

The confusion for newcomers with Rust Development given they have worked in other languages

[help](#)

[sahilmaddi](#) January 19, 2025, 2:39am 1

In rust we have to use `i32`, `i64` for defining the data type instead of like `int` in Java/C/C++ or auto-detecting in python is there any way to understand this easily or is it something considered to be a problem only for newcomers. Will it be easy to handle with experience?

```
let a: i32 = 100;
```

[Kyllingene](#) January 19, 2025, 2:44am 2

The different integer types aren't very unusual once you get used to them. Is there a particular point of confusion we could help clarify?

2 Likes

[sfackler](#) January 19, 2025, 2:49am 3

Java, C, and C++ all have integer types of different sizes just like Rust. They just have different names.

8 Likes

[binarycat](#) January 19, 2025, 3:01am 4

C also has the `uintXX_t` types that are exactly the same as rust's `uXX` types.

1 Like

[collinvandyck](#) January 19, 2025, 4:44am 5

Rust also has `usize` and `isize` if you don't need to be explicit about the width.

2 Likes

[scottmcm](#) January 19, 2025, 5:31am 6

sahilmaddi:

instead of like `int` in Java/C/C++

You mean "like `short` and `int` and `long` and ... in Java/C/C++".

Also types like `char` that are very different between those languages.

[jumpnbrownweasel](#) January 19, 2025, 5:52am 7

sahilmaddi:

is there any way to understand this easily

Does this [section of the Rust book on integers](#) help?

So how do you know which type of integer to use? If you're unsure, Rust's defaults are generally good places to start: integer types default to `i32`. The primary situation in which you'd use `isize` or `usize` is when indexing some sort of collection.

[ZiCog](#) January 19, 2025, 8:11am 8

I'm curious, what. programming languages don't have a selection of different number types for different sizes of signed integers, unsigned integers and floating point numbers?

The only ones I can think of are old Javascript, before it got `BigInts` and typed buffers. Even old Javascript made a distinction between 64 bit floats and 32 bit integers, in its own way. Or old style BASIC back in the 1980's. Oh, and the simple language I wrote a toy compiler for once that only had 32 bit integers.

1 Like

[RustyJoeM](#) January 19, 2025, 8:21am 9

it is also possible that OP means explicit type annotations in general, but used integers as example from some introductory book snippets...

If this is the case, to answer such question - specific types have to be there in the end everytime, but many times compiler can "guess"/identify these from context (if possible) and you dont need to write them yourself into code...

1 Like

[Kyllingene](#) January 19, 2025, 4:01pm 10

At first I thought the same, but in Java/C/C++, you do have to use type annotations. Unless they mean the very specific corner case where you have to specify the type of a variable because it gets used later on in an ambiguous fashion.

[Jesper](#) January 19, 2025, 4:24pm 11
sahilmaddi:

In rust we have to use i32, i64 for defining the data type instead of like int in Java/C/C++ or auto-detecting in python

Python is different from the others - the integers you see there are "BigInts" - they are nice because they never overflow. The price for that of course is they don't have a fixed size.

If you are looking for something like that in rust, take a look at the [bigint crate](#).

[system](#) Closed April 19, 2025, 4:25pm 12

This topic was automatically closed 90 days after the last reply. We invite you to open a new topic if you have further questions or comments.

Related topics

Topic	Replies	Views	Activity
Use of integer types in practice help	18	3402	April 18, 2021
Simple questions on integers help	4	449	June 20, 2020
"Native" integer type? help	37	1054	July 15, 2024
Whether to use type isize or i32, fsize or f64, usize or u32 help	27	3819	March 24, 2023
`(Integer)` type -- Default integer type -- Methods help	5	2166	July 25, 2020

- [Home](#)
- [Categories](#)
- [Guidelines](#)
- [Terms of Service](#)