Rust cheat sheet

Description	Syntax
variable (with type inference)	let x = 10;
mutable variable	let mut x = 5;
variables with difrent types	let sx: i32 = -1; let ux: i32 = 2 let f: f32 = 1.5 let b: bool = false
array	let a: [i32; 5] = [1, 3, 2, 0]
tuple	let t: (i32, f64, u8) = (-5, 1.5, 1)
print with new row	println!("Hello!");
print	print!("Hello!");

if	if x < 5 {
	    println!("true");
	    } else {
	    println!("false")
	}

instantiating with if	let y = if x < 5 { 2 } else { 5 };
function	fn function (x: i32) -> i32

structure	struct Rectangle {
	     width: u32,
	     height: u32,
	}

initiate structure	let r = Rectangle {
	    width: 4
	    height: 5
	}

implement function on struct	impl Rectangle {
	    fn area(&self) -> u32 {
	        self.width * self.height

Description	Syntax
	} }
refrence	let x = &s;
mutable refrence	let x = &mut s;

enum	enum Message {
	    Quit,
	    Move { x: i32, y: i32 },
	    Write(String),
	    ChangeColor(i32, i32, i32),
	}

match	match message {
	    Message::Quit => {}
	    Message::Move { x, y } => {}
	    Message::Write(text) => {}
	    Message::ChangeColor(r, g, b)=> {}
	}

closure (no parameters or return)	let closure = || {
	     //do stuf
	};

closure (parameters and return)	let closure = |num: u32| -> u32 {
	     //do stuf
	};

open tread	let handle = thread::spawn(|| {
	    // do stuf
	})

close tread	handle.join().unrawp()