



Dans le langage RUST existe-t-il des MACROS ?

Oui, le langage Rust dispose d'un système puissant de macros pour la métaprogrammation.

Types de Macros

Rust propose deux grandes catégories de macros. ¹ ²

Les **macros déclaratives** utilisent `macro_rules!` pour générer du code via des motifs similaires à `match`, comme la macro `vec!` qui crée un vecteur. ³ ¹

Elles s'expansent en arbres syntaxiques abstraits à la compilation, évitant les pièges de préprocesseurs comme en C. ⁴ ²

Macros Procédurales

Ces macros agissent comme des fonctions sur les tokens. ⁵ ¹

Elles incluent les `#[derive]` personnalisés, les macros attributs (ex. `#[ma_macro]`) et celles ressemblant à des fonctions. ⁶ ³

Utilisez des crates comme `syn` et `quote` pour les implémenter.¹

✱

<https://jmskapt.github.io/rust-book-fr/ch19-06-macros.html>

<https://doc.rust-lang.org/rust-by-example/macros.html>

<https://doc.rust-lang.org/book/ch20-05-macros.html>

https://web.mit.edu/rust-lang_v1.25/arch/amd64_ubuntu1404/share/doc/rust/html/book/first-edition/macros.html

<https://rustwiki.org/en/reference/macros.html>

<https://blog.guillaume-gomez.fr/Rust/3/1>

<https://rust.developpez.com/tutoriels/rust-par-l-exemple/?page=macro-rules>

<https://lafor.ge/macro-rust/>

https://www.reddit.com/r/rust/comments/t0gogu/macros_in_rust/?tl=fr

<https://blog.guillaume-gomez.fr/Rust/2/13>