

CC00: Controle de connaissance en programmation avancée

Axel Viala <axel@darnuria.eu>

17 juin 2022

Nom et Prénom : _____

Groupe : L2 _____

Notation : Les réponses fausses en QCM font perdre des points.

Attendus : Questions basées sur des concepts abordés dans les 4 premiers chapitre du livre "The Rust Programming Language", Rustlings jusqu'à `quiz1.rs` et le cours.

1 Rust syntaxe, typage, concepts de base

1. En Rust, avez-vous à écrire les types par vous-même :
 - ☐ Oui, comme en C
 - ☐ Non, le compilateur infère les types et parfois me demande du secours
 - ☐ Comme en JavaScript ou Python, les types sont gérés à l'exécution
2. Expliquez avec vos mots ce qu'est un type et le typage. Exemples de types : `i32`, `bool` ou même `String`.

2 Syntaxe de Rust

3. Expliquez à quoi sert le mot clef `let` donner un exemple court. Bonus : expliquez avec vos mots le destructuring.

4. En Rust les variables sont ? (exemple : `let val = 42;`)
 - ☐ *Immutable* : ne peuvent pas changer de valeur.
 - ☐ *Mutable* : peuvent changer de valeur.
5. Le code suivant peut-il compiler ? Expliquer pourquoi et comment corriger si il ne compile pas.

```
let age = 29;  
age += 1;
```

6. Anatomie de la signature (ou type) d'une fonction. Combien de paramètres accepte la fonction ? Quel est le type de `a` ? quel est le nom de la fonction ? Quel est le type de retour de la fonction ?

```
fn ultimate_function(a: i32, b: i32) -> bool { a == b }
```

—
—
—
—

7. Voici le code mystère suivant : Aide mémoire : `""` est de type `&str`.

```
let love_chocolate = true;  
let a = if love_chocolate {  
    "Chocolate";  
} else {  
    "Raspberry";  
};  
println!("{:?}", a);
```

De quel type est `a` il y a t'il une faute de frappe ? Quelle correction peut-être appliquée ?

8. Super Bonus : Expliquer avec vos mots le concept de *move semantics* (capture ou déplacement en Français).