

# Rust : Tests

Qualité de Développement — R4.02

---

C. Raïevsky

2023-2024



Département Informatique

*BUT Informatique 2<sup>ème</sup> année*

## Tests en Rust

- Les tests sont du code Rust standard
- Pas de bibliothèque externe
- Intégrés dans les outils : `cargo test`

### Deux principaux types de tests supportés

- Tests unitaires
- Tests d'intégrations

# Un test est une fonction

## Fonction

- Annotée : `#[test]`
- Le test passe si la fonction se termine
- Le test échoue si elle panique

```
1 #[test]
2 fn it_works() {
3     let result = add(2, 2);
4     assert_eq!(result, 4);
5 }
```

## Deux assertions :

- `assert!(e);` → panique si 'e' est faux
- `assert_eq!(e1, e2);` → panique si 'e1' et 'e2' sont différents

## Organisation des tests

---

### Tests unitaires

- Placés au plus près du code qu'ils testent
- Dans un sous module dédié
- Dans le même fichier ou dans un module à part

### Tests d'intégration

- Au niveau du *package*
- Dans des modules dédiés

## Tests unitaires

### Exemple minimal (cargo new --lib)

```
1 pub fn add(left: usize, right: usize) -> usize {
2     left + right
3 }
4
5 #[cfg(test)]
6 mod tests {
7     use super::*;
8
9     #[test]
10    fn it_works() {
11        let result = add(2, 2);
12        assert_eq!(result, 4);
13    }
14 }
```

## Tests d'intégration

### À la racine du *package*

- Dans un répertoire dédié
- Compilés et exécutés uniquement lors des tests
- Répartition dans des fichiers au choix

```
adder
├── Cargo.lock
├── Cargo.toml
├── src
│   └── lib.rs
└── tests
    ├── integration_tests.rs
    └── other_tests.rs
```

## Tests d'intégration

### Exemple - integration\_tests.rs

```
1 use adder;
2
3 #[test]
4 fn it_adds_two() {
5     assert_eq!(4, adder::add(2, 2));
6 }
```

## Les exemples sont des tests

---

### Les exemples de code présents dans la documentation

- Sont exécutés lors de la phase de test
- Permettent d'assurer la cohérence entre la documentation et le code.



## Tests dans la documentation

### Exemple comme test

```
1  /// A wonderful function that computes the sum of its arguments.
2  /// # Example usage:
3  /// ```
4  /// let sum = package::add(11, 31);
5  /// assert_eq!(sum, 42);
6  /// ```
7  pub fn add(left: usize, right: usize) -> usize {
8      left + right
9  }
```

Le code des lignes 4 et 5 va être passé comme un test lors d'un "cargo test"