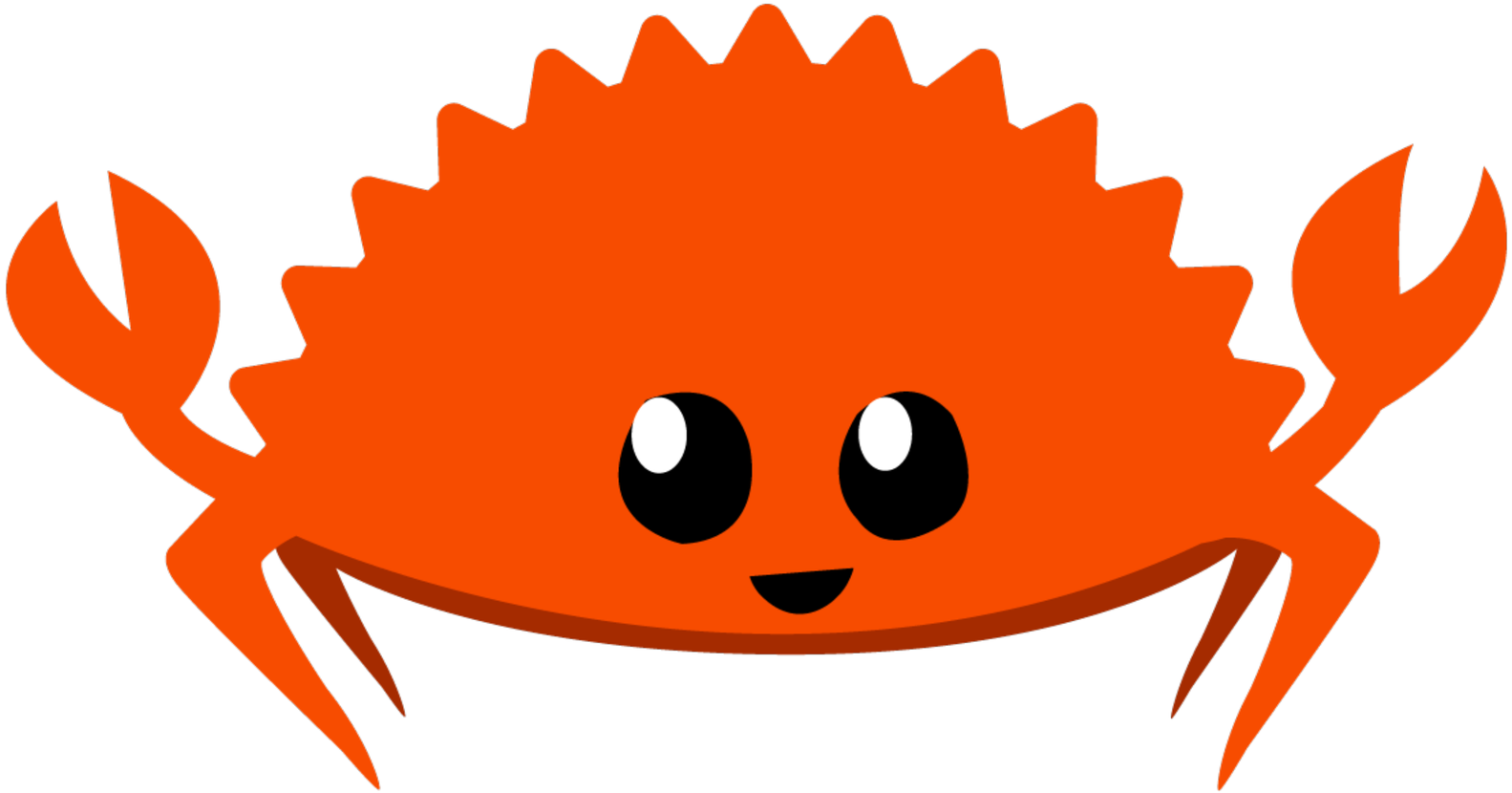


Rust: Просто о сложном

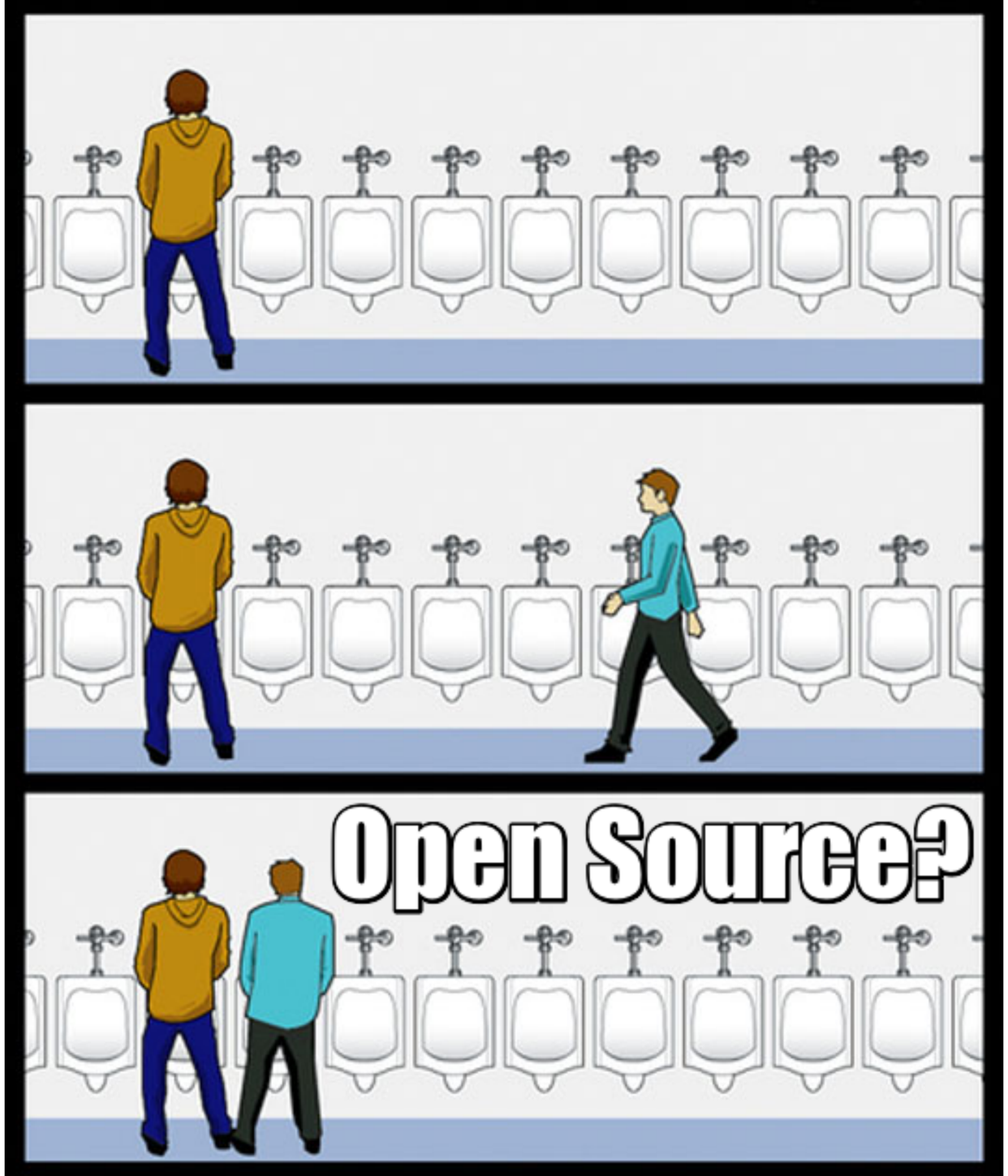
mgrachev

mgrachev

BACKEND • TEAMLEAD • DEVOPS

Грачев Михаил

/evrone.



Я ❤️ Open Source

goreleaser/goreleaser

reviewdog/reviewdog

Я ❤️ Open Source

dotenv-linter/dotenv-linter

Мой путь

☹️ Object Pascal

😣 Delphi

😖 Visual Basic

😩 PHP

🙂 Ruby

🙃 Go

🤩 Rust

Google

rust

Все | Видео | Картинки | Новости | Карты | Ещё | Настройки | Инструменты

Результатов: примерно 436 000 000 (0,41 сек.)

store.steampowered.com › app › Rust ▾ Перевести эту страницу

Rust on Steam

★★★★½ Рейтинг: 9/10 - 308 061 отзыв - 39,99 $

8 февр. 2018 г. - The only aim in **Rust** is to survive - Overcome struggles such as hunger, thirst and cold. Build a fire. Build a shelter. Kill animals. Protect yourself ...

Rust Item Store · Rust on Steam · Hurtworld · Fortify

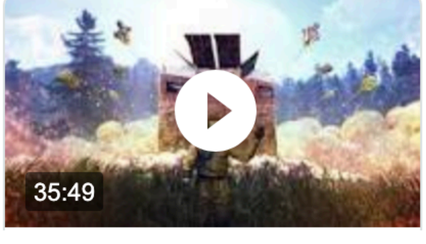
rust.facepunch.com ▾ Перевести эту страницу

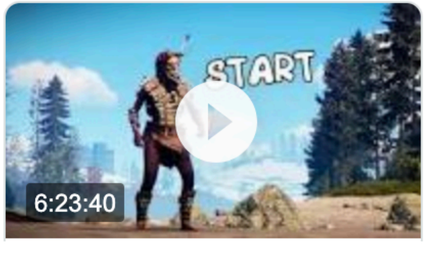
Rust

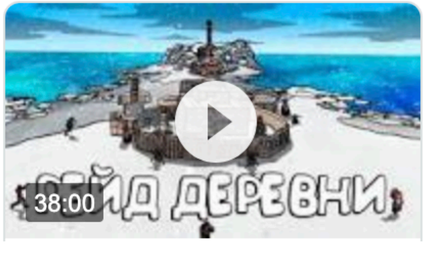
Survive, Cooperate & Raid. Everything wants you to die. You need to find food, shelter and weapons to prevent it. View **Rust** on Steam ...

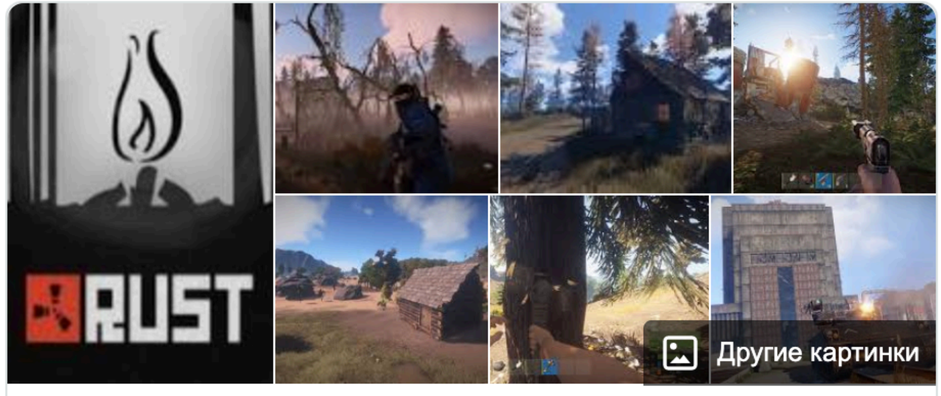
Rust | Facepunch · Rust – Facepunch Studios ... · Changelist · Blog

Видео









Rust

Компьютерная игра

Эта видеоигра понравилась 95% пользователей

Пользователи Google

Rust — компьютерная игра в жанре симулятора выживания, была создана независимой британской студией Facepunch во главе которой с 2014 года состоит Гарри Ньюмэн. Игра вышла на платформах Microsoft Windows, macOS .Выпуск игры

Rust

Компьютерная игра

Эта видеоигра понравилась 95% пользователей

Пользователи Google

Rust — компьютерная игра в жанре симулятора выживания, была создана независимой британской студией Facepunch во главе которой с 2014 года состоит Гарри Ньюмэн. Игра вышла на платформах Microsoft Windows, macOS .Выпуск игры

Rust

- Мультипарадигменный 🛠️

Rust

- ✓ Мультипарадигменный 🛠️
- ▸ Компилируемый ⚡

Rust

- ✓ Мультипарадигменный 🛠️
- ✓ Компилируемый ⚡
- ▸ Без Runtime и GC 🚀

История Rust

- Разработка началась в 2006 (Грэйдон Хор) 👨‍💻

История Rust

- ✓ Разработка началась в 2006 (Грэйдон Хор) 👨‍💻
- ▸ В 2009 подключилась Mozilla 🦊

История Rust

- ✓ Разработка началась в 2006 (Грэйдон Хор) 👨‍💻
- ✓ В 2009 подключилась Mozilla 🦊
- ▸ Официально представлен в июле 2010 🥳

История Rust

- ✓ Разработка началась в 2006 (Грэйдон Хор) 👨‍💻
- ✓ В 2009 подключилась Mozilla 🦊
- ✓ Официально представлен в июле 2010 🥳
- ▸ Первая стабильная версия в мае 2015 🎉

История Rust

- ✓ Разработка началась в 2006 (Грэйдон Хор) 👨‍💻
- ✓ В 2009 подключилась Mozilla 🦊
- ✓ Официально представлен в июле 2010 🥳
- ✓ Первая стабильная версия в мае 2015 🎉
- ▸ Последняя версия: 1.43.1 (май 2020) 🦀

История Rust

- ✓ Разработка началась в 2006 (Грэйдон Хор) 👨‍💻
- ✓ В 2009 подключилась Mozilla 🦊
- ✓ Официально представлен в июле 2010 🥳
- ✓ Первая стабильная версия в мае 2015 🎉
- ✓ Последняя версия: 1.43.1 (май 2020) 🦀
- ▸ ❤️ язык на Stack Overflow с 2016

Ключевые особенности

- Производительность 🚀

Ключевые особенности

- ✓ Производительность 🚀
- ▸ Безопасность ☠️

Ключевые особенности

- ✓ Производительность 🚀
- ✓ Безопасность ☠️
- ▸ Параллелизм 🔥

Области применения

- Web (API, WebAssembly)

Области применения

- ✓ Web (API, WebAssembly) 🕸
- ▸ CLI 💻

Области применения

- ✓ Web (API, WebAssembly) 🕸
- ✓ CLI 💻
- ▸ Embedded (IoT) 💡

Кто использует Rust

- Mozilla 🦊

Кто использует Rust

- Mozilla 🦊
- Dropbox 📦

Кто использует Rust

- Mozilla 🦊
- Dropbox 📦
- CloudFlare ⛅

Кто использует Rust

- Mozilla 🦊
- Dropbox 📦
- CloudFlare ⛅️
- Evrone 🤘

Hello World

```rust
fn main() {
    println!("Hello, world!");
}
```

```rust
use actix_web::{web, App, HttpRequest, HttpServer, Responder};

async fn greet(req: HttpRequest) -> impl Responder {
    let name = req.match_info().get("name").unwrap_or("World");
    format!("Hello {}!", &name)
}

#[actix_rt::main]
async fn main() -> std::io::Result<()> {
    HttpServer::new(|| {
        App::new().route("/{name}", web::get().to(greet))
    })
    .bind("127.0.0.1:8000")?.run().await
}
```

Переменные

```
1 fn main() {
2     let x = 5;
3     x = 6;
4     println!("The value of x is: {}", x);
5 }
```

Иммутабельность

```
error[E0384]: cannot assign twice to immutable variable `x`
 --> src/main.rs:3:5
  |
2 |     let x = 5;
  |         -
  |         |
  |         first assignment to `x`
  |         help: make this binding mutable: `mut x`
3 |     x = 6;
  |     ^^^^^ cannot assign twice to immutable variable
```

Мутабельность

```
1 fn main() {
2     let mut x = 5;
3     x = 6;
4     println!("The value of x is: {}", x);
5 }
```

Функции

```rust
fn main() {
    let x = plus_one(5);
    println!("The value of x is: {}", x);
}

fn plus_one(x: i32) -> i32 {
    x + 1
}
```

Монады: Maybe

```
1 enum Option<T> {
2     Some(T),
3     None,
4 }
5
6 let value: Option<i8> = Some(5);
7 // Or
8 let value: Option<i8> = None;
```

Монады: Result

```rust
enum Result<T, E> {
    Ok(T),
    Err(E),
}

let value: Result<i8, &str> = Ok(8);
// Or
let value: Result<i8, &str> = Err("message");
```

Pattern matching

```
let f = File::open("hello.txt");

let f = match f {
    Ok(file) => file,
    Err(error) => {
        panic!("Problem opening the file: {:?}", error)
    },
};
```

Модули

```rust
mod front_of_house {
    pub mod hosting {
        pub fn add_to_waitlist() {}
    }
}

pub fn eat_at_restaurant() {
    crate::front_of_house::hosting::add_to_waitlist();
    // Or
    front_of_house::hosting::add_to_waitlist();
}
```

Generics

```
fn largest<T>(list: &[T]) -> T {
    let mut largest = list[0];

    for &item in list.iter() {
        if item > largest {
            largest = item;
        }
    }

    largest
}
```

Замыкания

```
1 fn  add_one_v1   (x: u32) -> u32 { x + 1 }
2 let add_one_v2 = |x: u32| -> u32 { x + 1 };
3 let add_one_v3 = |x|             { x + 1 };
4 let add_one_v4 = |x|               x + 1  ;
5
6 let six = add_one_v4(5);
```

Итераторы

```rust
let v1: Vec<i32> = vec![1, 2, 3];

let v2: Vec<_> = v1.iter()
                   .map(|x| x + 1)
                   .collect();

assert_eq!(v2, vec![2, 3, 4]);
```

Traits

```rust
trait Summary {
    fn summarize_author(&self) -> String;
}

impl Summary for Tweet {
    fn summarize_author(&self) -> String {
        format!("@{}", self.username)
    }
}
```

Тестирование

```
1 #[cfg(test)]
2 mod tests {
3     #[test]
4     fn it_works() {
5         assert_eq!(2 + 2, 4);
6     }
7 }
```

Инструменты: rustup

```
# Install rustup nd rust
$ curl https://sh.rustup.rs -sSf | sh

# Update rust
$ rustup update

# Uninstall rustup
$ rustup self uninstall
```

Инструменты: cargo

```
# Create a new cargo package
$ cargo new

# Build and run the package
$ cargo run

# Run tests
$ cargo test

# Update dependencies
$ cargo update

# Package and upload the package to the registry
$ cargo publish
```

Инструменты: rustfmt

```
# Install via rustup
$ rustup component add rustfmt

$ cargo fmt
```

rust-lang/rustfmt

Инструменты: clippy

```
# Install via rustup
$ rustup component add clippy

$ cargo clippy
```

rust-lang/rust-clippy

Инструменты: IDE

- JetBrains IntelliJ https://intellij-rust.github.io
- Vim https://github.com/rust-lang/rust.vim
- Emacs https://github.com/rust-lang/rust-mode
- Sublime Text 3 https://packagecontrol.io/packages/Rust%20Enhanced
- VS Code https://marketplace.visualstudio.com/items?itemName=rust-lang.rust

Пара слов о сложном

- Работа с памятью 🙈
- Время жизни ссылок (Lifetimes) 🙉
- Заимствование (Borrowing) 🙊

С чего начать?

- Book https://doc.rust-lang.org/book 😐
- Examples https://doc.rust-lang.org/rust-by-example 🧐
- Cookbook https://rust-lang-nursery.github.io/rust-cookbook 🤓

Rust ❤️ Open Source

dotenv-linter/dotenv-linter