

Indice

Symboles

! opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Opérateurs unaires](#)

!= opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Comparaisons d'équivalence](#)

Attribut #![feature] , [Attributs](#)

#[allow] attribut , [Attributs](#)

#[cfg] attribut , [Attributs](#) , [Fonctionnalités spécifiques à la plate-forme](#)

Attribut #[derive] , [diffusion de pages sur le Web](#)

Attribut #[inline] , [Attributs](#)

Attribut #[link] , [Utilisation des fonctions des bibliothèques](#)

Attribut #[repr(C)] , [Recherche de représentations de données communes](#)

Attribut #[repr(i16)] , [Recherche de représentations de données communes](#)

Attribut #[should_panic] , [Tests et Documentation](#)

#[test] attribut , [Attributs](#)

\$ (invite de commande) , [rustup et Cargo](#)

Opérateur % , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

& opérateur , [Gestion des arguments de ligne de commande](#) , [Références aux valeurs](#) , [Opérateurs arithmétiques, binaires, de comparaison et logiques](#) , [Modèles de référence](#)

& modèle , [Modèles de référence](#)

Opérateur && , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

Opérateur &mut , [Références aux valeurs](#)

&mut type , [mut et mutex](#)

&mut [T] type , [Références](#)

&str (tranche de chaîne) , [Chaînes en mémoire](#)

Type &[T] , [Références](#)

* opérateur

accéder à la valeur référencée , [Opérateurs de référence](#)

déréférencement , [Gestion des arguments de ligne de commande](#) ,
[Références Rust versus références C++](#) , [Deref et DerefMut](#) , [Pointeurs bruts](#)
multiplication , [arithmétique, au niveau du bit, comparaison et opérateurs logiques](#)
surcharge de , [Deref et DerefMut](#)
correspondance de motifs et , [Motifs de référence](#)
* caractère générique, pour les versions de caisse , [Versions](#)
*const T , [pointeurs bruts](#)
*mut T , [pointeurs bruts](#)
+ , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) ,
[Opérateurs binaires](#) , Ajout [et insertion de texte](#)
- opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Opérateurs unaires](#)
. opérateur , [Rust References Versus C++ References](#) , [Fields and Elements](#) , [Deref et DerefMut](#)
.. opérateur , [Champs et Eléments](#)
Opérateur ..= , [Champs et Eléments](#)
/ opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)
/// (commentaires de la documentation) , [Ce qu'est réellement l'ensemble de Mandelbrot](#) , [Documentation](#)
:: opérateur , [chemins et importations](#)
::<...> (symbole turbofish) , [Appels de fonction et de méthode](#) , [Structures génériques](#)
< opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Comparaisons ordonnées](#) , [Pointeurs bruts](#)
<< opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)
Opérateur <= , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Comparaisons ordonnées](#)
= opérateur , [Affectation](#)
Opérateur == , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Comparaisons d'équivalence](#) , [Pointeurs bruts](#)
=> opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)
> opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Comparaisons ordonnées](#)
>= opérateur , [comparaisons ordonnées](#)

>> opérateur , opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

? opérateur , [propagation des erreurs](#)

@ Patterns , [Reliure avec @ Patterns](#)

^ opérateur , opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

Paramètre de format {:?} , [Formatage des valeurs pour le débogage](#)

Paramètre de format {:p} , [Formatage des pointeurs pour le débogage](#)

| (barre verticale) dans les motifs correspondants , [Possibilités multiples de correspondance](#)

| operator , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

|| operator , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

~ opérateur , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

UN

abandon , [Abandon](#)

chemin absolu , [chemins et importations](#)

"L'abstraction et le modèle de machine C++" (Stroustrup) , [et pourtant la rouille est toujours rapide](#)

méthodes d'accumulation pour les itérateurs , [Accumulation simple : count, sum, product](#) , [fold](#) et [rfold](#)

Framework actix-web , [Servir des pages sur le Web](#) - [Servir des pages sur le Web](#) , [Rappels](#) , Mise en [réseau](#)

méthodes d'adaptation

énumérer , [Un programme de Mandelbrot simultané](#)

fusible , [Futures](#)

méthodes d'adaptation pour les itérateurs , [Adaptateurs](#) d'itérateurs - [cycle](#)

by_ref , [by_ref](#) - [by_ref](#)

chaîne , [chaîne](#)

cloné , [cloné, copié](#)

copié , [cloné, copié](#)

cycler , [cycler](#)

énumérer , [énumérer](#) , [zip](#)

filter_map et flat_map , [filter map et flat map](#) - [filter map et flat map](#)

aplatir , [aplatir](#) - [aplatir](#)
fusible , [fusible](#)
inspecter , [inspecter](#)
mapper et filtrer , [mapper et filtrer](#) - [mapper et filtrer](#)
visible , [visible](#)
itérateurs réversibles et rev , [Itérateurs réversibles et rev](#) - [Itérateurs réversibles et rev](#)
sauter et sauter_pendant , [sauter et sauter_pendant](#)
prendre et prendre_pendant , [prendre et prendre_pendant](#) , [by_ref](#)
fermeture éclair , [fermeture éclair](#)
méthodes d'adaptation pour les lecteurs
méthode des octets , [Lecteurs](#)
méthode de la chaîne , [Lecteurs](#)
prendre la méthode , [Lecteurs](#)
types de données algébriques , [Enums et Patterns](#)
valeur d'alignement, requise par les types , les [tailles de caractères et les alignements](#)
Fonction align_of , [Tailles et alignements des caractères](#)
Fonction align_of_val , [Tailles de type et alignements](#)
all, iterator method , [any and all](#)
#[allow] attribut , [Attributs](#)
any, iterator method , [any et all](#)
de toute façon la caisse de gestion des erreurs , [Travailler avec plusieurs types d'erreurs, Types d'erreurs et de résultats](#)
Type de pointeur d'arc , [Rc et Arc : Propriété partagée](#) - [Rc et Arc : Propriété partagée](#) , [Passage de soi en tant que boîte, Rc ou Arc](#) - [Passage de soi en tant que boîte, Rc ou Arc](#) , [Partage de données immuables entre les threads](#) , [Mutex<T>](#)
fonction args , [Gestion des arguments de ligne de commande](#) , [skip et skip_while](#)
Type d'arguments, pour le formatage de chaîne , [Utilisation du langage de formatage dans votre propre code](#)
opérateurs arithmétiques , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , Opérateurs [arithmétiques et binaires](#) - Opérateurs [arithmétiques et binaires](#)
opérateurs d'affectation composés , [Opérateurs d'affectation composés](#) - Opérateurs d' [affectation composés](#)
surcharge , [opérateurs binaires](#)

arithmétique, pointeur , [Arithmétique du pointeur](#) - [Arithmétique du pointeur](#)

modèles de tableau , modèles de [tableau et de tranche](#)
tableaux , [tableaux](#)

concaténation de tableaux de , [Rejoindre](#)

pointeurs bruts vers , [Pointer Arithmetic](#) - [Pointer Arithmetic](#)

tranches et , [tranches](#)

tuples contre , [Tuples](#)

tableaux, joindre des tableaux de , [Joindre](#)

en tant qu'opérateur , [Conversions vers et à partir d'entiers](#)

Caractères ASCII , [Types d'entiers](#) , [ASCII, Latin-1 et Unicode](#) , [Caractères de classification](#) , [Traitement des chiffres](#)

Type de chaîne Ascii, code non sécurisé pour la conversion en chaîne ,
[Exemple : Un type de chaîne ASCII efficace](#) - [Fonctions non sécurisées](#)

Trait AsMut , [AsRef et AsMut](#)

Caractéristique AsRef , [AsRef et AsMut](#) , [OsStr et Path](#)

affirmer! macro , [Fonctions Rust](#) , [Tests et Documentation](#)

assert_eq ! macro , [Tests et documentation](#) , [Macros](#) , [Principes de base des macros](#) - Principes de base des macros

mission

C++ versus Rust , [Moves](#) - [Moves](#)

opérateurs d'affectation composés , [Affectation](#) , [Opérateurs](#) d'affectation composés - Opérateurs d' [affectation composés](#)

expressions , [Affectation](#)

coups et (voir coups)

Python contre Rust , [Moves](#) - [Moves](#)

références , [Attribuer des références](#)

dans Rust , [plus d'opérations qui bougent](#) - [Déplacements et contenu indexé](#)

à une variable , [Plus d'opérations qui bougent](#)

opérateurs d'affectation , [Affectation](#)

Const associés , Const associés , [Const associés](#)

fonctions associées , [Définir des méthodes avec impl](#)

types associés , [Types associés \(ou Fonctionnement des itérateurs\)](#) -
[Types associés \(ou Fonctionnement des itérateurs\)](#)

associativité , [priorité et associativité](#)

fonctions asynchrones , fonctions asynchrones [et expressions d'attente](#)
- [Appel de fonctions asynchrones à partir de code synchrone : blo-](#)

[ck on](#) , [création de fonctions asynchrones à partir de blocs asynchrones](#)

Blocs de déplacement asynchrones , [Blocs asynchrones](#)
flux asynchrones , [Réception de paquets : plus de flux asynchrones](#) -
[Réception de paquets : plus de flux asynchrones](#)
async-std crate , [Programmation asynchrone](#) , [Futures](#) , [Fonctions](#)
asynchrones et expressions d'attente , Création de [tâches asynchrones](#)
programmation asynchrone , [Programmation asynchrone](#) - [Quand le](#)
[code asynchrone est-il utile ?](#)

blocs asynchrones , [Programmation asynchrone](#) , [Blocs asyn-](#)
[chrones](#) - [Création de fonctions asynchrones à partir de blocs](#)
[asynchrones](#)

fonctions asynchrones , fonctions asynchrones [et expressions d'at-](#)
[tente](#) - [Appel de fonctions asynchrones à partir de code synchrone :](#)
[block on](#) , [création de fonctions asynchrones à partir de blocs](#)
[asynchrones](#)

client et serveur , [Un client et un serveur asynchrones](#) - [Groupes de](#)
[discussion : canaux de diffusion de tokio](#)

par rapport à la programmation synchrone , [De synchrone à asyn-](#)
[chrone](#) - [Un vrai client HTTP asynchrone](#)

futurs et exécuteurs, coordination , [Primitive Futures and Execu-](#)
[tors: Quand un futur mérite-t-il d'être interrogé à nouveau?](#) - [Im-](#)
[plémentation de block on](#)

Caisse de client HTTP , [un vrai client HTTP asynchrone](#)
épingler des contrats à terme , [Épingler](#) - [Le trait de détachement](#)
tâches versus threads traditionnels , [Programmation asynchrone](#)
situations utiles pour , [Quand le code asynchrone est-il utile ?](#) -
[Quand le code asynchrone est-il utile ?](#)

méthode as_mut_ptr , [pointeurs bruts](#)

méthode as_ptr , [pointeurs bruts](#)

opérations sur les entiers atomiques , [variables globales](#)

nombre de références atomiques (voir type de pointeur d'arc)

Types et opérations atomiques , Atomiques , Variables [globales](#)

attributs , [Ecrire et exécuter des tests unitaires](#) , [Attributs](#) - [Attributs](#)

await expressions , [Async Functions et Await Expressions](#) - [Appel de](#)
[fonctions asynchrones à partir de code synchrone : block on](#) , Généra-
tion de [tâches](#) asynchrones - Génération de tâches [asynchrones](#)

fil d'arrière-plan , [Concurrence](#)

contre-pression

client asynchrone et serveur de discussion , [Groupes de discussion : canaux de diffusion de tokio](#) - [Groupes de discussion : canaux de diffusion de tokio](#)

approche pipeline , [fonctionnalités et performances du canal](#)
commande bat , [systèmes de fichiers et outils de ligne de commande](#)
entrée/sortie binaire , [données binaires, compression et sérialisation](#)
littéral numérique binaire , [types entiers](#)

opérateurs binaires , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , Opérateurs [binaires](#)

binaire, formatage des nombres dans , [Formatage des nombres](#)

BinaryHeap , [Vue](#) d'ensemble , [BinaryHeap<T>](#) - [BinaryHeap<T>](#)

Type BinaryTree , [Énumérations génériques](#) , Remplir [un arbre binaire](#) , [Implémentation de vos propres itérateurs](#) - [Implémentation de vos propres itérateurs](#)

bindgen crate , [une interface brute pour libgit2](#)

opérateurs au niveau du bit , Opérateurs [arithmétiques, au niveau du bit, de comparaison et logiques](#) , Opérateurs [d'affectation](#) composés -
Opérateurs [d'affectation composés](#)

blocs , [blocs et points-virgules](#) - [Déclarations](#)

asynchrone , [Programmation asynchrone](#) , [Blocs](#) asynchrones -
[Création de fonctions asynchrones à partir de blocs asynchrones](#)
déclarations dans , [Déclarations](#)

extern block , [Déclarer des fonctions étrangères et des variables](#) -
[Déclarer des fonctions étrangères et des variables](#)

bloc impl , [Définir des méthodes avec impl](#) - [Fonctions associées au type](#)

unsafe , [Raw Pointers](#) , [Unsafe Code](#) , [Unsafe Blocks](#) - [Exemple : Un type de chaîne ASCII efficace](#) , [Unsafe Block ou Unsafe Function ?](#)

block_on , [Appel de fonctions asynchrones à partir de code synchrone : block_on](#) - [Appel de fonctions asynchrones à partir de code synchrone : block_on](#) , [Futurs primitifs et exécuteurs : quand un futur mérite-t-il d'être interrogé à nouveau ?](#) , [Implémenter block_on](#) - [Implémenter block_on](#)

Type booléen (bool) , [Le type bool](#) , [Mise en forme d'autres types](#)
Emprunter le trait , [Emprunter et EmprunterMut](#) - [Emprunter et EmprunterMut](#)

Emprunter<str> trait , [Emprunter sous d'autres types de texte](#)

emprunt , [Références](#)

futurs et , Génération de [tâches asynchrones](#) , [Les deux étapes de la vie d'un futur](#) - [Les deux étapes de la vie d'un futur](#)

itération et , [Gestion des arguments de la ligne de commande](#) , [by_ref](#)

variables locales et , [Emprunter une variable locale](#) - [Emprunter une variable locale](#)

syndicats , syndicats [emprunteurs](#)

valeurs d'expressions arbitraires , [Emprunter des références à des expressions arbitraires](#)

Trait BorrowMut , [Emprunter et EmprunterMut](#) - [Emprunter et EmprunterMut](#)

limites, rétro-ingénierie , [Reverse-Engineering Bounds](#) - [Reverse-Engineering Bounds](#)

Type de boîte , [Boîtes](#) , [Propriété](#) , [Se faire passer pour une boîte](#), Rc ou [Arc](#) - [Se passer pour une boîte](#), Rc ou Arc

break expressions , [flux de contrôle dans les boucles](#)

canal de diffusion , [Groupes de chat : canaux de diffusion de tokio](#) - [Groupes de chat : canaux de diffusion de tokio](#)

Type de collection BTreeMap<K, V> , [HashMap<K, V> et BTreeMap<K, V>](#) - [Itération de carte](#)

Type de collection BTreeSet , [HashSet<T> et BTreeSet<T>](#) - [Opérations sur l'ensemble](#)

BTreeSet::new , [HashSet<T> et BTreeSet<T>](#)

lecteurs tamponnés , [Lecteurs tamponnés](#) - [Lignes de collecte](#)

Caractéristique BufRead , [lecteurs tamponnés](#)

méthode de consommation , [lecteurs tamponnés](#)

méthode fill_buf , [lecteurs tamponnés](#)

méthode read_line , [lecteurs tamponnés](#)

Méthode read_until , [Lecteurs mis en mémoire tampon](#)

méthode fractionnée , [lecteurs tamponnés](#)

Type BufReader , [Lecteurs tamponnés](#)

BufReader<TcpStream> , [Réception de paquets : Plus de flux asynchrones](#)

BufWriter::with_capacity , [Écrivains](#)

bogues, code non sécurisé et fonctions [non sécurisées](#)

créer des profils , [Créer des profils](#)

build script , [Utilisation des fonctions des bibliothèques](#)

Trait BuildHasher , [Utilisation d'un algorithme de hachage personnalisé](#)

par valeur/par référence

passage d'une collection , [Implémentations IntoIterator](#) ,
[Collections](#)

passage des arguments de la fonction , [Références aux valeurs](#) - [Références aux valeurs](#)

Littéraux d'octets , [Types entiers](#)

chaînes d'octets , chaînes d' [octets](#)

byteorder crate , [données binaires, compression et sérialisation](#)

adaptateur itérateur by_ref , [by_ref](#) - [by_ref](#)

C

C , les [programmeurs système peuvent avoir de belles choses](#)

(voir aussi fonctions étrangères)

énumérations , [énumérations](#) - [énumérations](#)

passage de chaînes entre Rust et , [Recherche de représentations de données communes](#)

pointeurs dans , [Pointeurs bruts](#) , Pointeurs [bruts](#)

règles pour éviter les comportements indéfinis , les [programmeurs système peuvent avoir de belles choses](#)

représentations de type , [Recherche de représentations de données communes](#) - [Recherche de représentations de données communes](#)

C#

fonctions asynchrones , [Comparaison de conceptions asynchrones](#)

Enums , [Enums et Patterns](#) - [Enums](#)

traits versus méthodes virtuelles , [Utilisation des traits](#)

C++ , [les programmeurs système peuvent avoir de belles choses](#)

(voir aussi fonctions étrangères)

affectation dans , [Moves](#) - [Moves](#)

constexpr , [Variables globales](#)

Enums , [Enums et Patterns](#) - [Enums](#)

erreurs d'invalidation et , [Partage contre mutation](#) , [Collections](#) ,
[Rust exclut les erreurs d'invalidation](#)

macros , [Macros](#) , [Conséquences inattendues](#)

mutex dans , [Qu'est-ce qu'un mutex ?](#) - [Qu'est-ce qu'un mutex ?](#)

propriété dans , [Propriété](#) - [Propriété](#)

pointeurs dans , [Pointeurs bruts](#) , Pointeurs [bruts](#)

pointeurs vers const versus références partagées , [partage versus mutation](#)

création de références dans , [Références Rust Versus Références C++](#)

règles pour éviter les comportements indéfinis , les [programmeurs système peuvent avoir de belles choses](#)

traits versus méthodes virtuelles , [Utilisation des traits](#)

méthode calculate_tides , [Gestion des erreurs dans main\(\)](#)

rappels, fermetures et , [Rappels - Rappels](#)

annulation, atomique et , [Atomique](#)

capacité d'un vecteur , [Vecteurs](#) , Vecteurs [croissants et rétrécissants](#)

méthode captures_iter sur le type Regex , [utilisation de base de Regex](#)

Cargo , [Rust facilite la collaboration](#)

build script , [Utilisation des fonctions des bibliothèques](#)

documentation , [Documentation](#) - [Doc-Tests](#)

rustup et , [rustup et Cargo](#) - [rustup et Cargo](#)

Répertoire src/bin , [Le répertoire src/bin](#) - [Le répertoire src/bin](#)

gestion des versions , [spécification des dépendances](#)

construction de cargaison , [Caisses](#) - [Caisses](#)

Commande de fret , [Rustup et Cargo](#)

Commande cargo doc , [Documentation](#)

commande de package de fret , [publication de caisses sur crates.io](#)

commande de test cargo , [Tests et Documentation](#) - [Tests et Documentation](#)

Cargo.lock , [Cargo.lock](#)

conversion de casse

pour les caractères , [Conversion de casse pour les caractères](#)

pour les chaînes , [Conversion de casse pour les chaînes](#)

moulages , moulages de [type](#)

Fonction catch_unwind , [Déroulement](#)

Type cellulaire , [Mutabilité intérieure](#)

#[cfg] attribut , [Attributs](#) , [Fonctionnalités spécifiques à la plate-forme](#)

cfg ! macro , [Macros intégrées](#)

Méthode ch.to_digit , [Gestion des chiffres](#)

adaptateur de chaîne , [chaîne](#)

canaux , [Canaux](#) - Au- [delà des pipelines](#)

canal de diffusion , [Groupes de chat : canaux de diffusion de tokio](#) - [Groupes de chat : canaux de diffusion de tokio](#)

impasse avec , [impasse](#)

Fonctionnalités et performances , [Fonctionnalités et performances de la chaîne](#) - Fonctionnalités et performances de la chaîne canaux multiconsommateurs utilisant mutex , [Canaux multiconsommateurs utilisant mutex](#)

utilisations hors pipeline , Au- [delà des pipelines](#)

rediriger l'itérateur vers , [Diriger presque n'importe quel itérateur vers un canal](#) - [Diriger presque n'importe quel itérateur vers un canal](#)

recevoir des valeurs , [recevoir des valeurs](#)

Envoyer et synchroniser pour la sécurité des threads , [Sécurité des threads : envoyer et synchroniser](#) - [Sécurité des threads : envoyer et synchroniser](#)

envoyer des valeurs , [Canaux](#) - Au- [delà des pipelines](#)

caractères littéraux , [Caractères](#)

caractères (char) , [Caractères](#) - [Caractères](#) , [Caractères \(char\)](#) - [Conversions vers et depuis des entiers](#)

conversion de casse , [Conversion de casse pour les caractères](#)

classer , [Classer les caractères](#) - [Classer les caractères](#)

chiffres , [Manipulation des chiffres](#)

conversion d'entiers , [Conversions vers et depuis des entiers](#)

Méthode is_digit , [Gestion des chiffres](#)

Méthode is_lowercase , [Conversion de casse pour les caractères](#)

Méthode is_uppercase , [Conversion de casse pour les caractères](#)

types numériques versus , [Caractères](#)

Points communs de type Rust et C , [Trouver des représentations de données communes](#)

Méthode to_digit , [Gestion des chiffres](#)

Méthode to_lowercase , [Conversion de casse pour les caractères](#)

Méthode to_uppercase , [Conversion de casse pour les caractères](#)

groupes de discussion, chaînes de diffusion de tokio , [Groupes de discussion : chaînes de diffusion de tokio](#) - [Groupes de discussion : chaînes de diffusion de tokio](#)

opérations vérifiées , [Arithmétique vérifiée, enveloppante, saturée et débordante](#)

processus enfant , [Autres types de lecteurs et d'enregistreurs](#)

Type ChildStdin , [Autres types de lecteurs et d'enregistreurs](#)

client et serveur, chat asynchrone , [Un client et un serveur asynchrones](#) - [Groupes de discussion : canaux de diffusion de tokio](#)

connexions de chat avec des mutex asynchrones , [Gestion des connexions de chat](#) : mutex asynchrones - [Gestion des connexions de chat : mutex asynchrones](#)

groupes de discussion, chaînes de diffusion de tokio , [Groupes de discussion : chaînes de diffusion de tokio](#) - [Groupes de discussion : chaînes de diffusion de tokio](#)

la fonction principale du client , La fonction principale du [client](#) - [La fonction principale du client](#)

types d' erreur et de résultat , [Types d'erreur et de résultat](#)

protocole , [Le protocole](#) - [Le protocole](#)

réception de paquets , [Réception de paquets : plus de flux asynchrones](#) - [Réception de paquets : plus de flux asynchrones](#)

envoi de paquets , [envoi de paquets](#)

fonction principale du serveur , La fonction principale du [serveur](#)

flux pour prendre l'entrée de l'utilisateur , [Prise de l'entrée de l'utilisateur : flux asynchrones](#) - [Prise de l'entrée de l'utilisateur : flux asynchrones](#)

méthode de clonage , [Mouvements](#) , [Utilisation de caractéristiques](#)

Clone trait , [Clone](#) , [Copy et Clone for Closures](#) , [Accessing Elements](#)

méthode de l'adaptateur cloné pour les itérateurs , [cloné, copié](#)

plages fermées (inclusives) , [champs et éléments](#)

Fermetures , [Un programme Mandelbrot simultanément](#) , [Fermetures](#) , [Fermetures](#) - [Utilisation efficace des fermetures](#)

références d'emprunt , [Fermetures qui empruntent](#)

rappels , [Rappels](#) - [Rappels](#)

capture de variables , [Capture de variables](#)

Cloner pour , [copier et cloner pour les fermetures](#)

Copier pour , [copier et cloner pour les fermetures](#)

laisser tomber des valeurs , des [fermetures qui tuent](#) - [FnOnce](#)

utilisation efficace de , [Utilisation efficace des fermetures](#)

FnMut , [FnMut](#) - [FnMut](#)

FnOnce , [FnOnce](#) - [FnOnce](#)

inspecter l'adaptateur et , [inspecter](#)

mise en page en mémoire , [performance de fermeture](#)

Mot- clé de déplacement , [Des fermetures qui volent](#)

performances , performances de [fermeture](#)

sécurité , [fermetures et sécurité](#) - [Copier et cloner pour les fermetures](#)

"qui tue" , [Des fermetures qui tuent](#)

types , types de [fonction et de fermeture](#) - Types [de fonction et de fermeture](#)

dans l'exemple de serveur Web , [Servir des pages sur le Web](#)
fragments de code, macros , [Principes de base de l'expansion des macros](#) - [Conséquences inattendues](#)
collaboration, Rust et , [Rust facilite la collaboration](#)
méthode collect , [Un programme Mandelbrot concurrent](#) , [L'interface de ligne de commande](#) , [Construire des collections : collect et FromIterator](#) - [Construire des collections : collect et FromIterator](#) , [partition](#) , [Collecter des lignes](#)
collections , [Collections](#) - Au- [delà des collections standard](#)
BinaryHeap<T>type de collection , [BinaryHeap<T>](#) - [BinaryHeap<T>](#)
BTreeMap<K, V> , [HashMap<K, V>](#) et [BTreeMap<K, V>](#) - [Itération de carte](#)
BTreeSet<T> , [HashSet<T>](#) et [BTreeSet<T>](#) - [Opérations sur l'ensemble](#)
sur mesure , [Au-delà des collections standard](#)
hachage , [Hachage](#) - [Utilisation d'un algorithme de hachage personnalisé](#)
HashMap<K, V> , [HashMap<K, V>](#) et [BTreeMap<K, V>](#) - [Itération de carte](#)
HashSet<T> , [HashSet<T>](#) et [BTreeSet<T>](#) - [Opérations sur l'ensemble](#)
itération par valeur , [Implémentations IntoIterator](#) , [Collections](#)
itérateurs et , [iter et iter mut](#) Méthodes
chaînes en tant que génériques , [chaînes en tant que collections génériques](#)
Vec<T> , [Vec<T>](#) - [Rust élimine les erreurs d'invalidation](#)
VecDeque<T> , [VecDeque<T>](#) - [VecDeque<T>](#)
colonne! macro , [Macros intégrées](#)
invite de commande (\$) , [rustup et Cargo](#)
Type de commande , [Autres types de lecteurs et d'enregistreurs](#)
arguments de ligne de commande , [Gestion des arguments de ligne de commande](#) - [Gestion des arguments de ligne de commande](#)
interface de ligne de commande , [Systèmes de fichiers et outils de ligne de commande](#) - [L'interface de ligne de commande](#)
communauté, Rust , [Plus de belles choses](#)

opérateurs de comparaison , [Type booléen](#) , [Opérateurs arithmétiques, binaires, de comparaison et logiques](#)

avec des itérateurs , [Comparer des séquences d'éléments surcharge](#) , [comparaisons ordonnées](#)

références et , [Comparaison des références](#)

avec des chaînes , [Utilisation de chaînes](#)

équivalence de compatibilité pour les caractères Unicode , [Formes de normalisation](#)

nombres complexes , [Analyse des arguments de la ligne de commande d'une paire](#)

caractères Unicode composés ou décomposés , [Normalisation](#)

opérateurs d'affectation composés , [Affectation](#) , [Opérateurs](#) d'affectation composés - Opérateurs d' [affectation composés](#)

compression , [données binaires, compression et sérialisation](#)

méthode concat , [chaîne](#)

concat ! macro , [Macros intégrées](#)

simultanéité , [La programmation parallèle est apprivoisée](#) , [Concurrence - À quoi ressemble le piratage de code simultané dans Rust](#)

canaux , [Canaux](#) - Au- [delà des pipelines](#)

parallélisme fork-join , [Parallélisme fork-join](#) - [Revisiter l'ensemble de Mandelbrot](#)

Ensemble de Mandelbrot , [Un programme de Mandelbrot concurrent](#) - [Un programme de Mandelbrot concurrent](#)

Prise en charge de Rust pour , [Concurrency](#) - [Safety Is Invisible](#) , [Ownership and Moves](#)

état mutable partagé , État [mutable partagé](#) - [Variables globales](#)

condition (avec instruction if) , [if et match](#)

Variables de condition (Condvar) , [Variables de condition \(Condvar\)](#)

fonction const , [variables globales](#)

const generics , [structures génériques avec paramètres constants](#) , [fonctions génériques et paramètres de type](#)

Constantes , [Modules](#) , [Statique et constantes](#) , Variables [globales](#)

consts , [statiques et constantes](#)

*const T , [pointeurs bruts](#)

associés , [Const](#) associés , Const [associés](#)

références partagées versus pointeurs vers , [partage versus mutation](#)

méthode de consommation , [lecteurs tamponnés](#)

consommer des itérateurs , [Consommer des itérateurs - for_each et try_for_each](#)

méthodes d'accumulation , [Accumulation simple : comptage, somme, produit](#)

toutes les méthodes , [toutes et toutes](#)

méthode collect , [Création de collections : collect et FromIterator - Création de collections : collect et FromIterator](#)

comparaison de séquences d'articles , [Comparaison de séquences d'articles](#)

méthode de comptage , [Accumulation simple : comptage, somme, produit](#)

ExactSizeIterator , [position, rposition et ExactSizeIterator](#)

Trait d'extension , [Le trait d'extension](#)

Méthodes find, rfind et find_map , [find, rfind et find_map](#)

méthode de pliage , [plier et replier](#)

méthode for_each , [for_each et try_for_each](#)

Caractéristique FromIterator , Création [de collections : collect et FromIterator](#)

dernière méthode , [dernière](#)

méthodes max_by et min_by , [max_by, min_by](#)

Méthodes max_by_key et min_by_key , [max_by_key, min_by_key](#)

méthodes min et max , [max, min](#)

Méthodes nth et nth_back , [nth, nth_back](#)

méthode de partition , [partition](#)

position method , [position, rposition et ExactSizeIterator](#)

méthode du produit , [Accumulation simple : compte, somme, produit](#)

méthode rfind , [find, rfind et find_map](#)

méthode rfold , [fold et rfold](#)

méthode rposition , [position, rposition et ExactSizeIterator](#)

méthode somme , [Accumulation simple : compte, somme, produit](#)

méthodes try_fold et try_rfold , [try_fold et try_rfold - try_fold et try_rfold](#)

méthode try_for_each , [for_each et try_for_each](#)

contrats

fonctionnalité non sécurisée et , [Dangereux de quoi ?](#)

fonctions non sécurisées et , [Fonctions non sécurisées](#)

traits dangereux et , [traits dangereux](#)

adaptateur copié , [cloné, copié](#)

méthode de copie , [lecteurs et écrivains](#)

Type de copie , [types de copie : l'exception aux mouvements](#) - [Types de copie : l'exception aux mouvements](#) , [copie](#) , [copie et clonage pour les fermetures](#)

méthode de comptage , [Accumulation simple : comptage, somme, produit](#)

Type de vache (clone en écriture) , [emprunter et posséder au travail : la vache humble](#) , reporter l' [allocation](#) - [reporter l'allocation](#)

caisses , [Caisses](#) - [Construire des profils](#)

Attribut #[inline] , [Attributs](#)

\$crate fragment versus crate keyword , [Importation et exportation de macros](#)

doc-tests , [Doc-Tests](#) - [Doc-Tests](#)

publication sur crates.io , [Publication de caisses sur crates.io](#)

spécification des dépendances , [spécification des dépendances](#) - [Cargo.lock](#)

Répertoire src/bin et , [Le répertoire src/bin](#) - [Le répertoire src/bin](#)
espaces de travail , Espaces de [travail](#)

crates.io , [Publier des caisses sur crates.io](#)

section critique du code , [Qu'est-ce qu'un mutex ?](#)

caisse à traverses , [un programme Mandelbrot simultané](#)

Cursor::new , [Autres types de lecteurs et d'enregistreurs](#)

adaptateur vélo , [vélo](#)

ré

pointeur suspendu , [Propriété et mouvements](#) , [Partage contre mutation](#)

parallélisme des données , [Concurrence](#)

courses aux données , [Concurrence](#) , La [sécurité est invisible](#) , [Qu'est-ce qu'un mutex ?](#) , [Pourquoi les mutex ne sont pas toujours une bonne idée](#)

impasse , [impasse](#)

Trait de formatage de débogage , [Conversion d'autres types en chaînes](#) , [Pointeurs bruts](#)

débogage

valeurs de formatage pour , [Valeurs de formatage pour le débogage](#)
- [Pointeurs de formatage pour le débogage](#)

macros , [Macros](#) de débogage - [Macros de débogage](#)

debug_assert ! macro , [Fonctions Rust](#) , [Tests et Documentation](#)

debug_assert_eq ! macro , [Tests et Documentation](#)
déclarations , [Déclarations](#) , [Déclarer des fonctions étrangères et des variables](#) - [Utilisation des fonctions des bibliothèques](#)
caractères Unicode décomposés et composés , [Normalisation](#)
Trait par défaut , [Par défaut](#) - [Par défaut](#) , [partition](#)
implémentation des traits par défaut , [méthodes par défaut](#)
dépendances
Cargo.lock , [Cargo.lock](#)
dans le contexte de la caisse , [Caisses](#)
spécifiant , [Spécification des dépendances](#) - [Cargo.lock](#)
versions et , [Spécification des dépendances](#)
graphe de dépendance , [Crates](#)
Deref coercions , [Type Casts](#) , [Deref et DerefMut](#) , [Raw Pointers](#)
Deref trait , [Deref et DerefMut](#) - [Deref et DerefMut](#)
déréférencement
* opérateur , [Gestion des arguments de ligne de commande](#) , [Références Rust versus références C++](#) , [Pointeurs bruts](#)
pointeurs bruts , pointeurs [bruts](#) , [dangereux de quoi ?](#) , [Blocs non sécurisés](#) , [Déréférencement des pointeurs bruts en toute sécurité](#) - [Exemple : RefWithFlag](#)
DerefMut trait , [Deref et DerefMut](#) - [Deref et DerefMut](#)
Attribut #[derive] , [diffusion de pages sur le Web](#)
Trait de désérialisation , [Réception de paquets : plus de flux asynchrones](#)
chiffres, manipulation , [Manipulation des chiffres](#)
directionnalité du texte , [Directionnalité du texte](#)
répertoires
modules et , [Modules dans des fichiers séparés](#)
lecture , [Répertoires](#) de lecture - [Répertoires de lecture](#)
src/bin , [Le répertoire src/bin](#) - [Le répertoire src/bin](#)
Structure DirEntry , [Lecture des répertoires](#)
méthode nom_fichier , [Lecture des répertoires](#)
méthode file_type , [Lecture des répertoires](#)
méthode des métadonnées , [lecture des répertoires](#)
méthode path , [Lecture des répertoires](#)
unions discriminées , [énumérations et modèles](#)
Trait de formatage d'affichage , [Conversion d'autres types en chaînes](#) , [Pointeurs bruts](#)
fonction divergente , [pourquoi Rust a une boucle](#)

commentaires doc , [Documentation](#)
doc-tests , [Doc-Tests](#) - [Doc-Tests](#)
documentation , [Documentation](#) - [Doc-Tests](#)
commentaires de documentation (///) , [Ce qu'est réellement l'ensemble de Mandelbrot](#) , [Documentation](#)
guillemets doubles , [servir des pages sur le Web](#)
Trait DoubleEndedIterator , [itérateurs réversibles et rev](#)
méthode de vidange , [Méthodes de vidange](#)
Trait de chute , [Drop](#) - [Drop](#) , [Lecteurs](#)
suppression de valeurs
 dans les fermetures , [Les fermetures qui tuent](#) - [FnOnce](#)
 FnOnce , [FnOnce](#) - [FnOnce](#)
 propriété et , [Propriété](#)
 dans Rust , [Propriété](#)
typage canard , [Types fondamentaux](#)
largeurs et précisions dynamiques , largeurs et précisions [dynamiques](#)

E

éditions , [éditions](#)
méthode écoulée , [Traiter les erreurs qui "ne peuvent pas se produire"](#)
éléments
 valeurs de structure de type tuple , structures de type [tuple](#)
 Type de collection Vec<T> , [Accès aux éléments](#) - [Accès](#) aux éléments , [Éléments aléatoires](#)
algorithme parallèle embarrassant , [Concurrence](#)
plages exclusives (semi-ouvertes) , [Champs et éléments](#) , [Possibilités multiples correspondantes](#)
plages inclusives (fermées) , [champs et éléments](#)
entrées, mappez les paires clé-valeur comme , [HashMap<K, V>](#) et [BTreeMap<K, V>](#)
Type d'entrée, HashMap et BTreeMap , [Entrées](#) - [Entrées](#)
adaptateur d'énumération , [un programme Mandelbrot concurrent](#) , [énumération](#) , [zip](#)
type énuméré (enum) , [Qu'est-ce que l'ensemble de Mandelbrot est réellement](#) , [Énumérations et modèles](#) - [Vue d'ensemble](#)
 Style C , [Enums](#) - [Enums](#)
 avec données , [Énumérations avec données](#)
 générique , [Énumérations](#) génériques - [Énumérations génériques](#)
 implémentation de hachage , [Hachage](#)

en mémoire , [Énumérations en mémoire](#)
structures de données riches avec , [Structures de données riches utilisant des énumérations](#) - [Structures de données riches utilisant des énumérations](#)
module env , [gestion des arguments de la ligne de commande](#)
env! macro , [Macros intégrées](#)
eprintln ! macro , [Gestion des arguments de ligne de commande](#)
Trait d'équation , [Hachage](#)
opérateurs d'égalité , Opérateurs [arithmétiques, binaires, de comparaison et logiques](#) , [Comparaisons](#) d'équivalence - [Comparaisons d'équivalence](#)
gestion des erreurs , [Gestion](#) des erreurs - [Pourquoi des résultats ?](#)
sur les threads , [Gestion des erreurs sur les threads](#)
anyhow crate , [Travailler avec plusieurs types d'erreurs](#)
chat asynchrone , [types d'erreur et de résultat](#)
éviter les erreurs de syntaxe dans la correspondance des macros , [Éviter les erreurs de syntaxe lors de la correspondance](#)
attraper les erreurs , [Attraper les erreurs](#) - [Attraper les erreurs](#)
canaux et , [Valeurs d'envoi](#)
déclaration d'un type d'erreur personnalisé , [Déclaration d'un type d'erreur personnalisé](#)
erreurs qui "ne peuvent pas arriver" , [Traiter les erreurs qui "ne peuvent pas arriver"](#)
types d'erreurs de formatage , [Autres types de formatage](#)
ignorer les erreurs , [Ignorer les erreurs](#)
erreurs d'invalidation , [Collections](#) , [Rust exclut les erreurs d'invalidation](#)
dans la fonction main , [Gestion des erreurs dans main\(\)](#)
avec plusieurs types d'erreurs , [Utilisation de plusieurs types d'erreurs](#) - [Utilisation de plusieurs types d'erreurs](#)
panique , [Panique](#) - [Abandon](#)
PoisonError::into_inner , [mutex empoisonnés](#)
erreurs d'impression , [Erreurs](#) d'impression - [Erreurs d'impression](#)
propagation des erreurs , [propagation des erreurs](#)
Type de résultat , [Résultat](#) - [Pourquoi des résultats ?](#) , [Types d'erreur et de résultat](#)
code non sécurisé et , [Une interface sécurisée pour libgit2](#)
Caractéristique d'erreur
méthode source , [Erreurs d'impression](#)

méthode to_string , [Erreurs d'impression](#)
fonction escape_time , [from fn et successeurs](#)
ExactSizeIterator trait , [position, rposition et ExactSizeIterator](#)
exceptions, Résultat versus , [Pourquoi des résultats ?](#)
plages exclusives (semi-ouvertes) , [champs et éléments](#)
exécuteurs (asynchrones)
block_on , [Appel de fonctions asynchrones à partir de code synchrone : block on](#) - [Appel de fonctions asynchrones à partir de code synchrone : block on](#) , [Futurs primitifs et exécuteurs : quand un futur mérite-t-il d'être interrogé à nouveau ?](#) , [Implémenter block on](#) - [Implémenter block on](#)
fonction spawn , [spawn and join](#) - [spawn and join](#) , [Error Handling Across Threads](#) , [Spawning Async Tasks](#) , [Spawning Async Tasks on a Thread Pool](#)
spawn_local , Génération de [tâches](#) asynchrones - Génération de tâches asynchrones , Génération de [tâches asynchrones sur un pool de threads](#)
méthode expect , [Gestion des arguments de ligne de commande](#) , [Écriture de fichiers image](#) , [Gestion des erreurs dans main\(\)](#)
expressions , [Expressions](#) - [En avant](#)
affectation , [affectation](#)
blocs et points-virgules , [Blocs et points-virgules](#) - [Déclarations](#)
Fermetures , [Fermetures](#)
déclarations , [Déclarations](#)
champs et éléments , [Champs et éléments](#)
appels de fonction/méthode , Appels de [fonction et de méthode](#)
si et match , [si et match](#) - [si let](#)
si laissé , [si laissé](#)
loops , [Flux de contrôle dans les boucles](#) - [Flux de contrôle dans les boucles](#)
Priorité et associativité , [Priorité et associativité](#)
opérateurs de référence , [Opérateurs de référence](#)
expressions régulières , [Expressions régulières](#) - [Création de valeurs régulières paresseusement](#)
retour , [retour Expressions](#)
Rust comme langage d'expression , [Un langage d'expression](#)
déclarations versus , [un langage d'expression](#)
struct , [Structures de champ nommé](#)
moulages de type , moulages de [type](#)

fonction d'extension , [partage contre mutation](#)
Trait d'extension , [Le trait d'extension](#)
méthode extend_from_slice , [partage contre mutation](#)
Traits d' extension , [Traits et autres types de personnes](#) , [Prise d'entrée utilisateur : flux asynchrones](#)
extern block , [Déclarer des fonctions étrangères et des variables](#) - [Déclarer des fonctions étrangères et des variables](#)

F

pointeur gras , [tranches](#) , [références aux tranches et aux objets caractéristiques](#) , [pointeurs bruts](#)
Attribut #![feature] , [Attributs](#)
FFI (voir fonctions étrangères)
champs, expressions et , [champs et éléments](#)
Type de fichier , [Utilisation du langage de formatage dans votre propre code](#) , [Recherche](#)
dossier! macro , [Macros intégrées](#)
Fichier :: créer , [Fichiers](#)
Fichier :: ouvrir , [Fichiers](#)
types de noms de fichiers , [OsStr et Path](#)
fichiers , [Fichiers et Répertoires](#) - [Fonctionnalités spécifiques à la plate-forme](#)
fonctions d'accès au système de fichiers , Fonctions d'accès au système de [fichiers](#) - [Fonctions d'accès au système de fichiers](#)
OsStr et Chemin , [OsStr et Chemin](#) - [OsStr et Chemin](#)
Types Path et PathBuf , [Méthodes](#) Path et PathBuf - Méthodes [Path et PathBuf](#)
fonctionnalités spécifiques à la plate-forme , fonctionnalités spécifiques à la plate-forme - [fonctionnalités spécifiques à la plate-forme](#)
lire et écrire , [Lire et écrire des fichiers](#) , [Fichiers](#)
répertoires de lecture , [Répertoires](#) de lecture - [Répertoires de lecture](#)
systèmes de fichiers , [Systèmes de fichiers et outils de ligne de commande](#) - [Rechercher et remplacer](#) , [Fonctions d'accès au système de fichiers](#)
méthode fill_buf , [lecteurs tamponnés](#)
adaptateur de filtre , [carte et filtre](#) - [carte et filtre](#)
adaptateur filter_map , [filter_map et flat_map](#) - [filter_map et flat_map](#)
rechercher et remplacer , [rechercher et remplacer](#)

find method , [Parsing Pair Command-Line Arguments](#) , [find, rfind et find_map](#)

find_iter iterator , [Utilisation de base de Regex](#)

méthode find_map , [find, rfind et find_map](#)

types numériques à largeur [fixe](#) , Types numériques à largeur fixe -

Types à virgule [flottante](#)

flate2 crate , [données binaires, compression et sérialisation](#)

aplatir l'adaptateur , [aplatir](#) - [aplatir](#)

adaptateur flat_map , [filter_map et flat_map](#) - [filter_map et flat_map](#)

littéraux à virgule flottante , [Types à virgule flottante](#)

types à virgule flottante , Types à virgule [flottante](#) - [Types](#) à virgule

flottante , [max, min](#) , [Formatage des nombres](#)

analyses sensibles au flux , [boucle Why Rust Has](#)

méthode flush , [Méthodes par défaut](#)

Architecture de flux , [Utilisation efficace des fermetures](#)

module fmt , [Formatage de vos propres types](#)

fn mot-clé , [Fonctions Rust](#) , [Déclarations](#)

Trait Fn , [FnMut](#)

type fn , [Rappels](#)

Trait FnMut , [FnMut](#) - [FnMut](#) , [from fn et successeurs](#)

Trait FnOnce , [FnOnce](#) - [FnOnce](#)

fnv crate , [Utilisation d'un algorithme de hachage personnalisé](#)

méthode fold , [itérateurs](#) , [fold et rfold](#)

boucle for , [gestion des arguments de la ligne de commande](#)

contrôler le flux dans , [contrôler le flux dans les boucles](#)

IntoIterator , [L'itérateur et les traits](#) IntoIterator , [Implémentations](#)

IntoIterator , Implémentations [IntoIterator](#)

fonctions étrangères , [Fonctions étrangères](#) - [Une interface sécurisée pour libgit2](#)

déclaration de fonctions et de variables étrangères , [Déclaration de fonctions et de variables étrangères](#) étrangères - [Déclaration de fonctions et de variables étrangères](#)

recherche de représentations de données communes , [Recherche de représentations de données communes](#) - [Recherche de représentations de données communes](#)

à partir des bibliothèques , [Utilisation des fonctions des bibliothèques](#) - [Utilisation des fonctions des bibliothèques](#)

interface brute vers libgit2 , [Une interface brute vers libgit2](#) - [Une interface brute vers libgit2](#)

interface sécurisée vers libgit2 , [Une interface sécurisée vers libgit2](#)
- [Une interface sécurisée vers libgit2](#)
code non sécurisé et , [Blocs non sécurisés](#)
parallélisme fork-join , [Parallélisme fork-join](#) - [Revisiter l'ensemble de Mandelbrot](#)
gestion des erreurs sur les threads , [Gestion des erreurs sur les threads](#)
Rendu de l'ensemble de Mandelbrot , [Revisiter l'ensemble de Mandelbrot](#) - [Revisiter l'ensemble de Mandelbrot](#)
Bibliothèque Rayon , [Rayon](#) - [Rayon](#)
partagé des données immuables entre les threads , [Partage de données immuables entre les threads](#) - [Partage de données immuables entre les threads](#)
engendrer et rejoindre , [engendrer et rejoindre](#) - [engendrer et rejoindre](#)
paramètres de format , [Valeurs de formatage](#)
format! macro , [chaîne](#) , [valeurs de formatage](#)
formatage des arguments, par index ou nom , [référence aux arguments par index ou nom](#)
formatage des nombres , [Formatage des nombres](#) - [Formatage des nombres](#)
valeurs de formatage , [Valeurs](#) de formatage - [Utilisation du langage de formatage dans votre propre code](#)
Valeurs booléennes , [Mise en forme d'autres types](#)
pour le débogage , [Formatage des valeurs pour le débogage](#) - [Formatage des pointeurs pour le débogage](#)
Trait d'affichage , [Conversion d'autres types en chaînes](#) , [Pointeurs bruts](#)
largeurs et précisions dynamiques , largeurs et précisions [dynamiques](#)
types d'erreurs , [Formatage d'autres types](#)
notation de directive de format de chaîne , [formatage de vos propres types](#)
langage de formatage dans votre propre code , [Utilisation du langage de formatage dans votre propre code](#) - [Utilisation du langage de formatage dans votre propre code](#)
implémenter des traits pour vos propres types , [Formatting Your Own Types](#) - [Formatting Your Own Types](#)
types d'adresses de protocole Internet , [Formatage d'autres types](#)

Trait de pointeur , [Pointeurs bruts](#)

se référant aux arguments par index ou nom , [se référant aux arguments par index ou nom](#)

exemples de chaînes , [Formatage des valeurs](#)

valeurs de texte , [Formatage des valeurs de texte](#) - [Formatage des valeurs de texte](#)

format_args ! macro , [Formatage des valeurs](#) , [Utilisation du langage de formatage dans votre propre code](#)

méthode for_each , [for_each](#) et [try_for_each](#)

fonctions libres , [Définir des méthodes avec impl](#)

From trait , [From and Into](#) - [From and Into](#) , [Utilisation de traits avec des macros](#)

Trait FromIterator , Création [de collections : collect](#) et [FromIterator](#) , [Le trait Extend](#) , [Création de valeurs de chaîne](#)

FromStr trait , [Analyse d'autres types à partir de chaînes](#)

Méthode from_digit , [Gestion des chiffres](#)

méthode from_fn , [from_fn](#) et [successeurs](#) - [from_fn](#) et [successeurs](#)

fonction from_slice , [Fonctions associées au type](#)

Méthode from_str , [Gestion des arguments de la ligne de commande](#)

module fs , [Fonctions d'accès au système de fichiers](#)

Appels de méthode entièrement qualifiés , Appels de méthode entièrement qualifiés - [Appels de méthode entièrement qualifiés](#)

arguments de fonction, réception de références en tant que , [Réception de références en tant qu'arguments de fonction](#) - [Réception de références en tant qu'arguments de fonction](#)

pointeurs de fonction (type fn) , [Callbacks](#)

langage fonctionnel , [Vecteurs](#)

les fonctions

associés , [Définir des méthodes avec impl](#)

async , [Fonctions asynchrones et expressions Await](#) - [Appel de fonctions asynchrones à partir de code synchrone : block_on](#) , Création

de fonctions asynchrones [à partir de blocs asynchrones](#)

appelant , [Appels de fonction et de méthode](#)

const , [variables globales](#)

accès au système de fichiers , Fonctions d'accès au système de [fichiers](#) - [Fonctions d'accès au système de fichiers](#)

étranger (voir fonctions étrangères)

gratuit , [Définir des méthodes avec impl](#)

générique , [Arguments de ligne de commande de paire d'analyse](#) ,
[Types fondamentaux](#) , [Fonctions génériques et paramètres de type](#) -
[Fonctions génériques et paramètres de type](#)
passage de références à , [passage de références à des fonctions](#)
syntaxe pour , [Fonctions Rust](#) - [Fonctions Rust](#)
associées au type , Fonctions associées au type , [Fonctions associées
au type](#)
types , types de [fonction et de fermeture](#) - Types [de fonction et de
fermeture](#)
non sécurisé , [Code](#) non sécurisé , [Fonctions](#) non sécurisées - Fonc-
tions [non sécurisées](#)
adaptateur de fusible , [fusible](#) , [Futures](#)
Trait futur , [Futures](#) - [Futures](#)
Trait FutureExt , [la fonction principale du client](#)
futurs , [De synchrone à asynchrone](#) - [Un vrai client HTTP asynchrone](#) ,
[Futurs primitifs et exécuteurs : quand un futur mérite-t-il d'être à nou-
veau interrogé ?](#) - [Implémentation de block on](#)
blocs asynchrones , [Blocs asynchrones](#) - [Création de fonctions asyn-
chrones à partir de blocs asynchrones](#)
fonctions asynchrones , fonctions asynchrones [et expressions d'at-
tente](#) - [Appel de fonctions asynchrones à partir de code synchrone :](#)
[block on](#) , [création de fonctions asynchrones à partir de blocs
asynchrones](#)
caisse de client HTTP asynchrone , [un vrai client HTTP asynchrone](#)
attendre l'expression , [les fonctions asynchrones et les expressions
d'attente](#)
block_on , [Appel de fonctions asynchrones à partir de code syn-
chrone : block on](#) - [Appel de fonctions asynchrones à partir de code
synchrone : block on](#) , [Implémentation de block on](#) - [Implémenta-
tion de block on](#)
emprunt et , génération de [tâches asynchrones](#)
comparaison de conceptions asynchrones , [Comparaison de
conceptions asynchrones](#)
implémente Send , [mais votre futur implémente-t-il Send ?](#) - [Mais
votre futur outil envoie-t-il ?](#)
calculs de longue durée , calculs de [longue durée : yield now et
spawn blocking](#) - calculs de [longue durée : yield now et
spawn blocking](#)
épinglage , [Épinglage](#) - [Le trait de désépinglage](#)

généralisant des tâches asynchrones , [Généralisant](#) des tâches asynchrones - [Généralisant](#) des tâches asynchrones , Généralisant des tâches asynchrones [sur un pool de threads](#)
spawn_blocking , [Appel des Wakers : spawn_blocking](#) - [Appel des Wakers : spawn_blocking](#)

og

GapBuffer , [Exemple : GapBuffer](#) - [Sécurité anti-panique dans le code non sécurisé](#)
garbage collection , [Types de pointeurs](#) , [Propriété et déplacements](#) , [Capture de variables](#)
fonction gcd , [Fonctions Rust](#) , [Servir des pages sur le Web](#)
code générique , [Traits et génériques](#) , [Traits et génériques](#)
types associés et , [Types associés \(ou Fonctionnement des itérateurs\)](#) - [Types associés \(ou Fonctionnement des itérateurs\)](#)
Const , [Const associé](#)
fonctions génériques , [Analyse d'arguments de ligne de commande de paires](#) , [Types fondamentaux](#) , [Fonctions génériques et paramètres de type](#) - [Fonctions génériques et paramètres de type](#)
traits génériques , traits [génériques \(ou comment fonctionne la surcharge d'opérateur\)](#) - [impl Trait](#)
Implémentations IntoIterator et , [IntoIterator](#)
limites de la rétro-ingénierie , [Limites de la](#) rétro-ingénierie - Limites [de la rétro-ingénierie](#)
objets de trait versus , [lequel utiliser](#) - [lequel utiliser](#)
collections génériques, chaînes en tant que , [chaînes en tant que collections génériques](#)
énumérations génériques , [Énumérations](#) génériques - [Énumérations génériques](#)
fonctions génériques
avec des paramètres constants , [des fonctions génériques et des paramètres de type](#)
paramètres génériques
constantes , [structures génériques avec paramètres constants](#) , [fonctions génériques et paramètres de type](#)
structures génériques , [Ce qu'est réellement l'ensemble de Mandelbrot](#) , [Structures](#) génériques - [Structures génériques](#)
swaps génériques , [Tuples](#)
types génériques

à paramètres constants , [structures génériques à paramètres constants](#)
méthode get , [mutabilité intérieure](#)
Fonction get_form , [Concurrence](#)
Fonction get_index , [Servir des pages sur le Web](#)
git2-rs crate , [Fonctions étrangères](#)
boucle d'événements globale par rapport aux exécuteurs Rust , [Comparaison des conceptions asynchrones](#)
variables globales , [Variables globales](#) - [Variables globales](#)
utilitaire grep , [lecture des lignes](#)
gardes , gardes de [match](#)

H

plages semi-ouvertes (exclusives à la fin) , [champs et éléments](#)
gammes semi-ouvertes , [possibilités multiples assorties](#)
méthode handle.join , [Gestion des erreurs sur les threads](#)
méthode de hachage , [Utilisation d'un algorithme de hachage personnalisé](#)
Trait de hachage , [Hachage - Utilisation d'un algorithme de hachage personnalisé](#)
Hachage , [hachage](#)
Trait HashMap , [Structures de données riches à l'aide d'énumérations](#)
HashMap::with_capacity , [HashMap<K, V> et BTreeMap<K, V>](#)
Type de collection HashMap<K, V> , [HashMap<K, V> et BTreeMap<K, V> - Itération de carte](#)
HashSet::new , [HashSet<T> et BTreeSet<T>](#)
HashSet::with_capacity , [HashSet<T> et BTreeSet<T>](#)
Type de collection HashSet<T> , [HashSet<T> et BTreeSet<T> - Opérations sur l'ensemble](#)
Table de hachage , [partage contre mutation](#)
méthode heap.peek , [BinaryHeap<T>](#)
méthode heap.peek_mut , [BinaryHeap<T>](#)
Méthode heap.pop , [BinaryHeap<T>](#)
méthode heap.push , [BinaryHeap<T>](#)
littéral numérique hexadécimal , [types entiers](#) , [caractères](#)
hexadécimal, formatage des nombres dans , [Formatage des nombres](#)
Caisse de client HTTP , [un vrai client HTTP asynchrone](#)
macros hygiéniques , [cadrage et hygiène](#)

if expression , [Rust Functions](#) , [if et match](#) - [if let](#)
si let expressions , [si let](#)
fichiers image, pour l'ensemble de Mandelbrot , [écriture de fichiers image](#) - [écriture de fichiers image](#)
espace image, mappage au plan des nombres complexes , [mappage des pixels aux nombres complexes](#)
références immuables , [Références](#)
bloc impl , [Définir des méthodes avec impl](#) - [Fonctions associées au type](#)
trait impl , trait [impl](#) - trait [impl](#)
importations , [chemins et importations](#)
Méthode inbound.lines , [Réception de paquets : Plus de flux asynchrones](#)
comprendre! macro , [Macros intégrées](#)
inclure_octets ! macro , [Macros intégrées](#)
include_str ! macro , [Macros intégrées](#)
Index trait , [Index et IndexMut](#) - [Index et IndexMut](#)
contenu indexé , [Déplacements et contenu indexé](#) - [Déplacements et contenu indexé](#) , [Référence aux arguments par index ou nom](#) , [Envoi de valeurs](#) - [Exécution du pipeline](#) , [Transfert de presque tous les itérateurs vers un canal](#) - [Transfert de presque tous les itérateurs vers un canal](#)
IndexMut trait , [Index et IndexMut](#) - [Index et IndexMut](#)
boucles infinies , [Boucles](#)
Attribut #[inline] , [Attributs](#)
doublure , [Performance de fermeture](#)
entrée et sortie , [Entrée et sortie](#) - Mise en [réseau](#)
fichiers et répertoires , [Fichiers et répertoires](#) - [Fonctionnalités spécifiques à la plate-forme](#)
réseautage , [réseautage](#) - [réseautage](#)
lecteurs et écrivains , [Lecteurs et écrivains](#) - [Données binaires, compression et sérialisation](#)
inspecter l'adaptateur , [inspecter](#)
installation, Rust , [rustup et Cargo](#) - [rustup et Cargo](#)
Littéraux entiers , [Types entiers](#) , Types à virgule [flottante](#) , [Littéraux, Variables et Caractères génériques dans les modèles](#)
types entiers , [Types](#) entiers - Types [entiers](#) , [Formatage des nombres](#)

entiers , [Fonctions de Rust](#)

conversion de caractères vers/de , [Conversions vers et depuis des entiers](#)

conversion en pointeurs bruts , [Pointeurs bruts](#)

division par zéro panique , [opérateurs arithmétiques, binaires, de comparaison et logiques](#)

Points communs de type Rust et C , [Trouver des représentations de données communes](#)

tests d'intégration , [Tests d'intégration](#)

mutabilité intérieure , [Rc et Arc : Propriété partagée](#) , [Mutabilité intérieure](#) - [Mutabilité intérieure](#)

types d'adresses de protocole Internet, formatage , [Formatage d'autres types](#)

Dans le trait , [De et Dans](#) - [De et Dans](#)

IntoIter, type associé de , [The Iterator et IntoIterator Traits](#)

Trait IntoIterator , [Les traits Iterator et IntoIterator](#) - [Les traits Iterator et IntoIterator](#) , [Implémentations IntoIterator](#) - [Implémentations IntoIterator](#) , [Implémentation de vos propres itérateurs](#)

into_iter iterator , [Un programme Mandelbrot concurrent](#)

Méthode into_iter , [Les traits Iterator et IntoIterator](#)

erreurs d'invalidation , [Collections](#) , [Rust exclut les erreurs d'invalidation](#)

invariants, mutex et , [Qu'est-ce qu'un mutex ?](#) , [mutex empoisonnés](#)

index inversé , [envoi de valeurs](#) - [exécution du pipeline](#) , [transfert de presque tous les itérateurs vers un canal](#) - [transfert de presque tous les itérateurs vers un canal](#)

invoking wakers, in spawn_blocking , [Invoking Wakers: spawn blocking](#) - [Invoking Wakers: spawn blocking](#)

module io , [Lecteurs et écrivains](#)

Type IpAddr , [Analyse d'autres types à partir de chaînes](#) , [Formatage d'autres types](#)

modèles irréfutables , [où les modèles sont autorisés](#)

type isize , [Types entiers](#)

déclarations d'articles , [Déclarations](#)

éléments , [Modules](#) , [Attributs](#) - [Attributs](#)

méthode iter , méthodes [iter et iter_mut](#) , [implémentations IntoIterator](#)

Méthode iter.collect , [HashMap<K, V> et BTreeMap<K, V>](#) , [HashSet<T> et BTreeSet<T>](#) , [Création de valeurs de chaîne](#)

type itérable , [les traits Iterator et IntoIterator](#)

itérer

emprunt et , [by_ref](#)

sur une carte , [Itération de carte](#)

sur des ensembles , [itération d'ensemble](#)

sur texte , [Itération sur texte](#) - [Itération sur texte](#)

adaptateurs d'itérateur (voir méthodes d'adaptateur)

Méthodes Iterator , [Les traits Iterator et IntoIterator](#)

Trait Iterator , [Traits qui définissent les relations entre les types](#) , [Les traits Iterator et IntoIterator](#) , [Implémentation de vos propres itérateurs](#)

itérateurs , [Gestion des arguments de la ligne de commande](#) , [Itérateurs](#) - [Implémentation de vos propres itérateurs](#)

méthodes d'adaptateur , [adaptateurs d'itérateur](#) - [cycle](#)

types associés et , [Types associés \(ou Fonctionnement des itérateurs\)](#) - [Types associés \(ou Fonctionnement des itérateurs\)](#)

consommer (voir consommer des itérateurs)

création , [Création d'itérateurs](#) - [Autres sources d'itérateurs](#)

implémentation pour vos propres types , [Implémentation de vos propres itérateurs](#) - [Implémentation de vos propres itérateurs](#)

dans la bibliothèque standard , [Autres sources d'itérateurs](#)

traits , [Les traits Iterator et IntoIterator](#) - [Les traits Iterator et IntoIterator](#)

méthode iter_mut , méthodes [iter](#) et [iter_mut](#) , [implémentations IntoIterator](#)

J

Java

ConcurrentModificationException , [partage contre mutation](#)

relation objet-mutex dans [Qu'est-ce qu'un](#) mutex ?

JavaScript, fonction asynchrone , [Comparaison de conceptions asynchrones](#)

méthode de jointure

combinaison de chaînes , [Chaîne](#)

sur les itérateurs parallèles rayon , [Rayon](#)

en attente de thread , [spawn et join](#)

JSON (JavaScript Object Notation) , [Structures de données riches à l'aide d'énumérations](#)

json ! macro , [Construire le json! Macro](#) - [Importation et exportation de macros](#)

types de fragments , [Types](#) de fragments - [Types de fragments](#)
importation et exportation , [Importation et exportation de macros](#) -
[Importation et exportation de macros](#)
récursivité dans , Récursivité [dans les macros](#)
cadrage et hygiène , [cadrage et hygiène](#) - [cadrage et hygiène](#)
utiliser des traits avec , [Utiliser des traits avec des macros](#) - [Utiliser des traits avec des macros](#)

K

Garde, Daniel

Le petit livre des macros Rust , Au- [delà des macro règles !](#)
argument clé, map , [HashMap<K, V>](#) et [BTreeMap<K, V>](#)

L

traits d' extension du langage , traits [utilitaires](#)
dernière méthode , [dernière](#)
Jeu de caractères Latin-1 , [ASCII, Latin-1 et Unicode](#)
lazy_static crate , [Création de valeurs Regex paresseusement](#) , [Variables globales](#)
len method , [Vectors](#) , [Strings in Memory](#) , [position, rposition et ExactSizeIterator](#)
Instruction let , [Fonctions Rust](#) , [Plus d'opérations qui bougent](#) ,
[Déclarations](#)
Li, Peng , [les programmeurs systèmes peuvent avoir de belles choses](#)
libgit2 , [Fonctions étrangères](#) , [Utilisation des fonctions des bibliothèques](#) - [Utilisation des fonctions des bibliothèques](#)
interface brute vers , [Une interface brute vers libgit2](#) - [Une interface brute vers libgit2](#)
interface sécurisée vers , [Une interface sécurisée vers libgit2](#) - [Une interface sécurisée vers libgit2](#)
bibliothèques , [Transformer un programme en bibliothèque](#) - [Transformer un programme en bibliothèque](#)
doc-tests , [Doc-Tests](#) - [Doc-Tests](#)
documentation , [Documentation](#) - [Doc-Tests](#)
fonctions étrangères de , [Utilisation des fonctions des bibliothèques](#) - [Utilisation des fonctions des bibliothèques](#)
Répertoire src/bin , [Le répertoire src/bin](#) - [Le répertoire src/bin](#)
tiers (voir caisses)

durée de vie

paramètres pour les fonctions génériques , les fonctions [génériques et les paramètres de type](#)

paramètres pour les références , [Sécurité des références](#) , [Réception de références en tant qu'arguments de fonction](#) - [Omission des paramètres de durée de vie](#)

contraintes de référence , [pointeurs bruts](#)

structs with , [Generic Structs with Lifetime Parameters](#)

ligne! macro , [Macros intégrées](#)

méthodes lines, sur les flux d'entrée , [try_fold et try_rfold](#) , [Prise d'entrée utilisateur : flux asynchrones](#)

Attribut #[link] , [Utilisation des fonctions des bibliothèques](#)

Linux

Paquet de rouille pour , [rustup et Cargo](#)

Utilisation des fonctions des bibliothèques , [Utilisation des fonctions des bibliothèques](#)

Littéraux , dans les modèles , [Littéraux, Variables et Caractères génériques dans les modèles](#)

Le Petit Livre des Macros Rust (Keep) , Au- [delà des macro règles !](#)

méthode de verrouillage , [Mutex<T>](#)

verrouillage des données

mutex , [Mutex<T>](#) - [Canaux multiconsommateurs utilisant des mutex](#)

verrous de lecture/écriture , verrous de [lecture/écriture \(RwLock<T>\)](#)

enregistrement

canaux pour , Au- [delà des pipelines](#)

pointeurs de formatage pour , [Utilisation du langage de formatage dans votre propre code](#)

valeurs de formatage pour , [Valeurs de formatage pour le débogage](#)
opérateurs logiques , opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

log_syntaxe ! macro , [Macros de débogage](#)

calculs de longue durée, programmation asynchrone, calculs de

longue durée : [yield_now et spawn_blocking](#) - calculs de [longue durée : yield_now et spawn_blocking](#)

boucle (pour les boucles infinies) , [Boucles](#)

expressions de bouclage , [Boucles](#) - [Flux de contrôle dans les boucles](#)

lvalues , [champs et éléments](#)

langage machine , [Trouver des représentations de données communes](#)
types de machines, types entiers , [Types entiers](#) - [Types entiers](#)
mot machine , [Types numériques à largeur fixe](#)
macOS

Paquet de rouille pour , [rustup et Cargo](#)

Utilisation des fonctions des bibliothèques , [Utilisation des fonctions des bibliothèques](#)

macros , [Macros](#) - Au- [delà des macro règles !](#)

intégré , Macros [intégrées](#) - [Macros intégrées](#)

débogage , [Macros](#) de débogage - [Macros de débogage](#)

expansion , [Macros](#) , [Principes de base de l'expansion de macro](#) - [Principes de base de l'expansion de macro](#)

types de fragments , [Types](#) de fragments - [Types de fragments](#)

importation et exportation , [Importation et exportation de macros](#) - [Importation et exportation de macros](#)

json ! , [Construire le json! Macro](#) - [Importation et exportation de macros](#)

procédural , Au- [delà des macro règles !](#)

récurtivité dans , Récurtivité [dans les macros](#)

répétition , [Répétition](#) - [Répétition](#)

cadrage et hygiène , [cadrage et hygiène](#) - [cadrage et hygiène](#)

conséquences imprévues , [Conséquences](#) imprévues - [Conséquences imprévues](#)

utiliser des traits avec , [Utiliser des traits avec des macros](#) - [Utiliser des traits avec des macros](#)

macro_règles ! , Principes de [base des macros](#) , [Types de fragments](#)

fonction main , [Gestion des arguments de la ligne de commande](#) , [Servir des pages sur le Web](#) , [Gestion des erreurs dans main\(\)](#)

Ensemble de Mandelbrot , [Concurrence](#) - [Ce qu'est réellement l'ensemble de Mandelbrot](#)

implémentation simultanée , [Concurrency](#) - [Safety Is Invisible](#)

mappage des pixels aux nombres complexes , [Mappage des pixels aux nombres complexes](#)

analyse des arguments de ligne de commande d'une paire , [Analyse](#)
d'arguments de ligne de commande d'une paire - [Analyse d'arguments de ligne de commande d'une paire](#)

traçage , [Tracé de l'ensemble](#)

rendu avec parallélisme fork-join , [Revisiting the Mandelbrot Set](#) -

[Revisiting the Mandelbrot Set](#)

exécution du traceur , [Exécution du traceur de Mandelbrot](#)

écriture de fichiers image , [Écriture de fichiers image](#) - [Écriture de fichiers image](#)

méthodes map (HashMap et BTreeMap)

append method , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode btree_map.split_off , [HashMap<K, V> et BTreeMap<K, V>](#)

clear method , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode contains_key , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode d'entrée (clé) , [Entrées](#)

méthode entry(key).and_modify , [Entrées](#)

méthode entry(key).or_default , [Entrées](#)

méthode entry(key).or_insert , [Entrées](#)

méthode entry(key).or_insert_with , [Entrées](#)

méthode d'extension , [HashMap<K, V> et BTreeMap<K, V>](#)

get method , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode get_mut , [HashMap<K, V> et BTreeMap<K, V>](#)

insert method , [HashMap<K, V> et BTreeMap<K, V>](#)

Méthode into_iter , [Itération de la carte](#)

méthode into_keys , [itération de la carte](#)

Méthode into_values , [itération de la carte](#)

méthode is_empty , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode keys , [Map Itération](#)

méthode len , [HashMap<K, V> et BTreeMap<K, V>](#)

remove method , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode remove_entry , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode de conservation , [HashMap<K, V> et BTreeMap<K, V>](#)

méthode des valeurs , [Map Itération](#)

méthode values_mut , [itération de la carte](#)

adaptateur de carte , [carte et filtre](#) - [carte et filtre](#)

carte et cartographie , [Mappage des pixels aux nombres complexes](#)

BTreeMap<K, V> , [Overview](#) , [HashMap<K, V> et BTreeMap<K, V>](#) - [Itération de carte](#)

adaptateurs filter_map et flat_map , [map et filter](#) - [filter_map et flat_map](#)

méthode find_map , [find, rfind et find_map](#)

Trait HashMap , [Structures de données riches à l'aide d'énumérations](#)

HashMap<K, V> , [Vue d'ensemble](#) , [HashMap<K, V>](#) et [BTreeMap<K, V>](#) - [Itération de carte](#)

mapper et filtrer , [mapper et filtrer](#) - [mapper et filtrer](#)

traits de marqueur , traits [utilitaires](#) , [sécurité des threads : envoyer et synchroniser](#) - [sécurité des threads : envoyer et synchroniser](#) , [mais votre futur implémente-t-il des envois ?](#) - [Mais votre futur outil envoie-t-il ?](#) , [Le trait de détachement](#) - [Le trait de détachement](#) , [Traits dangereux](#)

expression de correspondance , [analyse d'arguments de ligne de commande de paire](#) , [si et correspondance](#) , [modèles](#) - [modèles](#)

déclaration de correspondance , [écriture de fichiers image](#)

allumettes! macro , [Macros intégrées](#)

unions assorties , unions [assorties](#)

Matsakis, Niko , [Rayonne](#)

méthode max , [max](#), [min](#)

méthode max_by , [max_by](#), [min_by](#)

méthode max_by_key , [max_by_key](#), [min_by_key](#)

Type MaybeUninit , [une interface brute vers libgit2](#)

mémoire , [propriété et déménagements](#)

(voir aussi propriété)

disposition de la fermeture dans , [Performances de fermeture](#)

enums in , [Enums in Memory](#)

pointeurs bruts et , [Entrer et sortir de la mémoire](#) - [Entrer et sortir de la mémoire](#)

réinterpréter avec les syndicats , [Réinterpréter la mémoire avec les syndicats](#) - [Syndicats d'emprunt](#)

chaînes en , [chaînes en mémoire](#) - [chaînes en mémoire](#)

types pour représenter une séquence de valeurs dans , des [tableaux](#), [des vecteurs et des tranches](#) - [tranches](#)

ordonnancement de la mémoire, pour les opérations atomiques , [Atomics](#)

méthodes

appelant , [Appels de fonction et de méthode](#)

définir avec impl , [Définir des méthodes avec impl](#) - [Fonctions associées au type](#)

Appels de méthode entièrement qualifiés , Appels de méthode entièrement qualifiés - [Appels de méthode entièrement qualifiés](#)

méthode min , [max](#), [min](#)

méthode min_by , [max_by](#), [min_by](#)

méthode `min_by_key` , [max_by_key](#), [min_by_key](#)

Modèle-Vue-Contrôleur (voir MVC)

modules , [Modules](#) - [Statique et constantes](#)

bibliothèques et , [Transformer un programme en bibliothèque](#) -

[Transformer un programme en bibliothèque](#)

imbriqués , [modules imbriqués](#)

chemins et importations , [Chemins et importations](#) - [Chemins et importations](#)

prélude , [Modules](#)

dans des fichiers séparés , [Modules dans des fichiers séparés](#)

prélude standard , [Le prélude standard](#)

monomorphisation , [fonctions génériques et paramètres de type](#)

Ver Morris , les [programmeurs systèmes peuvent avoir de belles choses](#)

déplacements , [déplacements](#) - [déplacements et contenu indexé](#)

fermetures et , [Fermetures qui volent](#)

construire de nouvelles valeurs , [plus d'opérations qui bougent](#)

flux de contrôle et , [mouvements et flux de contrôle](#)

Types de copie comme exception à , [Types de copie : l'exception](#)

[aux mouvements](#) - [Types de copie : l'exception aux mouvements](#)

contenu indexé et , [Déplacements et contenu indexé](#) - [Déplace-](#)

[ments et contenu indexé](#)

passer des valeurs à une fonction , [plus d'opérations qui bougent](#)

renvoyer des valeurs à une fonction , [plus d'opérations qui bougent](#)

assignation à une variable , [Plus d'opérations qui bougent](#)

module `mpsc` (multiproducteur, monoconsommateur) , [envoi de valeurs](#) , [fonctionnalités et performances](#) des canaux , [canaux multicon-](#)
[sommateurs utilisant des mutex](#)

`Mul` (trait de multiplication) , [Traits génériques \(ou Comment fonctionne la surcharge de l'opérateur\)](#)

plusieurs lecteurs , [Références aux valeurs](#)

programmation multithread , [Concurrence](#) - [Concurrence](#) , [Sécurité des threads : envoi et synchronisation](#)

(voir aussi programmation asynchrone ; concurrence)

Mot- clé `mut` (mutable) , [Fonctions Rust](#)

référence `mut` (mutable) , [mut et Mutex](#)

mutabilité, intérieur , [Mutabilité](#) intérieure - [Mutabilité intérieure](#)

références mutables (`&mut T`) , [Références](#) , [Références aux valeurs](#)

`FnMut` , [FnMut](#) - [FnMut](#)

Implémentation IntoIterator , [Implémentations IntoIterator](#)
Mutex et , [mut et Mutex](#)
règles pour , [partage contre mutation](#)
références partagées versus , [Références aux valeurs](#) , [Partage ver-
sus mutation](#) - [Partage versus mutation](#)
Fractionnement et , [Fractionnement](#) - [Fractionnement](#)
tranche mutable , [tableaux, vecteurs et tranches](#)
état mutable, partagé , [État mutable partagé](#) - [Variables globales](#)
statique mutable , [réception de références en tant qu'arguments de
fonction](#) , [statiques et constantes](#) , [blocs non sécurisés](#)
Type de mutex , [La table de groupe : mutex synchrones](#) - [La table de
groupe : mutex synchrones](#)
Mutex::new , [Mutex<T>](#)
mutex , [Qu'est-ce qu'un mutex ?](#) - [Qu'est-ce qu'un mutex ?](#)
connexions de chat avec des mutex asynchrones , [Gestion des
connexions de chat : mutex asynchrones](#) - [Gestion des connexions
de chat : mutex asynchrones](#)
création avec Mutex<T> , [Mutex<T>](#) - [Mutex<T>](#)
impasses et , [impasse](#)
invariants et , [Qu'est-ce qu'un mutex ?](#) , [mutex empoisonnés](#)
limitations , [pourquoi les mutex ne sont pas toujours une bonne
idée](#)
canaux multiconsommateurs utilisant , [Canaux multiconsomma-
teurs utilisant des mutex](#)
référence mut et , [mut et Mutex](#)
empoisonné , [mutex empoisonnés](#)
MVC (Model-View-Controller) , [Utilisation efficace des fermetures](#)

N

Structures de champs nommés , [Structures de champs nommés](#) -
Structures [de champs nommés](#)
espaces de noms (voir modules)
Valeurs NaN (not-a-number) , [Comparaisons d'équivalence](#)
Caisse native_tls , Mise en [réseau](#)
modules imbriqués , [Modules imbriqués](#)
module net , Mise en [réseau](#)
réseautage , [réseautage](#) - [réseautage](#)
newtypes , [Tuple-Like Structs](#) , [Exemple : Un type de chaîne ASCII
efficace](#)

méthode suivante , [Types associés \(ou fonctionnement des itérateurs\)](#) ,
[by_ref](#) , [Prise d'entrée utilisateur : flux asynchrones](#)
références non mut, fractionnement , [Fractionnement](#) -
[Fractionnement](#)
normalisation, Unicode , [Normalisation](#) - [La caisse de normalisation](#)
[unicode](#)
valeurs not-a-number (NaN) , [comparaisons d'équivalence](#)
Méthodes nth et nth_back , [nth, nth_back](#)
nième numéro de triangle , [itérateurs](#)
pointeurs nuls , [les références ne sont jamais nulles](#)
Pointeurs bruts nuls , [Pointeurs bruts](#) , Pointeurs [nullables](#)
références nulles non autorisées , [Références](#) , [Les références ne sont](#)
[jamais nulles](#)
nombres, complexes , [Analyse d'arguments de ligne de commande de](#)
[paires](#)
types numériques
Fixed -width , [Types numériques](#) à largeur fixe - Types à virgule
[flottante](#)
types à virgule flottante , Types à virgule [flottante](#) - [Types à virgule](#)
[flottante](#)
types entiers , [Types](#) entiers - Types [entiers](#) , [Formatage des](#)
[nombres](#)

O

OccupiedEntry type, HashMap et BTreeMap , [Entries](#)
littéral numérique octal , [types entiers](#)
octal, formatage des nombres dans , [Formatage des nombres](#)
méthode offset , [Dangereux de quoi ?](#) , [Pointeurs bruts](#) , [Déréféré-](#)
[ment des pointeurs bruts en toute sécurité](#) , [Arithmétique des](#)
[pointeurs](#)
Une règle de définition , [les traits et les autres types de personnes](#)
Structure OpenOptions , [Fichiers](#)
surcharge d'opérateur , [Surcharge](#) d'opérateur - [Autres opérateurs](#)
opérateurs arithmétiques/au niveau du bit , Opérateurs [arithmé-](#)
[tiques et au niveau du bit](#) - [Opérateurs d'affectation composés](#)
opérateurs binaires , [Opérateurs binaires](#)
opérateurs d'affectation composés , [Opérateurs d'affectation](#) com-
posés - Opérateurs d' [affectation composés](#)

tests d'égalité , [Comparaisons d' équivalence](#) - [Comparaisons d'équivalence](#)

traits génériques et , [Traits génériques \(ou Fonctionnement de la surcharge de l'opérateur\)](#)

Index et IndexMut , [Index et IndexMut](#) - [Index et IndexMut](#)

limitations sur , [autres opérateurs](#)

comparaisons ordonnées , [Comparaisons ordonnées](#) - [Comparaisons ordonnées](#)

opérateurs unaires , [opérateurs unaires](#)

priorité des opérateurs , [Priorité et Associativité](#)

les opérateurs

arithmétique , [Opérateurs arithmétiques, binaires, de comparaison et logiques](#) , Opérateurs [arithmétiques et binaires](#) - Opérateurs [arithmétiques et binaires](#) , Opérateurs d' [affectation](#) composés - Opérateurs [d'affectation composés](#)

en tant qu'opérateur , [Conversions vers et à partir d'entiers](#)

Opérateurs binaires , [arithmétiques, binaires, de comparaison et logiques](#) , Opérateurs [binaires](#)

opérateurs au niveau du bit , [arithmétiques, au niveau du bit, de comparaison et logiques](#) , opérateurs [d'affectation](#) composés - opérateurs [d'affectation composés](#)

comparaison , [Type booléen](#) , [Utilisation de chaînes](#) , [Comparaison de références](#) , Opérateurs arithmétiques, binaires, de [comparaison et logiques](#) , [max by, min by](#)

égalité , [Comparaisons d' équivalence](#) - [Comparaisons d'équivalence](#)

référence , [Opérateurs de référence](#)

unaire , [opérateurs unaires](#)

Option<&T> , [les références ne sont jamais nulles](#)

option_env ! macro , [Macros intégrées](#)

opérateurs de comparaison ordonnés , [Comparaisons](#) ordonnées - [Comparaisons ordonnées](#)

Ordre ::SeqCst, ordre de la mémoire atomique , [Atomique](#)

règle orpheline , [traits et autres types de personnes](#)

module os , [Fonctionnalités spécifiques à la plate-forme](#)

Type de chaîne OsStr , [OsStr et Path](#) - [OsStr et Path](#)

Type sortant , [gestion des connexions de chat : mutex asynchrones](#)

opérations de débordement , [Arithmétique vérifiée, enveloppante, saturée et débordante](#)

Propriété , [Propriété et Déménagements](#) - [Rc et Arc : Copropriété](#)
Arc , [Rc et Arc : Copropriété](#) - [Rc et Arc : Copropriété](#)
C++ versus Rust , [Propriété](#) - [Propriété](#)
Vache , [Emprunter et Propriétaire au travail : la vache humble](#)
itération et , [Gestion des arguments de la ligne de commande](#)
déplacements , [déplacements](#) - [déplacements et contenu indexé](#)
Rc , [Rc et Arc : Copropriété](#) - [Rc et Arc : Copropriété](#)
partagé , [Rc et Arc : Propriété partagée](#) - [Rc et Arc : Propriété](#)
[partagée](#)
type propriétaire , [OsStr et Path](#)

P

panique , [Fonctions Rust](#) , [Panique](#) - [Abandon](#)
abandon , [Abandon](#)
mutex empoisonnés , mutex [empoisonnés](#)
sécurité dans un code dangereux , [Sécurité panique dans un code](#)
[dangereux](#)
dérouler , [dérouler](#) - [dérouler](#)
panique! macro , [Panique](#) , [Valeurs de formatage](#) , [Macros](#)
programmation parallèle , la programmation [parallèle est apprivoisée](#)
(voir aussi concurrence)
ParallèleItérateur , [Rayon](#)
paramètres
formatage , [Valeurs](#) de formatage , Valeurs [de formatage pour le débogage](#) , [Largeurs et précisions dynamiques](#)
lifetime , [Reference Safety](#) , [Recevoir des références comme arguments de fonction](#) - [Omettre les paramètres](#) de durée de vie , les
[fonctions génériques et les paramètres de type](#)
type , [Analyse d'arguments de ligne de commande de paires](#) , [Structures](#) génériques , [Fonctions génériques et paramètres de type](#) ,
[Comparaisons d'équivalence](#)
méthode d' analyse , [analyse d'autres types à partir de chaînes](#)
fonction parse_args , [lecture et écriture de fichiers](#)
fonction parse_complex , [analyse des arguments de ligne de commande de la paire](#)
fonction parse_pair , [Analyse des arguments de la ligne de commande d'une paire](#)
Trait PartialEq , [Comparaisons](#) d'équivalence - [Comparaisons](#)
[d'équivalence](#)

Trait PartialOrd , [Comparaisons](#) ordonnées - [Comparaisons ordonnées](#)
méthode de partition , [partition](#)
méthode part_iter , [Rayon](#)
Type de chemin , [méthodes iter et iter_mut](#) , [formatage des valeurs de texte](#) , [OsStr et Path](#) - [méthodes Path et PathBuf](#)
méthode des ancêtres , [Méthodes Path et PathBuf](#)
méthode des composants , [Méthodes Path et PathBuf](#)
méthode d'affichage , [Méthodes Path et PathBuf](#)
méthode nom_fichier , [Méthodes Path et PathBuf](#)
méthode is_absolute , [Méthodes Path et PathBuf](#)
méthode is_relative , [Méthodes Path et PathBuf](#)
join method , [Path et PathBuf Methods](#)
méthode parent , [Méthodes Path et PathBuf](#)
méthode to_str , méthodes [Path et PathBuf](#)
méthode to_string_lossy , [Méthodes Path et PathBuf](#)
Path::new method , [Méthodes Path et PathBuf](#)
Type PathBuf , [Méthodes Path et PathBuf](#)
chemins, bibliothèque standard , [Chemins et importations](#) - [Chemins et importations](#)
patrons , [Patrons](#) - [La vue d'ensemble](#)
@ Patterns , [Reliure avec @ Patterns](#)
array , [Array et Slice Patterns](#)
éviter les erreurs de syntaxe lors de la correspondance dans les macros , [Éviter les erreurs de syntaxe lors de la correspondance](#)
gardes , gardes de [match](#)
littéraux dans , [littéraux, variables et caractères génériques dans les modèles](#)
correspondance des expressions et , [si et correspondance](#)
faire correspondre plusieurs possibilités avec , [Faire correspondre plusieurs possibilités](#)
remplir un arbre binaire , [Remplir un arbre binaire](#)
référence , [Motifs](#) de référence - [Motifs de référence](#)
recherche et remplacement , [Modèles de recherche de texte](#) - [Recherche et remplacement](#)
situations qui permettent , [Où les modèles sont autorisés](#)
slice , [Array et Slice Patterns](#)
Modèles struct , [tuple et struct](#)
Tuple , [Tuple et Struct Patterns](#)

variables dans , [littéraux, variables et caractères génériques dans les modèles](#)

caractères génériques dans , [littéraux, variables et caractères génériques dans les modèles](#)

méthode peek , [peekable](#)

Itérateur peekable , [peekable](#)

Type de broche , [pointeurs épinglés](#)

pointeur épinglé , Pointeurs épinglés - [Pointeurs épinglés](#)

épingler des contrats à terme , [Épingler](#) - [Le trait de détachement](#)

approche pipeline

programmation simultanée , [Concurrence](#)

étapes de l'itérateur , [Itérateurs](#) , [Création d'itérateurs](#)

plusieurs threads , [envoi de valeurs](#) - [exécution du pipeline](#) , [fonctionnalités et performances du canal](#) , [transfert de presque tous les itérateurs vers un canal](#) - [transfert de presque tous les itérateurs vers un canal](#)

fonction pixel_to_point , [Un programme de Mandelbrot concurrent](#)

tracer, ensemble de Mandelbrot , [tracer l'ensemble](#) , [exécuter le traqueur de Mandelbrot](#)

Caractéristique de formatage du pointeur , [Pointeurs bruts](#)

types de pointeurs , [Types](#) de pointeurs - [Pointeurs bruts](#)

boîtes , [Boîtes](#)

non-propriétaire , [Références aux valeurs](#)

pointeur épinglé , Pointeurs épinglés - [Pointeurs épinglés](#)

pointeurs bruts , [Pointeurs bruts](#) , Pointeurs [bruts](#) - [Sécurité anti-panique dans le code dangereux](#)

références (voir références (type pointeur))

pointeurs, restrictions de Rust sur , [propriété et déplacements](#)

PoisonError::into_inner , [mutex empoisonnés](#)

poll method , [Futurs](#) , [Long Running Calculs: yield now et spawn blocking](#)

interface d'interrogation, programmation asynchrone , [Futurs](#) - [Futurs](#) , [Comparaison de conceptions asynchrones](#) , [Futurs primitifs et exécuteurs : quand un futur mérite-t-il à nouveau d'être interrogé ?](#) - [Implémentation de block on](#)

polymorphisme , [Traits et Génériques](#)

position method , [position](#), [rposition](#) et [ExactSizeIterator](#)

Fonction post_gcd , [Servir des pages sur le Web](#) , [Concurrence](#)

priorité, opérateur , [priorité et associativité](#)

module prelude , [Modules](#) , [Fonctionnalités spécifiques à la plate-forme](#) , [Prise d'entrée utilisateur : flux asynchrones](#)
imprimer! macro , [Valeurs de formatage](#) , [Writers](#)
imprimez ! macro , [Servir des pages sur le Web](#) , [Chaîne](#) , [Valeurs de formatage](#) , [Rédacteurs](#)
imprimez ! méthode , [Erreurs d'impression](#)
fonction print_error , [Ignorer les erreurs](#)
Fonction print_padovan , [Propriété](#)
fonction print_usage , [L'interface de ligne de commande](#)
macros procédurales , Au- [delà des macro règles !](#)
fonction process_files , [Parallélisme Fork-Join](#)
méthode du produit , [Accumulation simple : compte, somme, produit](#)
profileur , [Construire des profils](#)
propagation des erreurs , [propagation des erreurs](#)
protocole
 client et serveur comme asynchrones , [Le protocole](#) - [Le protocole](#)
 types d'adresses de protocole Internet, formatage , [Formatage d'autres types](#)
module ptr , [Entrer et sortir de la mémoire](#)
ptr.copy_to , Entrer [et sortir de la mémoire](#)
ptr.copy_to_nonoverlapping , [Déplacement dans et hors de la mémoire](#)
ptr::copy , [Entrer et sortir de la mémoire](#)
ptr::copy_nonoverlapping , Entrer [et sortir de la mémoire](#)
ptr::read , [Entrer et sortir de la mémoire](#)
ptr::write , [Entrer et sortir de la mémoire](#)
traits de vocabulaire public , traits [utilitaires](#)
Python, affectation dans , [Moves](#) - [Moves](#)

R

méthode de course , [la fonction principale du client](#)
caisse rand , [éléments aléatoires](#)
méthode rand :: thread_rng , [éléments aléatoires](#)
gammes
 fermé , [champs et éléments](#)
 fin-exclusif , [Possibilités Multiples Assorties](#)
 semi-ouvert , [Champs et éléments](#) , [Correspondance de multiples possibilités](#)
 dans les expressions de boucle , [Boucles](#)
 illimité , [correspondant à de multiples possibilités](#)

module raw , [Trouver des représentations de données communes](#) ,
[Une interface brute vers libgit2](#) - [Une interface brute vers libgit2](#)
pointeurs bruts , [Pointeurs bruts](#) , [Code non sécurisé](#) , [Pointeurs bruts](#) -
[Sécurité anti-panique dans le code non sécurisé](#)
déréférencement , [pointeurs bruts](#) , [dangereux de quoi ?](#) , [Blocs non
sécurisés](#) , [Déréférencement des pointeurs bruts en toute sécurité](#) -
[Exemple : RefWithFlag](#)
Exemple de GapBuffer , [Exemple : GapBuffer](#) - [Exemple : GapBuffer](#)
entrer/sortir de la mémoire , [entrer et sortir de la mémoire](#) - [entrer
et sortir de la mémoire](#)
pointeurs nullables , [pointeurs nullables](#)
sécurité anti-panique dans le code non sécurisé , [Sécurité anti-pa-
nique dans le code non sécurisé](#)
arithmétique du pointeur , [Arithmétique du pointeur](#) - [Arithmé-
tique du pointeur](#)
Exemple RefWithFlag , [Exemple : RefWithFlag](#) - [Exemple :
RefWithFlag](#)
Tailles et alignements des caractères , [Tailles et alignements des
caractères](#)
chaînes brutes , [Littéraux de chaîne](#)
Bibliothèque Rayon (Matsakis et Stone) , [Rayon](#) - [Rayon](#)
Type de pointeur Rc , [Rc et Arc : Propriété partagée](#) - [Rc et Arc : Pro-
priété partagée](#) , [Se passant comme une boîte, Rc ou Arc](#) - [Se passant
comme une boîte, Rc ou Arc](#)
Trait de lecture , [entrée et sortie](#) , [lecteurs et écrivains](#)
méthode des octets , [Lecteurs](#)
méthode de la chaîne , [Lecteurs](#)
méthode des lignes , [lecteurs tamponnés](#)
méthode de lecture , [Lecteurs](#)
méthode read_exact , [Lecteurs](#)
méthode read_to_end , [Lecteurs](#)
méthode read_to_string , [Lecteurs](#)
prendre la méthode , [Lecteurs](#)
accès en lecture seule, accès partagé en tant que , [partage contre
mutation](#)
verrous de lecture/écriture (RwLock) , verrous de [lecture/écriture
\(RwLock<T>\)](#)
Trait ReadBytesExt , [données binaires, compression et sérialisation](#)
lecteurs , [Lecteurs](#) - [Collectionner les lignes](#)

données binaires, compression, sérialisation , [Données binaires, compression et sérialisation](#) - [Données binaires, compression et sérialisation](#)

tamponné , [Lecteurs tamponnés](#) - [Lignes de collecte](#)

lignes de collecte , [Lignes de collecte](#)

fichiers , [Fichiers](#)

autres types , [Autres types de lecteurs et d'enregistreurs](#) - [Autres types de lecteurs et d'enregistreurs](#)

lignes de lecture , [Lignes](#) de lecture - [Lignes de lecture](#)

Chercher le trait , [Chercher](#)

Méthode read_dir , [Lecture des répertoires](#)

Fonction read_numbers , [Utilisation de plusieurs types d'erreurs](#)

fonction read_to_string , [lecture et écriture de fichiers](#)

read_unaligned , Entrer [et sortir de la mémoire](#)

read_volatile , Entrer [et sortir de la mémoire](#)

Type de récepteur , [canaux multiconsommateurs utilisant des mutex](#)

réception de paquets, chat asynchrone , [Réception de paquets : Plus de flux asynchrones](#) - [Réception de paquets : Plus de flux asynchrones](#)

récurtivité, macros , Récursivité [dans les macros](#)

Type de cellule de référence

emprunt de méthode , [mutabilité intérieure](#)

méthode emprunter_mut , [mutabilité intérieure](#)

méthode try_borrow , [mutabilité intérieure](#)

méthode try_borrow_mut , [mutabilité intérieure](#)

RefCell::new(value) , [Mutabilité intérieure](#)

Structure RefCell<T> , [Mutabilité intérieure](#)

motifs de référence (réf) , [Motifs](#) de référence - [Motifs de référence](#)

opérateurs de référence , [Opérateurs de référence](#)

type de pointeur à référence comptée (Rc) , [Rc et Arc : Propriété partagée](#) - [Rc et Arc : Propriété partagée](#)

références (type pointeur) , [Références](#) , [Références](#) - [Prendre les armes contre une mer d'objets](#)

affectation , [affectation de références](#)

emprunt , [Emprunter des références à des expressions arbitraires](#) , [Emprunter une variable locale](#) - [Emprunter une variable locale](#) , [Fermetures qui empruntent](#)

C++ versus Rust , [Références Rust versus Références C++](#)

comparer , [Comparer des références](#)

contraintes sur , [Sécurité de référence](#) - [Structures contenant des références](#) , [Pointeurs bruts](#)
immuable , [Références](#)
Implémentation IntoIterator , [Implémentations IntoIterator](#)
itération et , [Gestion des arguments de la ligne de commande](#)
paramètres de durée de vie et , [Sécurité des références](#) , [Réception de références en tant qu'arguments de fonction](#) - [Omission des paramètres de durée de vie](#)
mutable (voir références mutables)
null , [Références](#)
pointeurs nuls et , [les références ne sont jamais nulles](#)
passage de références à des fonctions , [passage de références à des fonctions](#)
réception en tant qu'arguments de fonction , [Réception de références en tant qu'arguments de fonction](#) - [Réception de références en tant qu'arguments de fonction](#)
retour , [retour de références](#)
sécurité de , [Sécurité de référence](#) - [Omettre les paramètres de durée de vie](#)
"mer d'objets" et , [Prendre les armes contre une mer d'objets](#) - [Prendre les armes contre une mer d'objets](#)
partagé versus mutable , [Références aux valeurs](#) , [Partage versus mutation](#) - [Partage versus mutation](#)
structures contenant , [Structures contenant des références](#) - [Structures contenant des références](#)
aux références , [Références aux références](#)
aux tranches et aux objets de trait , [Références aux tranches et aux objets de trait](#)
aux valeurs , [Références aux valeurs](#) - [Références aux valeurs](#)
modèles réfutables , [Où les modèles sont autorisés](#)
RefWithFlag<'a, T> , [Exemple : RefWithFlag](#) - [Exemple : RefWithFlag](#)
Structure Regex , [Rechercher et remplacer](#)
Méthode Regex ::captures , [Utilisation de base de Regex](#)
Regex::new constructor , [Construire des valeurs Regex paresseusement](#) - [Construire des valeurs Regex paresseusement](#)
expressions régulières (regex) , [Expressions régulières](#) - [Création de valeurs régulières paresseusement](#)
utilisation de base , Utilisation de base de Regex - [Utilisation de base de Regex](#)

construire des valeurs à la demande , [construire des valeurs régulières paresseusement](#)

macros versus , [Principes de base de l'expansion des macros](#)
opérateurs relationnels , opérateurs [arithmétiques, binaires, de comparaison et logiques](#)

méthode replace_all , [Rechercher et remplacer](#)

Attribut #[repr(C)] , [Recherche de représentations de données communes](#)

Attribut #[repr(i16)] , [Recherche de représentations de données communes](#)

caisse reqwest , [Réseautage](#)

programmation avec contraintes de ressources , [Préface](#)

Type de résultat , [Gestion des arguments de la ligne de commande](#) ,
[Résultat - Pourquoi des résultats ?](#)

Méthode as_mut , Capture des [erreurs](#)

méthode as_ref , capture des [erreurs](#)

attraper les erreurs , [Gestion](#) des erreurs , [Attraper les erreurs - Attraper les erreurs](#)

traiter les erreurs qui "ne peuvent pas arriver" , [Traiter les erreurs qui "ne peuvent pas arriver"](#)

déclaration d'un type d'erreur personnalisé , [Déclaration d'un type d'erreur personnalisé](#)

méthode err , [Attraper les erreurs](#)

gestion des erreurs sur les threads , [Gestion des erreurs sur les threads](#)

expect method , [Attraper les erreurs](#)

gestion des erreurs dans la fonction main , [Gestion des erreurs dans main\(\)](#)

ignorer les erreurs , [Ignorer les erreurs](#)

Méthode is_err , Capture des [erreurs](#)

Méthode is_ok , Capture des [erreurs](#)

points clés de la conception , [Pourquoi des résultats ?](#)

avec plusieurs types d'erreurs , [Utilisation de plusieurs types d'erreurs - Utilisation de plusieurs types d'erreurs](#)

méthode ok , [Attraper les erreurs](#)

erreurs d'impression , [Erreurs](#) d'impression - [Erreurs d'impression](#)

propagation des erreurs , [propagation des erreurs](#)

alias de type , [Alias de type de résultat](#)

méthode unwrap , [Capture des erreurs](#)

méthode unwrap_or , Capture des [erreurs](#)
méthode unwrap_or_else , capture des [erreurs](#)
expressions de retour , [Fonctions Rust](#) , Fonctions [Rust](#) , [Expressions de retour](#)
adaptateur rev , [Itérateurs réversibles et rev](#) - [Itérateurs réversibles et rev](#)
méthode inverse , [Vecteurs](#)
itérateurs réversibles , [Itérateurs réversibles et rev](#) - [Itérateurs réversibles et rev](#)
méthode rfind , [find, rfind et find_map](#)
méthode rfold , [fold et rfold](#)
Paramètre de type Rhs , [Comparaisons d'équivalence](#)
module racine , [Transformer un programme en bibliothèque](#)
méthode route , [diffusion de pages sur le Web](#)
routeurs, rappels et , [Rappels](#) - [Rappels](#)
méthode rposition , [position, rposition et ExactSizeIterator](#)
Rust , [Un tour de Rust](#) - [Rechercher et remplacer](#)
arguments de ligne de commande , [Gestion des arguments de ligne de commande](#)
de commande - [Gestion des arguments de ligne de commande](#)
interface de ligne de commande , [Systèmes de fichiers et outils de ligne de commande](#) - [L'interface de ligne de commande](#)
communauté , [Plus de belles choses](#)
simultanéité , [Concurrence](#) - La [sécurité est invisible](#)
systèmes de fichiers , [Systèmes de fichiers et outils de ligne de commande](#) - [Rechercher et remplacer](#)
rechercher et remplacer , [rechercher et remplacer](#)
fonctions dans , [Fonctions Rust](#) - [Fonctions Rust](#)
installation , [rustup et Cargo](#) - [rustup et Cargo](#)
lecture de fichiers , [Lecture et écriture de fichiers](#)
raisons d'utiliser , [Rust assume la charge pour vous](#) - [Rust facilite la collaboration](#)
règles pour un programme qui se comporte bien , [Comportement indéfini](#)
serveur Web simple , [diffusion de pages sur le Web](#) - [diffusion de pages sur le Web](#)
tests unitaires en , [écriture et exécution de tests unitaires](#)
site web , [rustup et Cargo](#)
commande rustc , [rustup et cargaison](#) , [caisses](#) , [macros de débogage](#)
Commande rustdoc , [rustup et Cargo](#)

rustup , [rustup et Cargo](#) - [rustup et Cargo](#)
RwLock , Verrous en [lecture/écriture \(RwLock<T>\)](#)

S

interface sécurisée vers libgit2 , [Une interface sécurisée vers libgit2](#) -
[Une interface sécurisée vers libgit2](#)

sécurité

fermetures et , [Fermetures et sécurité](#) - [Copier et cloner pour les fermetures](#)

l' invisibilité de , [La sécurité est invisible](#)

avec références , [Sécurité](#) de référence - [Paramètres de durée de vie omis](#)

sécurité des threads avec envoi et synchronisation , [sécurité des threads : envoi et synchronisation](#) - [sécurité des threads : envoi et synchronisation](#)

opérations de saturation , [Arithmétique vérifiée, enveloppante, saturante et débordante](#)

fonction say_hello , fonctions [génériques et paramètres de type](#)
portées et cadrage , [Rayonne](#) , [Cadrage et hygiène](#) - [Cadrage et hygiène](#)
recherche

tranches , [Trier et rechercher](#) , [Rechercher et remplacer](#)
text , [Conventions de recherche et d'itération](#) - [Recherche et remplacement](#)

Chercher le trait , [Chercher](#)

self argument , [Passing Self as a Box, Rc, or Arc](#) - [Passing Self as a Box, Rc, or Arc](#)

mot- clé self , [chemins et importations](#)

Type de soi , [soi dans les traits](#) - [soi dans les traits](#)

points-virgules suivant les expressions , [blocs et points-virgules](#)

Variable SEMVER , [création de valeurs régulières paresseusement](#)

Send marker trait , [Thread Safety : Send and Sync](#) - [Thread Safety : Send and Sync](#) , [Mais votre futur implémente-t-il Send ?](#) - [Mais votre futur outil envoie-t-il ?](#) , [Traits dangereux](#)

send_as_json , [Envoi de paquets](#)

serde library/crate , [Traits et autres types de personnes](#) , [Données binaires, compression et sérialisation](#) , [Un client et un serveur asynchrones](#)

serde_json crate , [Structures de données riches utilisant des énumérations](#) , [Données binaires, compression et sérialisation](#) , [Un client et un](#)

[serveur asynchrones](#)

sérialisation , [données binaires, compression et sérialisation](#)

méthode ensembliste , [mutabilité intérieure](#)

définir les types (HashMap et BTreeMap)

contient method , [HashSet<T> et BTreeSet<T>](#)

méthode des différences , [Opérations sur l'ensemble](#)

méthode get , [lorsque des valeurs égales sont différentes](#)

insert method , [HashSet<T> et BTreeSet<T>](#)

méthode d'intersection , [Opérations sur l'ensemble](#)

Méthode is_disjoint , [Opérations sur l'ensemble](#)

méthode is_empty , [HashSet<T> et BTreeSet<T>](#)

Méthode is_subset , [Opérations sur l'ensemble](#)

Méthode is_superset , [Opérations sur l'ensemble](#)

méthode iter , [Définir l'itération](#)

méthode len , [HashSet<T> et BTreeSet<T>](#)

remove method , [HashSet<T> et BTreeSet<T>](#)

méthode de remplacement , [lorsque des valeurs égales sont différentes](#)

méthode de conservation , [HashSet<T> et BTreeSet<T>](#)

méthode symmetric_difference , [Opérations sur l'ensemble](#)

méthode take , [lorsque des valeurs égales sont différentes](#)

méthode union , [Opérations sur l'ensemble](#)

ensembles , [HashSet<T> et BTreeSet<T>](#)

(voir aussi l'ensemble de Mandelbrot)

BTreeSet<T> , [HashSet<T> et BTreeSet<T>](#) - [Opérations sur l'ensemble](#)

HashSet<T> type , [HashSet<T> et BTreeSet<T>](#) - [Opérations sur l'ensemble](#)

ombrage , [Déclarations](#)

accès partagé , [Partage contre mutation](#)

partagé des données immuables entre les threads , [Partage de données immuables entre les threads](#) - [Partage de données immuables entre les threads](#)

état mutable partagé , État [mutable partagé](#) - [Variables globales](#)

atomique , [Atomique](#) , [Variables globales](#)

Variables de condition (Condvar) , [Variables de condition \(Condvar\)](#)

impasse , [impasse](#)

variables globales , [Variables globales](#) - [Variables globales](#)

canaux multiconsommateurs utilisant mutex , [Canaux multiconsommateurs utilisant mutex](#)

mut et Mutex , [mut et Mutex](#)

limitations mutex , [pourquoi les mutex ne sont pas toujours une bonne idée](#)

Mutex<T> , [Mutex<T>](#) - [Mutex<T>](#)

mutex empoisonnés , mutex [empoisonnés](#)

verrous de lecture/écriture (RwLock) , verrous de [lecture/écriture \(RwLock<T>\)](#)

références partagées (&T) , [Références aux valeurs](#)

Les pointeurs de C vers les valeurs const contre , [Partage Versus Mutation](#)

Implémentation IntoIterator , [Implémentations IntoIterator](#)

références mutables versus , [Partage Versus Mutation](#) - [Partage Versus Mutation](#)

règles pour , [Partage contre mutation](#) , [Partage contre mutation](#)

tranche partagée de Ts , [Arrays, Vectors et Slices](#)

Structure partagée, spawn_blocking , [Invocation de Wakers :](#)
[spawn_blocking](#)

Attribut #[should_panic] , [Tests et Documentation](#)

fonction show_it , [Deref et DerefMut](#)

types entiers signés , [Types entiers](#)

auteur unique, règle de plusieurs lecteurs , [Références aux valeurs](#)

SipHash-1-3 , [Utilisation d'un algorithme de hachage personnalisé](#)

Trait calibré , [Dimensionné](#) - [Dimensionné](#) , [Tailles de caractères et alignements](#)

méthode size_hint , [by_ref](#) , Création [de collections : collect et FromIterator](#)

Fonction size_of_val , [Tailles et alignements de type](#)

adaptateurs skip et skip_while , [skip et skip_while](#)

modèles de tranche , Modèles de [tableau et de tranche](#)

tranches , [tranches](#)

&str (tranche de chaîne) , [Chaîne](#)

méthode binary_search , [Tri et recherche](#)

méthode binary_search_by , [tri et recherche](#)

méthode binary_search_by_key , [tri et recherche](#)

emprunt sous d'autres types de texte , [Emprunt sous d'autres types de texte](#)

méthode bytes , [itération sur le texte](#)

conversion de casse pour les chaînes , [Conversion de casse pour les chaînes](#)

Méthode chars , [Itération sur le texte](#)

Méthode char_indices , [Itération sur le texte](#)

choisir la méthode , [éléments aléatoires](#)

méthode des morceaux , [Fractionnement](#)

Méthode chunks_exact , [Fractionnement](#)

Méthode chunks_exact_mut , [Fractionnement](#)

Méthode chunks_mut , [Fractionnement](#)

comparer , [Comparer des tranches](#)

méthode concat , [Joindre](#)

contient la méthode , [Trier et rechercher](#) , [Rechercher et remplacer](#)

méthodes ends_with , [Comparaison de tranches](#) , [Recherche et remplacement](#)

méthode de recherche , [Recherche et remplacement](#)

première méthode , [Accéder aux éléments](#)

Méthode first_mut , [Accès aux éléments](#)

méthode get , [Accéder aux éléments](#)

Méthode get_mut , [Accès aux éléments](#)

Implémentation IntoIterator , [Implémentations IntoIterator](#)

méthode is_char_boundary , [inspection simple](#)

méthode is_empty , [vecteurs croissants et rétrécissants](#) , [inspection simple](#)

méthode iter , [Fractionnement](#)

itérer sur du texte , [Itérer sur du texte](#) - [Itérer sur du texte](#)

Méthode iter_mut , [Fractionnement](#)

méthode join , [Joindre](#)

jointure dans des tableaux de tableaux , [Joindre](#)

dernière méthode , [Accéder aux éléments](#)

Méthode last_mut , [Accès aux éléments](#)

méthode len , [vecteurs croissants et rétrécissants](#) , [inspection simple](#)

méthode des lignes , [Itération sur le texte](#)

correspond à la méthode , [Itération sur le texte](#)

Méthode match_indices , [Itération sur le texte](#)

sortie aléatoire , [éléments aléatoires](#)

Méthode rchunks , [Fractionnement](#)

Méthode rchunks_exact , [Fractionnement](#)

Méthode rchunks_exact_mut , [Fractionnement](#)

Méthode `rchunks_mut` , [Fractionnement](#)
références à , [références aux tranches et aux objets de trait](#)
méthode de remplacement , [Recherche et remplacement](#)
méthode `replacen` , [Recherche et remplacement](#)
méthode inverse , [tri et recherche](#)
Méthode `rfind` , [Recherche et remplacement](#)
Méthode `rmatch_indices` , [Itération sur le texte](#)
Méthode `rsplit` , [Fractionnement](#) , [Itération sur le texte](#)
Méthode `rsplitn` , [Fractionnement](#) , [Itération sur le texte](#)
Méthode `rsplitn_mut` , [Fractionnement](#)
Méthode `rsplit_mut` , [Fractionnement](#)
Méthode `rsplit_terminator` , [Itération sur le texte](#)
`rechercher` , [trier et rechercher](#) , [rechercher et remplacer](#)
méthode `shuffle` , [éléments aléatoires](#)
`slice[range]` , [Inspection simple](#)
méthode de tri , [Tri et recherche](#)
`tri` , [tri et recherche](#)
Méthode `sort_by` , [Tri et recherche](#)
Méthode `sort_by_key` , [Tri et recherche](#)
méthode `split` , [Fractionnement](#) , [Itération sur le texte](#)
méthode `splitn` , [Fractionnement](#) , [Itération sur le texte](#)
Méthode `splitn_mut` , [Fractionnement](#)
fractionnement des références non mut , [Fractionnement](#) -
[Fractionnement](#)
Méthode `split_ascii_whitespace` , [Itération sur le texte](#)
Méthode `split_at` , [Fractionnement](#) , [Inspection simple](#)
Méthode `split_at_mut` , [Fractionnement](#)
Méthode `split_first` , [Fractionnement](#)
Méthode `split_first_mut` , [Fractionnement](#)
Méthode `split_last` , [Fractionnement](#)
Méthode `split_last_mut` , [Fractionnement](#)
Méthode `split_mut` , [Fractionnement](#)
Méthode `split_terminator` , [Itération sur le texte](#)
Méthode `split_whitespace` , [Itération sur le texte](#)
méthodes `starts_with` , [Comparer les tranches](#) , [Rechercher et remplacer](#)
Méthode `strip_prefix` , [Trimming](#)
Méthode `strip_suffix` , [Trimming](#)
méthode d'échange , [Échange](#)

échanger le contenu de , [Échange](#)
Méthode to_lowercase , [Conversion de casse pour les chaînes](#)
Méthode to_owned , [Création de valeurs de chaîne](#)
Méthode to_string , [Création de valeurs de chaîne](#)
Méthode to_uppercase , [Conversion de casse pour les chaînes](#)
Méthode to_vec , [Accès aux éléments](#)
méthode de trim , [Trimming](#)
tailler les cordes , [tailler](#)
méthode trim_matches , [Trimming](#)
UTF-8 et , [Accès au texte en UTF-8 - Production de texte à partir de données UTF-8](#)
méthode windows , [Fractionnement](#)
cas de serpent , [Structures de champ nommé](#)
SocketAddr type , [Mise en forme d'autres types](#)
tri des tranches , [Tri et recherche](#)
fonction d'apparition
pour les tâches asynchrones , [Programmation asynchrone](#) , [Création](#) de tâches asynchrones , Création de tâches asynchrones [sur un pool de threads](#)
pour créer des threads , [spawn and join](#) - [spawn and join](#) , [Error Handling Across Threads](#)
générant des tâches asynchrones , [Générant](#) des tâches asynchrones - [Générant](#) des tâches asynchrones , Générant des tâches asynchrones [sur un pool de threads](#)
spawn_blocking , Calculs de [longue durée : yield now et spawn blocking](#) - Calculs de [longue durée : yield now et spawn blocking](#) , Appel de [Wakers : spawn blocking](#) - Appel de [Wakers : spawn blocking](#)
spawn_local , Génération de [tâches](#) asynchrones - Génération de tâches asynchrones , Génération de [tâches asynchrones sur un pool de threads](#)
méthode d'épissage , [Self in Traits](#)
Répertoire src/bin , [Le répertoire src/bin](#) - [Le répertoire src/bin](#)
dérouler la pile , [Dérouler](#) - [Dérouler](#)
prélude standard , [Le prélude standard](#)
déclarations, expressions versus , [un langage d'expression](#)
mot- clé statique , [Statique et constantes](#) , [Blocs non sécurisés](#)
méthodes statiques , [appels de fonction et de méthode](#)
valeurs statiques (statics) , [Recevoir des références en tant qu'arguments de fonction](#) , [Modules](#)

std (bibliothèque standard) , [Chemins et importations](#)

Type Stderr , [autres types de lecteurs et d'enregistreurs](#)

Type Stdin , [autres types de lecteurs et d'enregistreurs](#)

Type StdinLock , [autres types de lecteurs et d'enregistreurs](#)

Type de sortie standard , [autres types de lecteurs et d'enregistreurs](#)

Pierre, Josh , [Rayonne](#)

str::from_utf8 , [Production de texte à partir de données UTF-8](#)

str::from_utf8_unchecked , [Production de texte à partir de données UTF-8](#)

Caractéristique de flux , [prise d'entrée utilisateur : flux asynchrones](#)
ruisseaux

flux asynchrones , [Réception de paquets : plus de flux asynchrones](#)

- [Réception de paquets : plus de flux asynchrones](#)

client et serveur en tant qu'asynchrone , [Prise d'entrée utilisateur : flux asynchrones](#)

TcpStream , [Autres types de lecteurs et d'enregistreurs](#) , [Réception de paquets : davantage de flux asynchrones](#)

Types String et str , [Analyse d'arguments de ligne de commande de paires](#) , [Types String - Autres types](#) de type chaîne , [String et str - Chaînes en tant que collections génériques](#)

ajouter du texte , [Ajouter et insérer du texte](#) - [Ajouter et insérer du texte](#)

Ascii , [Exemple : Un type de chaîne ASCII efficace](#) - [Fonctions non sécurisées](#)

emprunt du contenu de la tranche , [Emprunter sous d'autres types de texte](#)

chaînes d'octets , chaînes d' [octets](#)

conversion de casse , [Conversion de casse pour les chaînes](#)

clear method , [Suppression et remplacement de texte](#)

conversion de valeurs non textuelles en , [Conversion d'autres types en chaînes](#) - [Conversion d'autres types en chaînes](#)

création de valeurs de chaîne , [Création de valeurs de chaîne](#)

méthode de vidange , [Suppression et remplacement de texte](#)

méthode extend , [Ajout et insertion de texte](#)

types de noms de fichiers , [OsStr et Path](#)

from_utf8 , [Production de texte à partir de données UTF-8](#)

from_utf8_lossy , [Production de texte à partir de données UTF-8](#)

from_utf8_unchecked , [Production de texte à partir de données UTF-8](#)

en tant que collections génériques , [Chaînes en tant que collections génériques](#)

méthode d'insertion , [Ajout et insertion de texte](#)

insertion de texte , [Ajout et insertion de texte](#) - [Ajout et insertion de texte](#)

Méthode insert_str , Ajout [et insertion de texte](#)

itérer sur du texte , [Conventions de recherche et d'itération](#) , [Itérer sur du texte](#) - [Itérer sur du texte](#)

chaînes non Unicode , [Autres types de type chaîne](#)

analyse des valeurs de , [analyse d'autres types de chaînes](#)

méthode pop , [Suppression et remplacement de texte](#)

produire du texte à partir de données UTF-8 , [Produire du texte à partir de données UTF-8](#)

méthode push , [Ajout et insertion de texte](#)

Méthode push_str , Ajout [et insertion de texte](#)

Reporter l'allocation , [Reporter l'allocation](#) - [Reporter l'allocation](#)

méthode de suppression , [Suppression et remplacement de texte](#)

suppression et remplacement de texte , [Suppression et remplacement de texte](#)

Méthode replace_range , [Suppression et remplacement de texte](#)

recherche de texte , [Conventions de recherche et d'itération](#) - [Recherche et remplacement](#)

inspection simple , inspection [simple](#)

chaînes en mémoire , [Chaînes en mémoire](#) - [Chaînes en mémoire](#)

rognage du texte , [rognage](#)

méthode truncate , [Suppression et remplacement de texte](#)

UTF-8 et , [Caractères](#) , [Accès au texte en UTF-8](#) - [Production de texte à partir de données UTF-8](#)

Littéraux de chaîne , [Littéraux](#) de chaîne , [Chaînes d'octets](#) , [Chaîne](#)

tranche de chaîne (&str) , [Chaîne](#)

String::new , [Création de valeurs de chaîne](#)

String::with_capacity , [Création de valeurs](#) de chaîne , [Ajout et insertion de texte](#)

stringifier ! macro , [Macros intégrées](#)

chaînes et texte , [Chaînes et texte](#) - [La caisse de normalisation unicode](#)

caractères (char) , [Caractères \(char\)](#) - [Conversions vers et depuis des entiers](#)

valeurs de formatage , [Valeurs](#) de formatage - [Utilisation du langage de formatage dans votre propre code](#)

normalisation , [Normalisation](#) - [La caisse de normalisation unicode](#)
passant entre Rust et C , [Trouver des représentations de données communes](#)
expressions régulières , [Expressions régulières](#) - [Création de valeurs régulières paresseusement](#)
Arrière-plan Unicode , [Certains](#) arrière-plans Unicode - [Directionnalité du texte](#)
code non sécurisé pour la conversion d'Ascii en chaîne , [Exemple : Un type de chaîne ASCII efficace](#) - [Fonctions non sécurisées](#)

Stroustrup, Bjarne

"L'abstraction et le modèle de machine C++" , [et pourtant la rouille est toujours rapide](#)

expression de structure , [Structures de champ nommé](#)

struct patterns , [Tuple et Struct Patterns](#)

structs , [Création de champs Struct pub](#) , [Structs](#) - [Mutabilité intérieure](#)
définir des méthodes avec impl , [Définir des méthodes avec impl](#) - [Fonctions associées au type](#)

Dérivation de traits communs pour les types struct , [Dérivation de traits communs pour les types struct](#)

générique , [Structures](#) génériques - [Structures génériques](#)

implémentation de hachage , [Hachage](#)

mutabilité intérieure , [Mutabilité](#) intérieure - [Mutabilité intérieure](#)

mise en page , mise en [page de la structure](#)

avec paramètres de durée de vie , [structures génériques avec paramètres de durée de vie](#)

champ- nommé , [Structures de champ nommé](#) - Structures [de champ nommé](#)

références dans , [Structures contenant des références](#) - [Structures contenant des références](#)

tuple-like , [Tuple-Like Structs](#)

unit-like , [Unit-Like Structs](#)

sous- modules , [Modules imbriqués](#)

sous- traits , sous- [traits](#)

méthode des successeurs , [from fn et successeurs](#) - [from fn et successeurs](#)

méthode somme , [Accumulation simple : compte, somme, produit](#)

types de somme , [énumérations et modèles](#)

supertrait , Sous- [traits](#)

instruction switch , [littéraux, variables et caractères génériques dans les modèles](#)

méthode de lien symbolique , Fonctionnalités spécifiques à la [plate-forme](#) - [Fonctionnalités spécifiques à la plate-forme](#)

Type de synchronisation , [Sécurité des threads : envoi et synchronisation](#) - [Sécurité des threads : envoi et synchronisation](#) , [Caractéristiques non sécurisées](#)

objets synchronisés , [Concurrence](#)

canal synchrone, simultanéité , [fonctionnalités et performances du canal](#)

programmation synchrone à asynchrone , [De synchrone à asynchrone](#) - [Un vrai client HTTP asynchrone](#)

blocs asynchrones , [Blocs asynchrones](#) - [Création de fonctions asynchrones à partir de blocs asynchrones](#)

fonctions asynchrones , fonctions asynchrones [et expressions d'attente](#) - [Appel de fonctions asynchrones à partir de code synchrone : block on](#) , [création de fonctions asynchrones à partir de blocs asynchrones](#)

caisse de client HTTP asynchrone , [un vrai client HTTP asynchrone](#)
attendre l'expression , [les fonctions asynchrones et les expressions d'attente](#)

comparaison de conceptions asynchrones , [Comparaison de conceptions asynchrones](#)

contrats à terme (voir contrats à terme)

implémente Send , [mais votre futur implémente-t-il Send ?](#) - [Mais votre futur outil envoie-t-il ?](#)

calculs de longue durée , calculs de [longue durée : yield now et spawn blocking](#) - calculs de [longue durée : yield now et spawn blocking](#)

générant des tâches asynchrones , [Générant](#) des tâches asynchrones - [Générant](#) des tâches asynchrones , Générant des tâches asynchrones [sur un pool de threads](#)

pool de threads, génération de tâches asynchrones à partir de , génération de tâches asynchrones sur [un pool de threads](#)

erreurs de syntaxe, macros et , [Éviter les erreurs de syntaxe lors de la correspondance](#)

appels système , [De synchrone à asynchrone](#)

programmation système , [Préface](#) , [Les programmeurs système peuvent avoir de belles choses](#)

<T> , [Tuples](#)

[T] tranches , [Tranches](#)

(voir aussi tranches)

adaptateurs take et take_while , [take et take_while](#) , [by_ref](#)

tâche_local ! macro , [Création de tâches asynchrones sur un pool de threads](#)

TcpStream , [Autres types de lecteurs et d'enregistreurs](#) , [Réception de paquets : davantage de flux asynchrones](#)

modèles, macro , [Principes de base de l'extension de macro](#)

#[test] attribut , [Attributs](#)

tests , [Tests et Documentation](#) - [Doc-Tests](#)

doc-tests , [Doc-Tests](#) - [Doc-Tests](#)

tests d'intégration , [Tests d'intégration](#)

texte , [ASCII, Latin-1 et Unicode](#)

(voir aussi chaînes et texte ; UTF-8)

ajouter et insérer , [Ajouter et insérer du texte](#) - [Ajouter et insérer du texte](#)

ASCII , [ASCII, Latin-1 et Unicode](#) , [Classifier les caractères](#) , [Gérer les chiffres](#) , [Exemple : Un type de chaîne ASCII efficace](#) - [Fonctions non sécurisées](#)

conversion de casse , [Conversion de casse pour les caractères](#) , [Conversion de casse pour les chaînes](#)

conventions de recherche/itération , [Conventions de recherche et d'itération](#)

directionnalité de , directionnalité du [texte](#)

Exemple de GapBuffer , [Exemple : GapBuffer](#) - [Sécurité anti-panique dans le code non sécurisé](#)

itération sur , [Itération sur le texte](#)

supprimer et remplacer , [supprimer et remplacer du texte](#)

recherche , [Conventions de recherche et d'itération](#) - [Recherche et remplacement](#)

rognage , [rognage](#)

valeurs de texte, formatage , [Formatage des valeurs de texte](#) - [Formatage des valeurs de texte](#)

pool de threads, générant des tâches asynchrones sur , Générant des tâches asynchrones [sur un pool de threads](#)

fil

tâches asynchrones versus , [Programmation asynchrone](#)
fil d'arrière-plan , [Concurrence](#)
canaux et , [Canaux](#) - Au- [delà des pipelines](#)
impasse , [impasse](#)
gestion des erreurs dans , [Gestion des erreurs dans les threads](#)
sécurité avec envoi et synchronisation , [Thread Safety : envoi et synchronisation](#) - [Thread Safety : envoi et synchronisation](#)
partagé des données immuables entre , [Partage de données immuables entre les threads](#) - [Partage de données immuables entre les threads](#)
faire! macro , [Macros intégrées](#)
arbre à jetons , [types de fragments](#)
jetons, modèles de macro , [Principes de base de l'expansion de macro](#)
tokio crate , [Comparaison de conceptions asynchrones](#) , [Un client et un serveur asynchrones](#) , [Groupes de discussion : canaux de diffusion de tokio](#) - [Groupes de discussion : canaux de diffusion de tokio](#)
Trait ToOwned , [ToOwned](#) - [Emprunter et ToOwned au travail : la vache humble](#)
méthode to_owned , [chaîne](#)
méthode to_string , [Rechercher et remplacer](#) , [Chaîne](#)
trace_macros ! macro , [Macros de débogage](#)
objets de trait , [Objets](#) de trait - [Disposition des objets de trait](#)
code générique versus , [lequel utiliser](#) - [lequel utiliser](#)
mise en page , mise en [page de l'objet Trait](#)
références à , [références aux tranches et aux objets de trait](#)
types non calibrés et , [calibrés](#)
traits , [Gestion des arguments de la ligne de commande](#) , [Types de copie : l'exception aux mouvements](#) , [Traits et génériques](#) - [Traits comme base](#)
Const associés , Const [associés](#)
définir et mettre en œuvre , [Définir et mettre en œuvre les traits](#) - [Fonctions associées au type](#)
pour définir les relations entre les types , [Traits qui définissent les relations entre les types](#) - [Consts associés](#)
Appels de méthode entièrement qualifiés , Appels de méthode entièrement qualifiés - [Appels de méthode entièrement qualifiés](#)
impl , [impl Trait](#) - [impl Trait](#)
implémentation pour vos propres types , [Formatage de vos propres types](#) - [Formatage de vos propres types](#)

itérateurs et types associés , [Types associés \(ou fonctionnement des itérateurs\)](#) - [Types associés \(ou fonctionnement des itérateurs\)](#)
avec des macros , [Utilisation de caractéristiques avec des macros](#) - [Utilisation de caractéristiques avec des macros](#)
pour la surcharge de l'opérateur , [Caractéristiques génériques \(ou Fonctionnement de la surcharge de l'opérateur\)](#) , [Surcharge de l'opérateur](#)
types d'autres personnes et , [Caractéristiques et types d'autres personnes](#) - [Caractéristiques et types d'autres personnes](#)
limites de la rétro-ingénierie , [Limites de la rétro-ingénierie](#) - Limites [de la rétro-ingénierie](#)
Soi en tant que type , [Soi en traits](#) - [Soi en traits](#)
pour les types struct , [Dérivation des traits communs pour les types struct](#)
[struct](#)
sous- traits , sous- [traits](#)
fonctions associées au type , Fonctions associées au [type](#)
dangereux , [Traits](#) dangereux - [Traits dangereux](#)
utilité (voir traits d'utilité)
dépendances transitives , [Crates](#)
Travis CI , [plus de belles choses](#)
arbres , [Propriété](#)
rognage du texte de la chaîne , [rognage](#)
essayer! macro , [propagation des erreurs](#)
Trait TryFrom , [TryFrom et TryInto](#)
Trait TryInto , [TryFrom et TryInto](#)
méthodes try_fold et try_rfold , [try_fold et try_rfold](#) - [try_fold et try_rfold](#)
méthode try_for_each , [for_each et try_for_each](#)
modèles de tuple , modèles de [tuple](#) et de [struct](#)
structures de type tuple , [Structures de type tuple](#)
tuples , [Tuples](#) - [Tuples](#)
alias de type , alias de type , alias de [type de résultat](#) , [utilisation des déclarations pub](#)
alignement des caractères, pointeurs bruts et , [tailles et alignements des caractères](#)
inférence de type , [Types fondamentaux](#)
paramètres de type , [analyse d'arguments de ligne de commande de paires](#) , structures génériques , [fonctions génériques et paramètres de type](#) , [comparaisons d'équivalence](#)

taille des caractères, pointeurs bruts et , [tailles et alignements des caractères](#)

fonctions associées au type , Fonctions associées au type , [Fonctions associées au type](#)

types , [Types fondamentaux](#) - Au- [delà des bases](#)

tableaux , [tableaux](#)

associés , [Types associés \(ou Fonctionnement des itérateurs\)](#) - [Types associés \(ou Fonctionnement des itérateurs\)](#)

moulages et , [Type Moulages](#)

des fermetures et des fonctions , Types de fonctions et de fermetures - [Types de fonctions et de fermetures](#)

gestion des erreurs , [Résultat](#) - [Pourquoi des résultats ?](#) , [Formatage d'autres types](#) , Types [d'erreur et de résultat](#)

nom de fichier , [OsStr et chemin](#)

virgule flottante , Types à virgule [flottante](#) - [Types](#) à virgule flottante , [max, min](#) , [Formatage des nombres](#)

valeurs de mise en forme , [Mise en forme d'autres types](#) , [Mise en forme de vos propres types](#) - [Mise en forme de vos propres types](#)

implémenter vos propres itérateurs , [Implémenter vos propres itérateurs](#) - [Implémenter vos propres itérateurs](#)

Implémentation IntoIterator , [Implémentations](#) IntoIterator - [Implémentations IntoIterator](#)

numeric , [Types numériques à largeur fixe](#) - Types à virgule [flottante](#) , [Formatage des nombres](#)

surcharge de l'opérateur et , [surcharge de l'opérateur](#)

Paramètres , [Structures](#) génériques , [Fonctions génériques et paramètres de type](#) , [Comparaisons d'équivalence](#)

pointeurs (voir types de pointeurs)

pour représenter une séquence de valeurs en mémoire , des [tableaux, des vecteurs et des tranches](#) - [tranches](#)

Dimensionné , [Dimensionné](#) - [Dimensionné](#)

tranches , [tranches](#)

String et str (voir Types String et str)

traits pour ajouter des méthodes à , [Traits et Types d'autres personnes](#)

traits pour définir les relations entre , [Traits qui définissent les relations entre les types](#) - [Consts associés](#)

tuples , [Tuples](#) - [Tuples](#)

non calibré , [calibré](#) - [calibré](#)

défini par l'utilisateur , [Making Struct Fields pub](#)
vecteurs , [Vecteurs](#) - [Vecteurs](#)

tu

opérateurs unaires , [opérateurs unaires](#)
gammes illimitées , [possibilités multiples assorties](#)
comportement indéfini , les [programmeurs systèmes peuvent avoir de](#)
[belles choses](#) , [dangereux de quoi ?](#) , [Comportement indéfini](#) - [Compor-](#)
[tement indéfini](#)

Unicode , [Certains arrière](#) -plans Unicode - [Directionnalité du texte](#)
ASCII et , [ASCII, Latin-1 et Unicode](#)
caractères littéraux , [Caractères](#)
Latin-1 et , [ASCII, Latin-1 et Unicode](#)
normalisation , [Normalisation](#) - [La caisse de normalisation unicode](#)
OsStr et , [OsStr et Chemin](#)
directionnalité du texte , [Directionnalité du texte](#)
UTF-8 , [UTF-8](#) - [UTF-8](#)

caisse de normalisation unicode , [La caisse de normalisation unicode](#) -
[La caisse de normalisation unicode](#)
non implémenté ! macro , [Macros intégrées](#)
unions , [Enums and Patterns](#) , [Unsafe Blocks](#) , [Réinterpréter la mé-](#)
[moire avec les unions](#) - [Emprunter des unions](#)
tests unitaires , [écriture et exécution de tests unitaires](#)
type d'unité , [Tuples](#)
structures de type unité , [Structures de type unité](#)

Unix

fichiers et répertoires , Fonctionnalités spécifiques à la [plate-forme](#)
- [Fonctionnalités spécifiques à la plate-forme](#)
canaux , [Canaux](#) - [Envoi de valeurs](#)

Trait de marqueur de détachement , [Le](#) trait de détachement - [Le trait](#)
[de détachement](#)

Détacher le trait , [Envoi de paquets](#)

blocs non sécurisés , [pointeurs bruts](#) , [code](#) non sécurisé , [blocs non sé-](#)
[curisés](#) - [Exemple : un type de chaîne ASCII efficace](#) , un [bloc non sécu-](#)
[risé ou une fonction non sécurisée ?](#)

code dangereux , [Code dangereux](#) - [Syndicats d'emprunt](#)

fonctions étrangères (voir fonctions étrangères)

interface brute libgit2 , [Une interface brute vers libgit2](#) - [Une inter-](#)
[face brute vers libgit2](#)

pointeurs bruts (voir pointeurs bruts)
comportement indéfini , [Comportement](#) indéfini - [Comportement indéfini](#)
syndicats , [Réinterpréter la mémoire avec les syndicats](#) - [Emprunter des syndicats](#)
blocs non sécurisés , [pointeurs bruts](#) , [code](#) non sécurisé , [blocs non sécurisés](#) - [Exemple : un type de chaîne ASCII efficace](#) , un [bloc non sécurisé ou une fonction non sécurisée ?](#)
fonctionnalité non sécurisée , [Dangereux de quoi ?](#) - [Pas à l'abri de quoi ?](#)
fonctions non sécurisées , [Code](#) non sécurisé , [Fonctions](#) non sécurisées - Fonctions [non sécurisées](#)
traits dangereux , [Traits](#) dangereux - [Traits dangereux](#)
fonctions non sécurisées , [Code](#) non sécurisé , [Fonctions](#) non sécurisées - Fonctions [non sécurisées](#)
traits dangereux , [Traits](#) dangereux - [Traits dangereux](#)
types entiers non signés , [Types entiers](#)
types non calibrés , [calibrés](#) - [calibrés](#)
dérouler , [dérouler](#) - [dérouler](#)
méthode unwrap , [écriture de fichiers image](#) , gestion des [erreurs dans les threads](#)
use declarations , [Paths and Imports](#) , [Making use Declarations pub](#)
types définis par l'utilisateur , [Making Struct Fields pub](#)
usize type , [Types entiers](#) , [Accès aux éléments](#)
UTF-8 , [UTF-8](#) - [UTF-8](#)
accéder au texte en tant que , [Accéder au texte en tant qu'UTF-8](#)
Méthodes ASCII avec , [Caractères de classification](#)
type de caractère et , [Caractères](#)
OsStr et , [OsStr et Chemin](#) - [OsStr et Chemin](#)
produire du texte à partir de données , [Produire du texte à partir de données UTF-8](#)
Gestion des chaînes et des chaînes , [Chaîne et chaîne](#)
chaînes en mémoire , [chaînes en mémoire](#)
unsafe code and , [Exemple : Un type de chaîne ASCII efficace](#) - [Exemple : Un type de chaîne ASCII efficace](#)
Traits d'utilité , [Traits](#) d'utilité - [Emprunter et posséder au travail : la vache humble](#)
AsRef et AsMut , [AsRef et AsMut](#)

Emprunter et EmprunterMut , [Emprunter et EmprunterMut](#) - [Emprunter et EmprunterMut](#)
Cloner , [cloner](#) , [copier et cloner pour les fermetures](#) , [accéder aux éléments](#)
Copier , [copier](#) , [copier et cloner pour les fermetures](#)
Vache , [Emprunter et Propriétaire au Travail : La Vache Humble](#) ,
[Différer l'Allocation](#) - [Différer l'Allocation](#)
Par défaut , [Par défaut](#) - [Par défaut](#) , [partition](#)
Deref et DerefMut , [Deref et DerefMut](#) - [Deref et DerefMut](#)
Goutte , [Goutte](#) - [Goutte](#)
De et vers , [De et vers](#) - [De et vers](#) , [Utilisation de caractéristiques avec des macros](#)
Dimensionné , [Dimensionné](#) - [Dimensionné](#)
ToOwned , [ToOwned](#) - [Emprunter et ToOwned au travail : la vache humble](#)
TryFrom et TryInto , [TryFrom et TryInto](#)
module utils , [Types d'erreur et de résultat](#)

V

Type VacantEntry, HashMap et BTreeMap , [Entrées](#)
valeurs
construire à la demande , [construire des valeurs régulières paresseusement](#)
chute , [propriété](#) , [propriété](#) , [fermetures qui tuent](#) - [FnOnce](#)
formatage , [Valeurs](#) de formatage - [Utilisation du langage de formatage dans votre propre code](#) , [Pointeurs bruts](#)
types fondamentaux pour représenter , [Types fondamentaux](#) - [Au-delà des bases](#)
se déplace , [plus d'opérations qui se déplacent](#)
en passant par , [Références aux valeurs](#) , [Implémentations IntoIterator](#) , [Collections](#)
recevoir via les canaux , [recevoir des valeurs](#)
références et , [Références aux valeurs](#) - [Références aux tranches et aux objets de trait](#)
envoi via les canaux , [Envoi de valeurs](#) - [Envoi de valeurs](#)
ensembles et différences de valeurs "égales" , [lorsque des valeurs égales sont différentes](#)
statique , [Recevoir des références en tant qu'arguments de fonction](#) , [Modules](#) , [Statique et Constantes](#)

Types String et str , [Analyse d'autres types à partir de chaînes](#) -

[Conversion d'autres types en chaînes](#)

capture de variables , [Capture de variables](#)

variables

affectation à , [Plus d'opérations qui bougent](#)

emprunt local , [Emprunter une variable locale](#) - [Emprunter une variable locale](#)

condition , [Variables de condition \(Condvar\)](#)

déclaration à partir de bibliothèques étrangères , [Déclaration de fonctions et de variables](#) étrangères - [Déclaration de fonctions et de variables étrangères](#)

global , [Variables globales](#) - [Variables globales](#)

propriété , [Gestion des arguments de la ligne de commande](#) , [Propriété et mouvements](#) - [Rc et Arc : Propriété partagée](#) , [Emprunter et ToOwned at Work : The Humble Cow](#)

dans les modèles , les [littéraux, les variables et les caractères génériques dans les modèles](#)

statique , [variables globales](#)

Type Vec , [Gestion des arguments de ligne de commande](#)

append method , [Croissance et rétrécissement des vecteurs](#)

bâtiment de VecDeque , [VecDeque<T>](#)

méthode de la capacité , [vecteurs croissants et rétrécissants](#)

méthode claire , [vecteurs croissants et rétrécissants](#)

méthode de déduplication , [vecteurs croissants et rétrécissants](#)

méthode dedup_by , [vecteurs croissants et rétrécissants](#)

méthode dedup_by_key , [vecteurs croissants et rétrécissants](#)

méthode de drainage , [vecteurs croissants et rétrécissants](#)

méthode d'extension , [vecteurs croissants et rétrécissants](#)

méthode d'insertion , [vecteurs croissants et rétrécissants](#)

méthode pop , [vecteurs croissants et rétrécissants](#)

méthode push , [vecteurs croissants et rétrécissants](#)

supprimer la méthode , [Croissance et rétrécissement des vecteurs](#)

méthode de réserve , [vecteurs croissants et rétrécissants](#)

méthode reserve_exact , [vecteurs croissants et rétrécissants](#)

méthode de redimensionnement , [vecteurs croissants et rétrécissants](#)

méthode resize_with , [vecteurs croissants et rétrécissants](#)

méthode de conservation , [vecteurs croissants et rétrécissants](#)

méthode shrink_to_fit , [vecteurs croissants et rétrécissants](#)

méthode split_off , [vecteurs croissants et rétrécissants](#)
Méthode swap_remove , [Permutation](#)
méthode tronquée , [vecteurs croissants et rétrécissants](#)
méthode with_capacity , [vecteurs croissants et rétrécissants](#)
vec! macro , [Vecteurs](#) , [Vec<T>](#) , [Répétition](#) - [Répétition](#)
Type de collection Vec<T> , [Tableaux, vecteurs et tranches](#) , [Vecteurs](#) ,
[Vue](#) d'ensemble , [Vec<T>](#) - [Rust exclut les erreurs d'invalidation](#)
accéder aux éléments , [Accéder aux éléments](#) - [Accéder aux éléments](#)
comparaison de tranches , [Comparaison de tranches](#)
vecteurs de croissance/rétrécissement , Vecteurs de croissance et de rétrécissement - [Vecteurs de croissance et de rétrécissement](#)
erreurs d'invalidation, exclusion , [Rust exclut les erreurs d'invalidation](#)
itération , [itération](#)
rejoindre , [joindre](#)
éléments aléatoires , [Éléments aléatoires](#)
recherche , [tri et recherche](#)
tri , [tri et recherche](#)
Fractionnement , [Fractionnement](#) - [Fractionnement](#)
échange , [échange](#)
Vecu8 , [Autres types de lecteurs et d'enregistreurs](#)
VecDeque , [Aperçu](#) , [VecDeque<T>](#) - [VecDeque<T>](#)
méthode retour , [VecDeque<T>](#)
méthode back_mut , [VecDeque<T>](#)
méthode avant , [VecDeque<T>](#)
méthode front_mut , [VecDeque<T>](#)
méthode make_contiguous , [VecDeque<T>](#)
méthode pop_back , [VecDeque<T>](#)
méthode pop_front , [VecDeque<T>](#)
méthode push_back , [VecDeque<T>](#)
méthode push_front , [VecDeque<T>](#)
VecDeque::from(vec) , [VecDeque<T>](#)
vecteur de Ts , [tableaux, vecteurs et tranches](#)
vecteurs , [vecteurs](#) - [vecteurs](#) , vecteurs [croissants et rétrécissants](#) -
vecteurs [croissants et rétrécissants](#)
versions, fichier , [Spécification des dépendances](#)
barre verticale (|) , [Correspondance de plusieurs possibilités](#)
table virtuelle (vtable) , [mise en page de l'objet Trait](#)

waker , [Futures](#) , [Primitive Futures et Executors : Quand un futur mérite-t-il à nouveau d'être interrogé ?](#) - [Implémentation de block on](#)
pointeurs faibles , [Rc et Arc : propriété partagée](#)
serveur Web, création avec Rust , [Servir des pages sur le Web](#) - [Servir des pages sur le Web](#)
programme bien conduit, Règles de Rust pour , [Comportement indéfini](#)
boucle while let , [Boucles](#)
boucle while , [Fonctions Rust](#) , [Boucles](#) , [Flux de contrôle dans les boucles](#)
opérations sur un ensemble [entier](#) , opérations sur un ensemble entier
caractères génériques , [chemins et importations](#) , [versions](#) , [littéraux](#) ,
[variables et caractères génériques dans les modèles](#)
les fenêtres
fichiers et répertoires , [Fonctionnalités spécifiques à la plate-forme](#)
OsStr et , [OsStr et Chemin](#)
Paquet de rouille pour , [rustup et Cargo](#)
Utilisation des fonctions des bibliothèques , [Utilisation des fonctions des bibliothèques](#)
valeur de travail , [rayonne](#)
pools de nœuds de calcul , [simultanéité](#)
espaces de travail , Espaces de [travail](#)
opérations d'habillage , [Arithmétique vérifiée, encapsulée, saturée et débordante](#) , [Arithmétique vérifiée](#) , [encapsulée, saturée et débordante](#)
méthode wrap_offset , [pointeurs bruts](#) , [déréférencement des pointeurs bruts en toute sécurité](#)
fonction d'écriture , [lecture et écriture de fichiers](#)
méthode d'écriture , [Caractéristiques et génériques](#) , [Méthodes par défaut](#)
Caractéristique d' écriture , [entrée et sortie](#)
ajouter et insérer du texte , [Ajouter et insérer du texte](#)
méthode flush , [écrivains](#)
utiliser le langage de formatage dans votre code , [Utiliser le langage de formatage dans votre propre code](#)
méthode d'écriture , [écrivains](#)
méthode write_all , [écrivains](#)
écrivez! macro , [Opérateurs binaires](#) , [Ajout et insertion de texte](#) , [Valeurs de formatage](#) , [Rédacteurs](#)

Trait WriteBytesExt , [données binaires, compression et sérialisation](#)
écrits ! macro , [Erreurs d'impression](#) , [Ajout et insertion de texte](#) , [Valeurs de formatage](#) , [Rédacteurs](#)

écrivains , [lecteurs et écrivains](#) , [écrivains - écrivains](#)

données binaires, compression, sérialisation , [Données binaires, compression et sérialisation](#) - [Données binaires, compression et sérialisation](#)

fichiers , [Fichiers](#)

autres types , [Autres types de lecteurs et d'enregistreurs](#) - [Autres types de lecteurs et d'enregistreurs](#)

Chercher le trait , [Chercher](#)

Fonction write_image , [Écriture de fichiers image](#)

write_unaligned , Entrer [et sortir de la mémoire](#)

write_volatile , Entrer [et sortir de la mémoire](#)

Oui

yield_now , Calculs de [longue durée : yield now et spawn blocking](#) -

Calculs de [longue durée : yield now et spawn blocking](#)

Z

principe de zéro frais généraux , [et pourtant la rouille est toujours rapide](#)

zéro-uplet , [Tuples](#)

Trait pouvant être mis à zéro , [Traits dangereux](#)

adaptateur zip , [zip](#)

[Soutien](#) [Se déconnecter](#)