

# MODULE 101

# INTRODUCTION



# A PROPOS DE RUST

- Existe depuis 2006.
- Version 1.0 en 2015.
- Plus grande force : Sûreté.
- Plus grande faiblesses : Temps d'adaptation pour rentrer dans le mode de pensée adéquat.

# POSITIONNEMENT

- Langage système, jusqu'à l'embarqué,
- Multiples paradigmes de programmation,
- Sûreté de la mémoire et du multithreading,
- Performance avec des abstractions à 0 coût,
- Support moderne (compilateur, éditeurs, gestionnaire de paquets, etc.),
- Demande une structuration des programmes souvent différente :
  - Data-oriented design

# DOCUMENTATION

Le langage Rust et son écosystème sont très bien documentés :

- Site web : <https://www.rust-lang.org/>
- Rust book : <https://doc.rust-lang.org/book/>
- Tous les liens sur : <https://www.rust-lang.org/learn>

# DOCUMENTATION

Tester en ligne !

- <https://play.rust-lang.org/>

# COMMUNAUTÉ



La communauté Rust est très accueillante. Voir

<https://www.rust-lang.org/community>

- Forum : <https://users.rust-lang.org/>
- Discord : <https://discordapp.com/invite/rust-lang>
- Reddit : <https://www.reddit.com/r/rust>

# CHAÎNE D'OUTILS

- Gestion des chaînes : `rustup`
- Compilateur : `rustc`
- Gestionnaire de paquets / système de build :  
`cargo`

# CARGO

- Formatter du code: `cargo fmt`
- Compiler: `cargo build`
- Compiler (release): `cargo build --release`
- Vérifier seulement (pas de binaire): `cargo check`
- Exécuter les tests: `cargo test`
- Générer la doc: `cargo doc`
- Executer `cargo run`
  - .. ou juste exécuter:
  - `./target/debug/my_program`