

Rust Programming Cheat Sheet

Rust is a programming language focusing on speed, concurrency, and safety.

Includes

```
[package]
name = "example"
version = "0.1.0"
edition = "2021"
[dependencies]
rand = "0.8.4"
```

List library dependencies in the file **cargo.toml**

```
use std::io;
use std::cmp::Ordering;
use rand::Rng;
```

List dependencies in your code

Functions

```
fn celsius(fahr: i32) -> i32 {
    let cel = (fahr - 32) * 5 / 9;
    return cel; }
```

fn indicates a function

-> designates the return type

```
fn main() {
    let cel = celsius(10);
    println!("{}", cel); }
```

Default return type (when no **return** statement is used) is **()**

Operators

<code>a & b</code>	Bitwise AND (1 if both bits are 1)	<code><</code>	<code>></code>	greater/less than
<code>a b</code>	Bitwise OR (1 if either bits are 1)	<code><=</code>	<code>>=</code>	greater/less equal
<code>a ^ b</code>	Bitwise XOR (1 if bits differ)	<code>==</code>	<code>!=</code>	equal / not equal
<code>a<<n</code>	Shift bits to the left	<code>&&</code>	<code> </code>	and / or
<code>a>>n</code>	Shift bits to the right	<code>!</code>		not



Rust Programming Cheat Sheet

Variables		
<code>let myvar = "bar";</code>	Set immutable variable myvar containing the string "bar"	
<code>let myvar = 100;</code>	Set immutable variable myvar containing the integer 100	
<code>let myvar: i32 = 100;</code>	Set immutable variable myvar as a 32-bit integer of 100	
<code>let myvar: i64;</code>	Create variable myvar as a 64-bit integer	
<code>let mut myvar: i32;</code>	Create mutable variable myvar as 32-bit integer	
<code>myvar = myvar+1;</code>	Increment myvar	<code>myvar += 1;</code>
<code>let n = 100 as char;</code>	Create immutable variable n , cast as a char	

If / else	Loop
<pre>if n < 0 { //statement; } else if n == 0 { //statement; } else { //statement; }</pre>	<pre>let mut count: i32 = 0; loop { count += 1; if count == 10 { break; } }</pre>
While	For
<pre>let mut n: i32 = 0; while n < 3 { println!("Hello"); n = n+1; }</pre>	<pre>for n in 1..21 { println!("{}", n); }</pre>
Iterator	
<pre>let names = vec!["Alice", "Bob", "Carol"]; for name in names.iter() { println!("Hello {}", name); }</pre>	