

Sintaxis

- Estilo
- Indentado
- Extensión
- Función `main()`
- Declaraciones
- Comentarios

# Sintaxis

## Estilo

- Por lo general se utilizar snake\_case para construcciones de `nivel de valor` y CamelCase para construcciones de `nivel de tipo`.
- Puede variar dependiendo del ítem, esto está determinado en el RFC #430.

Elemento	Convención
Crates	snake_case (pero preferible una sola palabra)
Módulos	snake_case
Tipos	CamelCase
Traits	CamelCase
Variantes enum	CamelCase
Funciones	snake_case
Métodos	snake_case
Constructores generales	new o con_mas_detalles
Conversión constructores	desde_algun_otro_tipo
Variables locales	snake_case
Variables estáticas	SCREAMING_SNAKE_CASE
Constantes	SCREAMING_SNAKE_CASE
Tipo parámetros	concise CamelCase, usualmente una letra mayuscula: T
Lifetimes	short, lowercase: 'a

## Indentado

- No se usa tab para indentar, sino 4 espacios

```
fn main() {  
    println!("Hello, world");  
    1234  
}
```

## Extensión

---

- Los archivos en rust tiene la extensión .rs `main.rs`.

## Función `main()`

---

- La palabra clave `fn` se usa para definir una función.
- `main()` es una función predefinida que actúa como punto de entrada al programa, es un orquestador y no tiene parámetros

```
fn main() {  
    // Código rust  
}
```

## Declaraciones

---

- Las declaraciones terminan con `;`, las expresiones no, pero si se le añade se les convierten en declaraciones.
- Las declaraciones no retornan ningún valor.

```
fn main() {  
    let saludo; // se declara la variable saludo  
    saludo=String::from("Hello, world"); // se asigna una cadena a la variable  
    saludo, se convierte esta expresion en una declaracion para que pueda  
    ejecutarse la siguiente linea  
    println!("{}",saludo)// es una expresion y puede o no terminar en ";" ya  
    que no existe ninguna linea siguiente a ejecutarse  
}
```

## Comentarios

---

- Son utilizados para agregar información adicional.
- Se agregan encima de la línea de código.
- El compilador ignora los comentarios

```
fn main() {  
    // Comentarios de una línea  
    /*  
    Comentarios de  
    varias líneas  
    */  
}
```