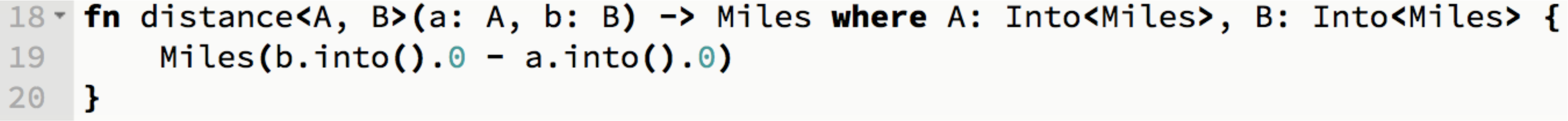
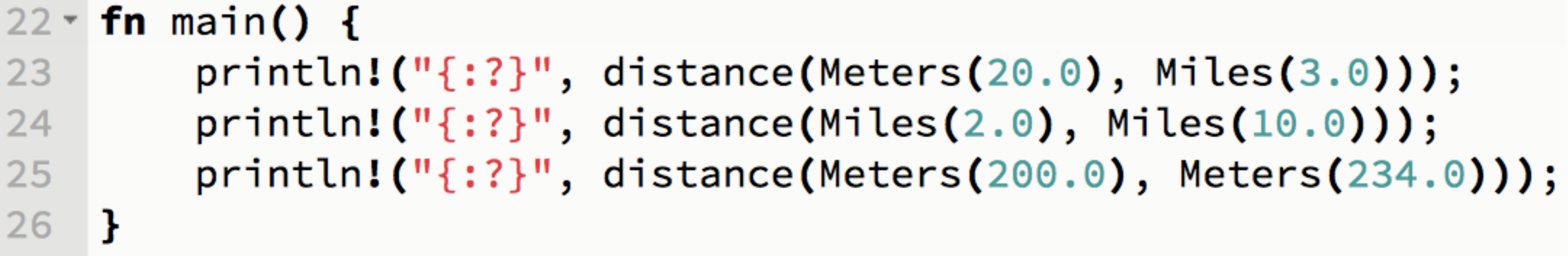
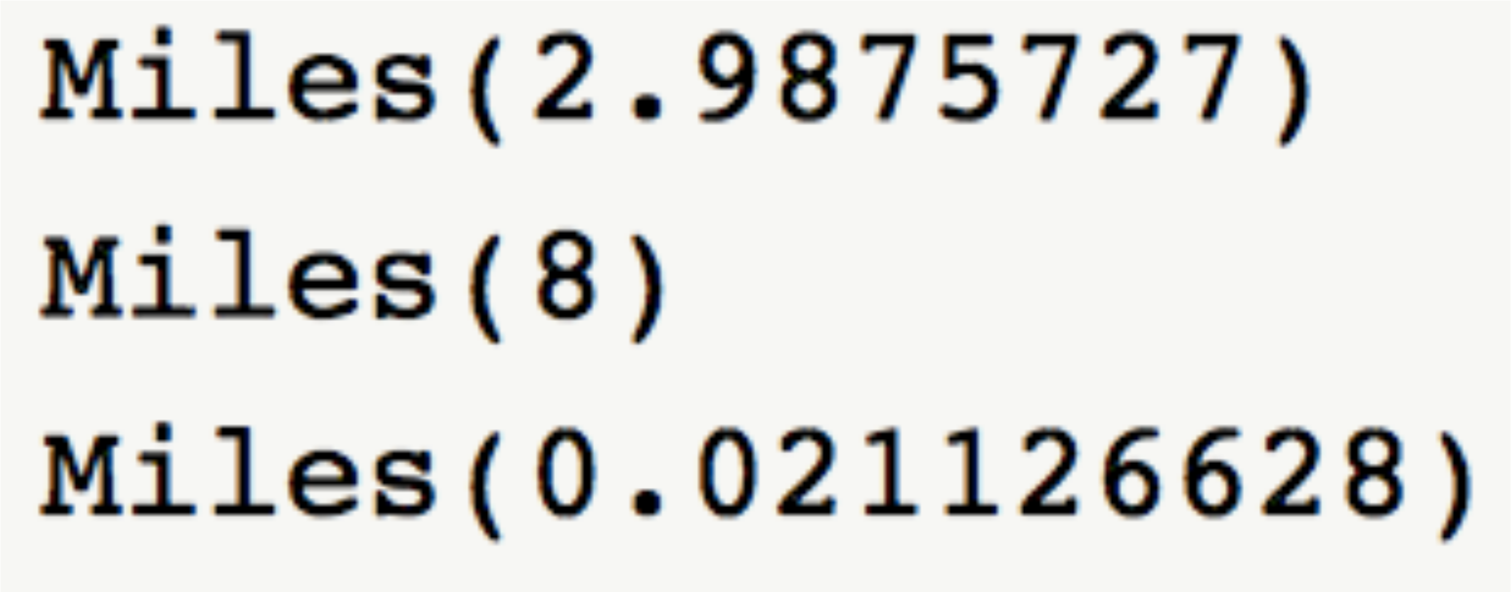
Into and generic argument bound



```rust
fn distance<A, B>(a: A, b: B) -> Miles where A: Into<Miles>, B: Into<Miles> {
    Miles(b.into().0 - a.into().0)
}
```



```rust
fn main() {
    println!("{:?}", distance(Meters(20.0), Miles(3.0)));
    println!("{:?}", distance(Miles(2.0), Miles(10.0)));
    println!("{:?}", distance(Meters(200.0), Meters(234.0)));
}
```



```
Miles(2.9875727)
Miles(8)
Miles(0.021126628)
```

From

- Mirror of `Into`
- Used as trait function: `From::from(a)`
- Used as constructor function: `Miles::from(a)`
- Implements `Into: impl<T, U> Into<U> for T where U: From<T>`