

# ! type (RFC 1216)

! represents empty type that has no value:

- they never exist at runtime because there is no way to create one
- they have no logical machine-level representation
- code that handles them never executes
- represent the return type of functions that never return
- can be converted to any other type
- e.g.: `baz() -> !`, `foo() -> Result<!, io::Error>`, `bar() -> Result<u32, !>`

# ! and TryFrom/TryInto

- TryFrom and TryInto with Error associated type of ! would be infallible like From and Into
- There is plan is to implement TryFrom/TryInto<Error=!> for every type that implements From/Into
- Currently it is supported only as return type:

```
1 ▾ fn foo() -> ! {  
2     panic!("boom!")  
3 }
```