Rust: Sudoku Time!

Je kunt rust zelf installeren (zonder admin!!!) en de voorbeelden volgen :-)

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Inhoud

- Wat is Rust?
- Sudoku, welke regels gebruik ik...
- Data opslag
- Algorithms
- Doe mee!

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Wat is Rust?

Net zo snel als C/C++,…

Pff, dat zijn de lastigste om op te sporen

“Rust is een systeemprogrammeertaal die razendsnel draait, segmentatie fouten voorkomt en korrekt gescheiden processen garandeerd.”

Fijn, want zelfs mijn telefoon heeft 4 cores

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Mutatie en initialisatie (src/lib.rs)

definitie en initialisatie

```
pub fn count_equal_options(a: &Sudoku,b: &Sudoku) -> usize {
        let mut count = 0;
        for ri in 0..DIMENSIONPWR2 {
                for ci in 0..DIMENSIONPWR2 {
                        for oi in 0..DIMENSIONPWR2 {
                                if a.data[ri][ci][oi] && b.data[ri][ci][oi] {
                                        count = count+1;
                                }
                        }
                }
        }
        count
}
```

Ga na de directory:

$ cd sudoku

Compileren doe je met dit commando:

$ cargo run

Wat gebeurt er als je “**mut**” of “**=0**” weglaat?

curl https://sh.rustup.rs -sSf | sh

git clone https://github.com/peter-scholtens/sudoku.git

Controle bereik van index (src/lib.rs)

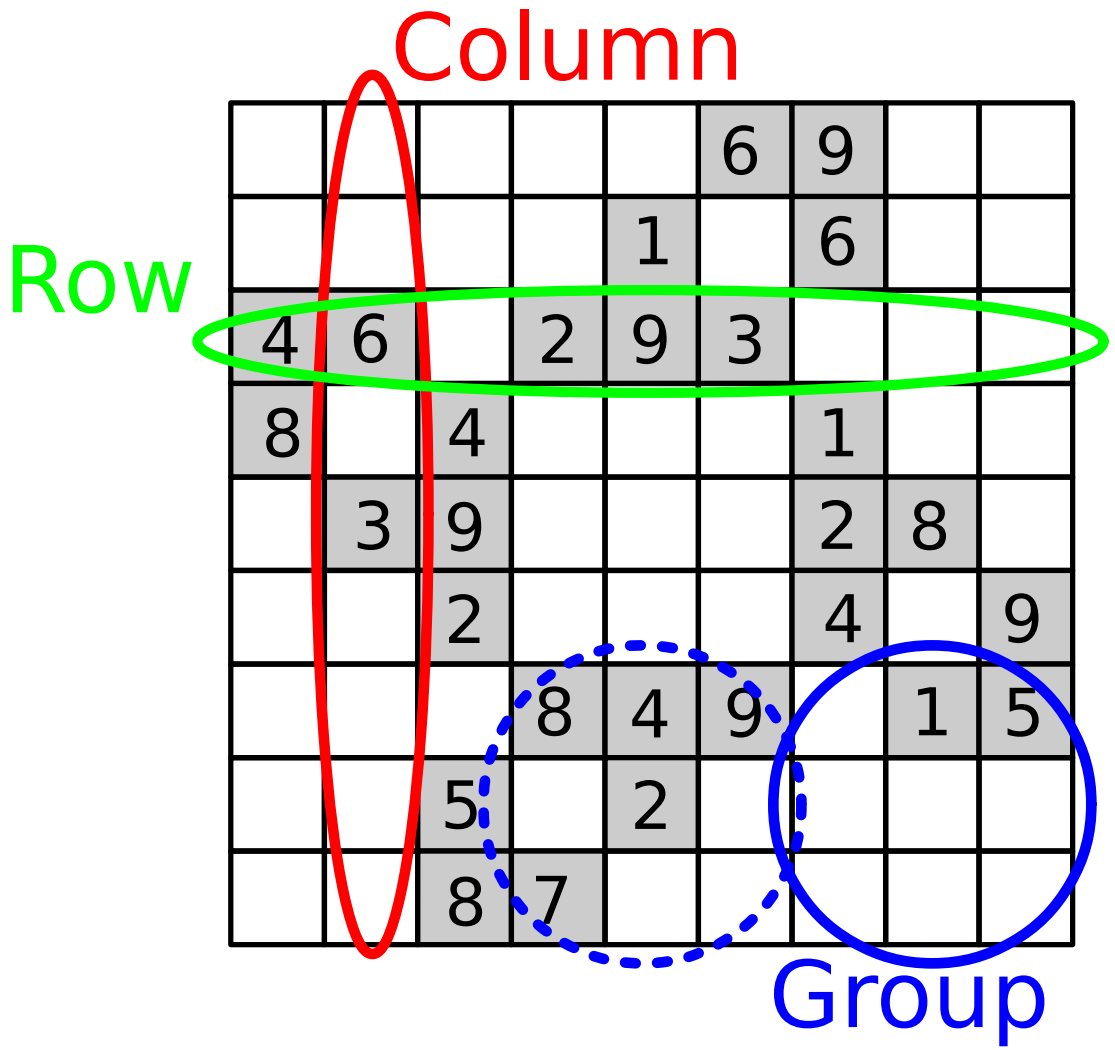
Tel er eens 1 bij op…

```
pub const  DIMENSION: usize = 3;
pub const  DIMENSIONPWR2: usize = DIMENSION*DIMENSION;

...

pub fn set(&mut self, r:usize, c:usize, v:usize) {

        for vi in 0..DIMENSIONPWR2 {

                self.data[r][c][vi]=false;

        }

        self.data[r][c][v]=true;
}
```

Compileer nu met dit commando:
$ RUST_BACKTRACE=1 cargo run
Wat gebeurt er als je bij “**DIMENSIONPWR2**” 1 optelt?

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Sudoku Puzzel Regels

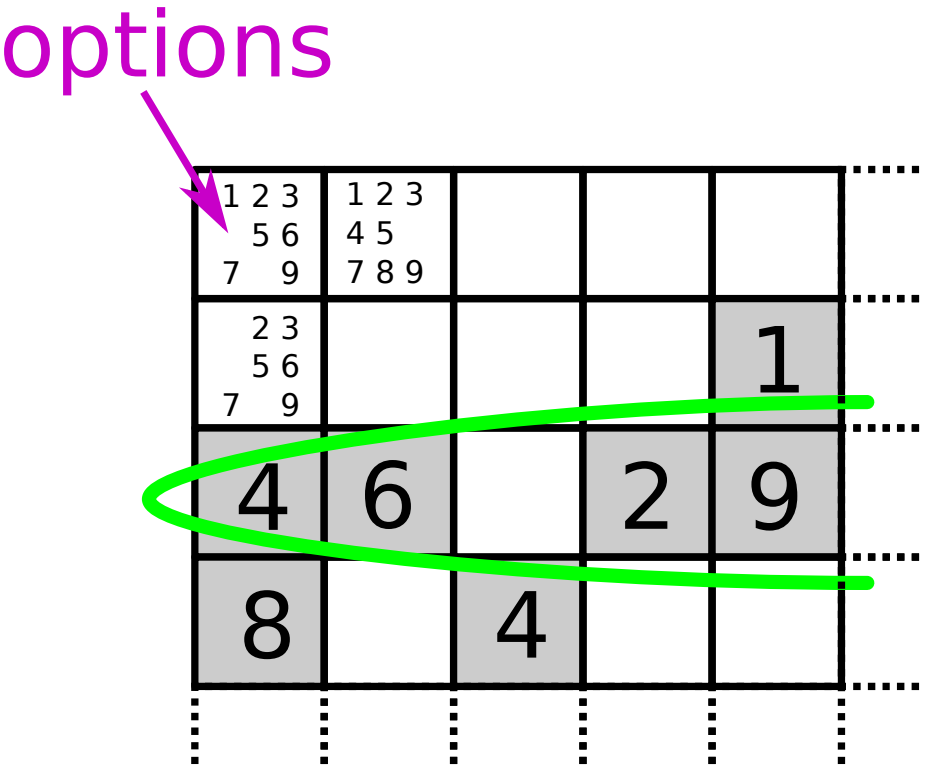


“Plaats elk getal van 1 tot 9 eenmaal, in elke rij, kolom en groep”

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Eerste data formaat, handig?

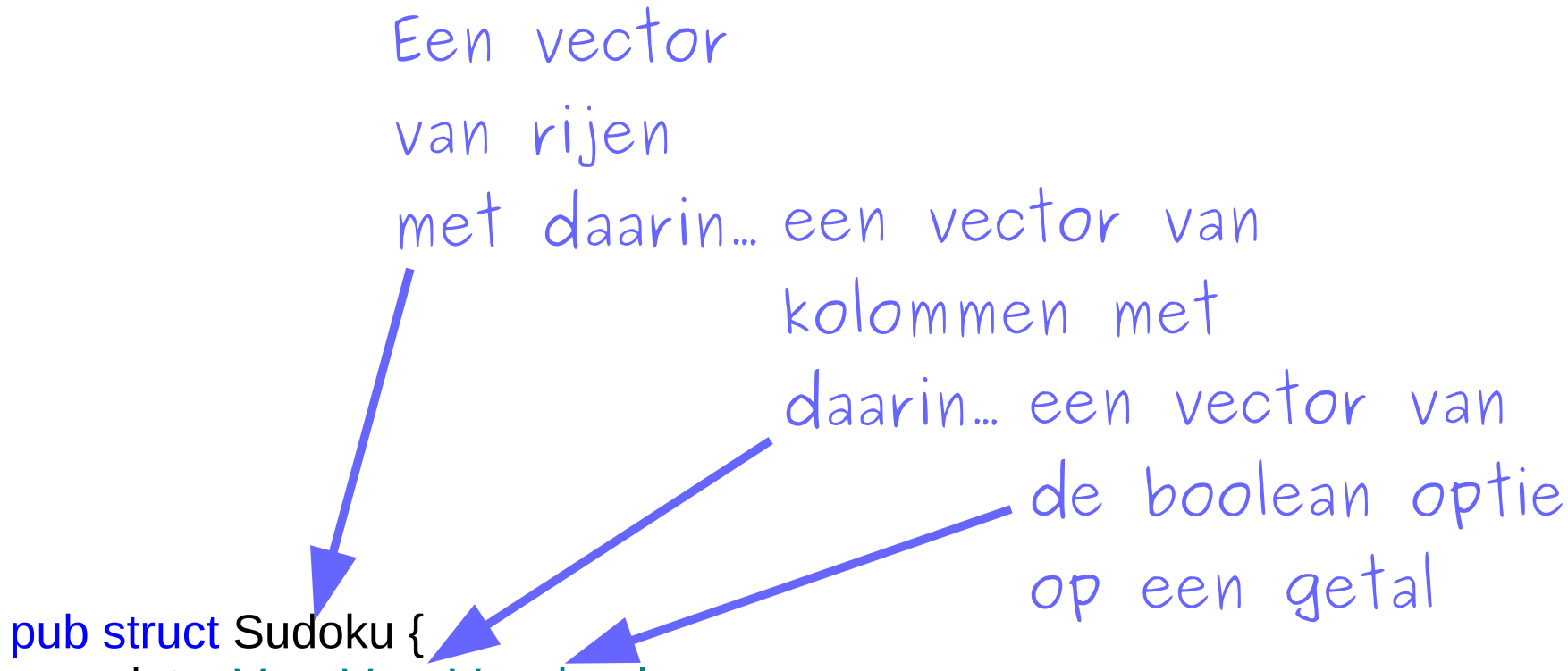
```
pub struct SudokuField {
        data: Vec<Vec<usize>>,
}

pub struct SudokuProblem {
        data: Vec<Vec<Option<usize>>>,
}

pub struct SudokuCell {
        options: Vec<bool>,
}

struct SudokuRow {
        column: Vec<SudokuCell>,
}

pub struct SudokuSolution {
        row: Vec<SudokuRow>,
}
```



Even terug in de tijd met git:
$ git checkout 17a991c5

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Alle data als 3D matrix...




```
pub struct Sudoku {
        data: Vec<Vec<Vec<bool>>>,
}
```

Dit data formaat sinds:
$ git checkout d528c7c2

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Uniforme data, minder code

- Minder:
  - Functies: van 24 naar 17
  - Code: 254-, 131+ regels
- Gemaakte fouten:
  - Out-of-range index
  - Formatted print output
- Nooit segmentation faults!

In slechts twee avonden :-)

$ git diff 17a991c5..d528c7c2 | awk /^+/ | wc

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Algoritme 1

1) Kies een willekeurige *symboolwaarde* voor willekeurige *cel* met opties, bijvoorbeeld “rij 3, kolom 1 wordt een 4”

2) Streep de opties bij de andere cellen in dezelfde rij, kolom of groep weg.

3) Zijn er nog steeds meerdere opties, ga naar 1.

Werkt niet :-(

Je hebts soms meer 1'en (of 2'en) dan de 9 in elke rij/kolom/groep

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Algoritme 2

1) Voor alle symboolwaarden 1..9 doe:
2) Herhaal 9 keer:
   a) Kies een willekeurige cel met opties, en kies de symboolwaarde
   b) Streep de opties (van dezelfde symbolwaarde) bij de andere cellen in dezelfde rij, kolom of groep weg.
   c) Zijn er geen opties meer, ga naar 3.
3) Einde

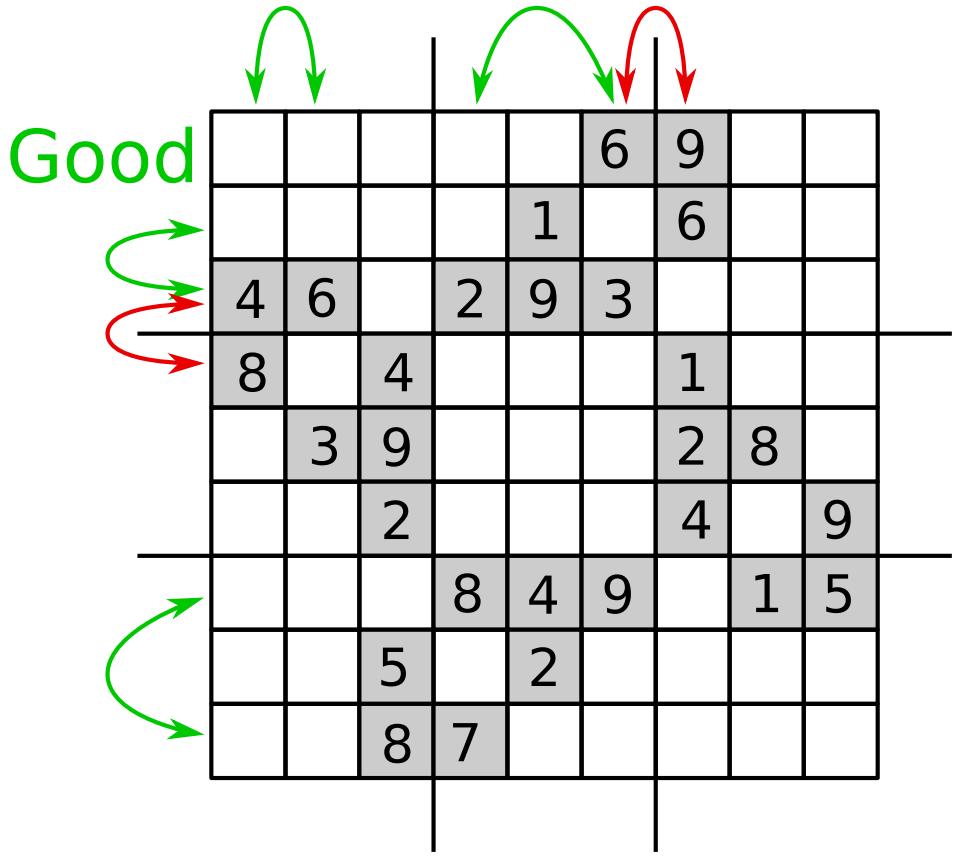
Werkt ook niet :-(

Na een paar iteraties hebben sommige cellen helemaal geen opties noch symbool…

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Algoritme 3

1) Begin altijd met dezelfde geldige oplossing (identity)
2) Verwissel willekeurig rijen of kolommen binnen grenzen van 3
3) Herhaal 99 keer



He, dit gaat goed!

Maar erg simpele sudoku's, 50 van 81 cellen gevuld.

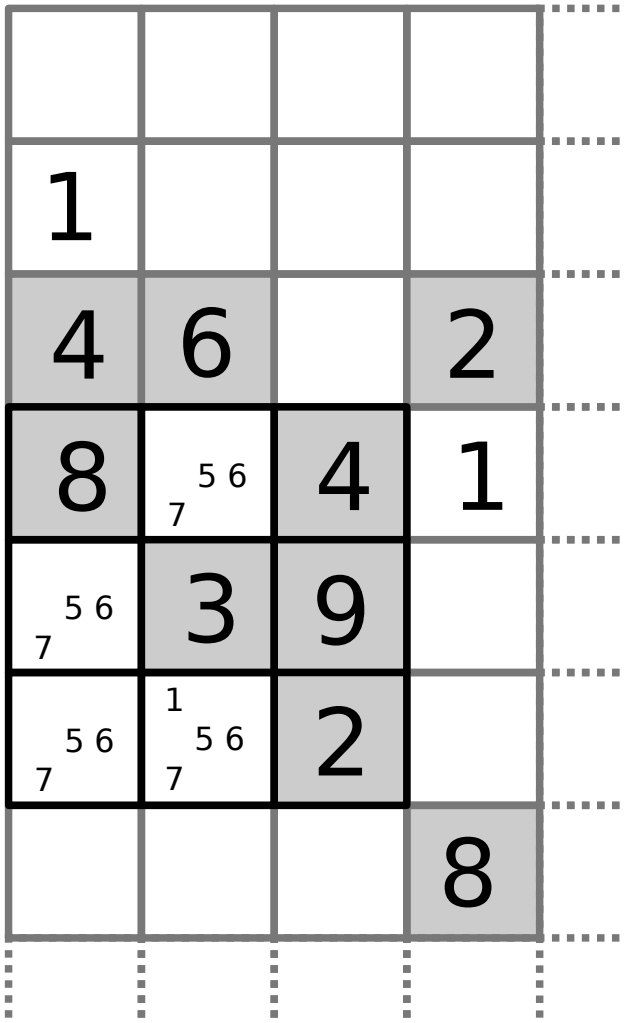
$ git checkout 8e29709a

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Algoritme 3B

Zelfde als [3], echter met reductie door uitputting:

Als symbool optie maar een keer voorkomt in rij, kolom of groep, “kies” die optie en reduceer verder.



En dit gaat beter!

Ongeveer 30 van 81 cellen gevuld.

$ git checkout 18c49474

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Cargo TOML

Tom's Obvious, Minimal Language

```
[package]
name = "sudoku"
version = "0.1.0"
authors = ["Peter C. S. Scholtens <peter.scholtens@xs4all.nl>"]

[dependencies]
rand = "0.3"
```

Dit is wat je vraagt en opgeeft (.toml)

$ cargo update

Updating libc v0.2.20 -> v0.2.21

Dit is wat je krijgt! (.lock)

```
[package]
name = "sudoku"
version = "0.1.0"
authors = ["Peter C. S. Scholtens <peter.scholtens@xs4all.nl>"]

[dependencies]
rand = "0.3"

[root]
name = "sudoku"
version = "0.1.0"
dependencies = [
 "rand 0.3.15 (registry+https://github.com/rust-lang/crates.io-index)",
]

[[package]]
name = "libc"
version = "0.2.20"
source = "registry+https://github.com/rust-lang/crates.io-index"

[[package]]
name = "rand"
version = "0.3.15"
source = "registry+https://github.com/rust-lang/crates.io-index"
dependencies = [
 "libc 0.2.20 (registry+https://github.com/rust-lang/crates.io-index)",
]
```

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Cargo test

```
// Verifies if every number is only seen at most once

// in each row, column and group.
pub fn has_only_unique_options(&self) -> bool {...}
```

Functie die ik wil testen uit ./src/lib.rs

```
#[test]
fn test_new_with_correct_data() {
        let correct = sudoku::Sudoku::identity_sudoku();
        assert_eq!(true, correct.has_only_unique_options());
}

#[test]
fn test_new_with_fault_data_rows() {
        let faulty = sudoku::Sudoku::new_with_data(vec![
        vec![1,1,1,1,1,1,1,1,1],
        …
        vec![9,9,9,9,9,9,9,9,9],
        ]);
        assert_eq!(false, faulty.has_only_unique_options());
}
```

Tests met correcte en foutieve data in ./tests/field.rs

$ cargo test

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Rust up

```
rustup show
```
Laat geïnstalleerde toolchain zien

```
rustup install beta
rustup install nightly
```
(Nodig voor benches)

```
rustup target list
```
ARMv7, MIPS, S390, PPC…
BSD, Linux, iOS, Darwin, Windows

```
rustup default nightly
```

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Cargo bench

```
#[bench]
fn bench_shake_randomly(b: &mut Bencher) {
        let mut field  = Sudoku::new(false);
        b.iter(|| field.shake_randomly() );
}

#[bench]
fn bench_shake_till_reached_var(b: &mut Bencher) {
        let     p  = Sudoku::identity_sudoku();
        let mut n  = Sudoku::identity_sudoku();
        b.iter(|| {
                while count_equal_options(&p,&n) > 1 {
                        n.shake_randomly();
                }
        } );
}
```

Meten is weten!

```
running 2 tests
test tests::bench_shake_randomly          .. bench: 46,940ns/iter (+/-2,236)
test tests::bench_shake_till_reached_var .. bench:    791ns/iter (+/-20)
[i7-4790@3.6GHz]
```

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Conclusies

- Sudoku: begonnen met te simpele aannames,...
  - Algoritmes ontwikkeld tijdens programmeren, dus geen/minder modellering nodig (Octave/Matlab)
- Rust: strong type checking
  - Verandering van data opslag gedurende ontwerptraject is geen probleem!
- Rustup: complete toolchain out-of-the box!
- Cargo voor compileren, testen en benchmarken
  - Voorkomt herhaling van fouten, als je tenminste genoeg regressietesten schrijft

curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git

Friends of Rust

(Organizations running Rust in production)

https://www.rust-lang.org/en-US/friends.html

ETHCORE

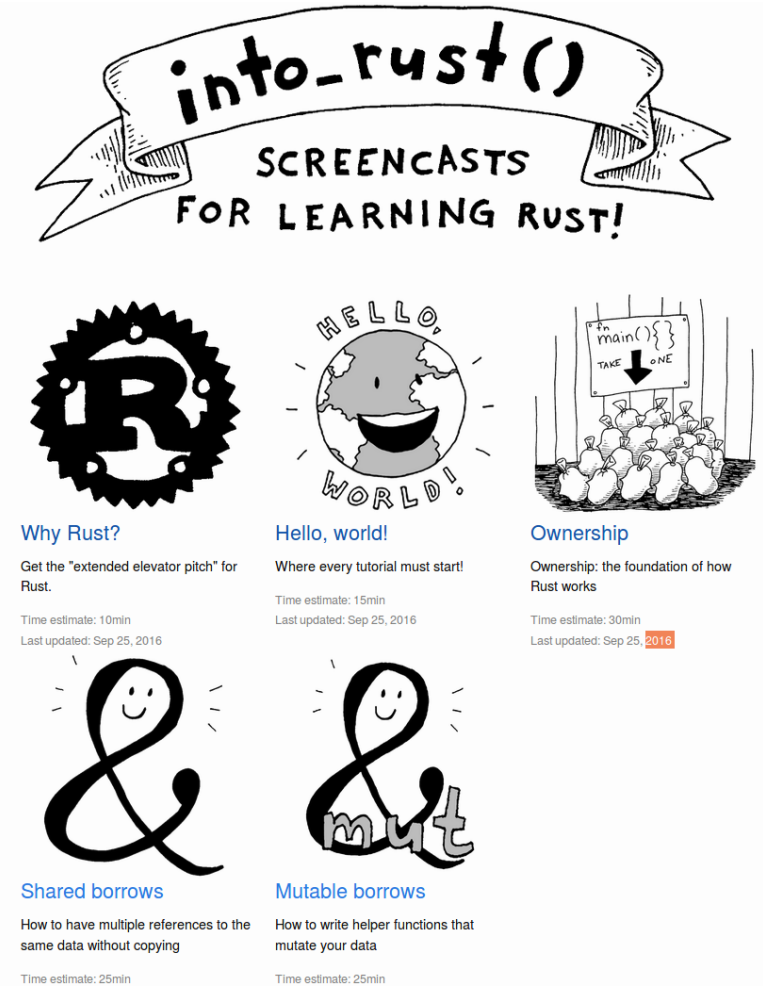
CodePicnic : We provide Rust sandboxes people can play around with, as well as fork to expand examples and apps they want to.

curl https://sh.rustup.rs -sSf | sh

git clone https://github.com/peter-scholtens/sudoku.git

Kijk hier verder...

- Rust:
  - Video tutorials:
    - http://intorust.com
    - https://air.mozilla.org/introduction-to-rust
  - Online uitproberen kan hier:
    - http://rustbyexample.com
    - https://play.rust-lang.org
    - https://www.rust-lang.org
  - Community in Nederland:
    - https://www.meetup.com/Rust-Amsterdam
    - https://www.meetup.com/Rust-Utrecht
- Sudoku's:
  - https://wiki.gnome.org/Apps/Sudoku



curl https://sh.rustup.rs -sSf | sh
git clone https://github.com/peter-scholtens/sudoku.git