de la console

Topic

Contexte de la collecte (1er semestre 22-23)

- les groupes de 2 amphis ont utilisé L1test + L1log avec présentation de l'outil + comment écrire des tests
- les autres ont utilisé `doctest` sans collecte
- bugs pour L1test et L1log

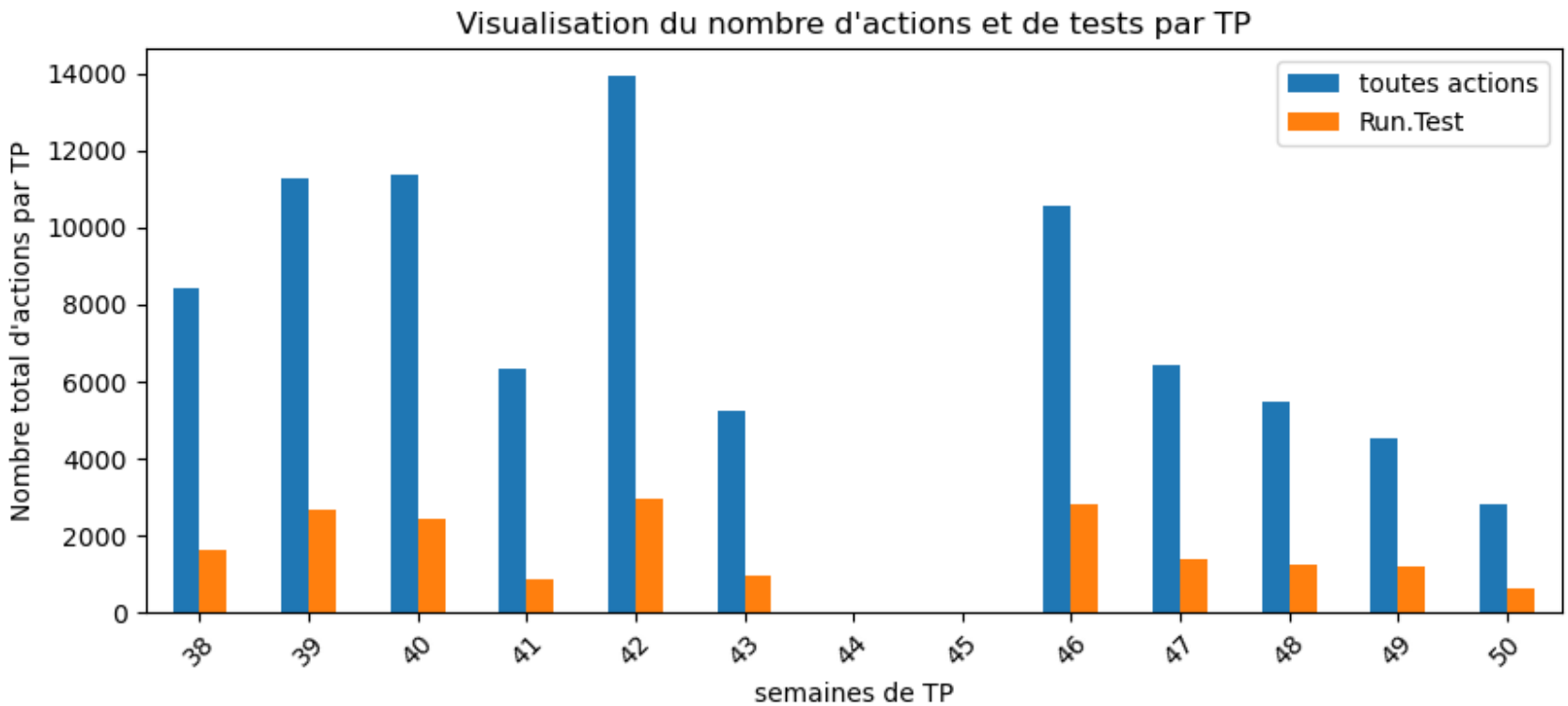
Ressenti et passage de questionnaire :

- bon accueil de L1test par les étudiant·es et les enseignant·es
- $\Rightarrow$ initier au test basique pendant l'initiation à la programmation en L1 est possible
- incompréhension de certain·es étudiant·es par rapport au test, vu comme une alternative à la réflexion sur le code
- plus-value du test pas bien mise en évidence

Analyse des traces

Aucun résultat exact pour le moment.

162 étudiant·es (ont cliqué au moins une fois pour lancer les tests)



Topic

Nouvelle collecte à la rentrée 2023

- Portail SESI $\rightarrow$ portail MI
- Utilisation généralisée de L1test à toute la promo
- Collecte nominative à usage pédagogique

Ce qu'on aimerait évaluer

Effet du test sur l'activité de programmation :

- focus sur la spécification et le comportement attendu ?
- focus sur la correction du code ?
- incite à réfléchir au code ou à faire des essais/erreurs ?
- forme des programmes (return/print, découpage en fonctions) ?

Quelle dose de test introduire :

- donner / faire écrire des tests ?
- comment expliquer ce qu'est "un bon" test ?
- écrire les tests avant le code ?

Impact sur l'apprentissage du test en POO en L2 ?

C'est fini !