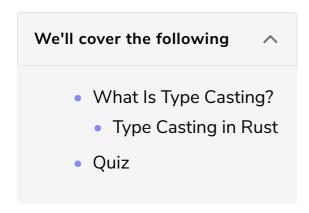
Type Casting Operator

This lesson teaches the type casting operator in Rust.

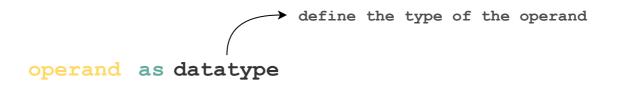


What Is Type Casting?

Type casting is when you convert the data type of the variable to some other data type.

Type Casting in Rust

In Rust, typecasting is done using the as keyword followed by the desired data type of the variable or value.

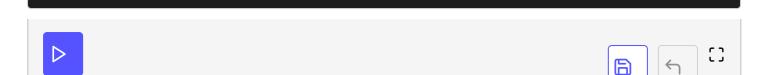


Operator	Operation		
as	convert the data type		

Type casting operator

The following example demonstrates the use of type casting operator in a program:

```
fn main() {
   let a = 15;
   let b = (a as f64) / 2.0;
   println!("a: {}", a);
   println!("b: {}", b);
```



What data types can be type casted?

- *Integer* can be type casted to *floating-point* and *vice versa*.
- Integer can be typecasted to String

What data types cannot be type casted?

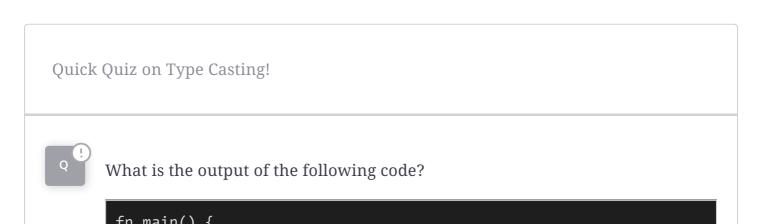
- String (&str) or character cannot be type casted to the data type of type integer or float.
- *Character* cannot be type casted to *String type* and vice versa.

The following code gives an error, \times , because of the invalid type casting operation:

```
fn main() {
    let a: char = 'r'; // cannot be type casted
    let b = a as &str;
    println!("a: {}", a);
    println!("b: {}", b);
}
```

Quiz

Test your understanding of type casting operator in Rust.



```
let a = 15;
let b = (a as f32) / 3.0; //7.5
println!("a:{}",a);
println!("b:{}",b);
```

The next lesson will discuss borrowing and dereferencing operator.

Retake Quiz