An identifier was used like a function name or a value was expected and the identifier exists but it belongs to a different namespace.

Erroneous code example:

```
struct Foo { a: bool };

let f = Foo();
// error: expected function, tuple struct or tuple variant, found `Foo`
// `Foo` is a struct name, but this expression uses it like a function name
```

Please verify you didn't misspell the name of what you actually wanted to use here. Example:

```
fn Foo() -> u32 { 0 }

let f = Foo(); // ok!
```

It is common to forget the trailing `!` on macro invocations, which would also yield this error:

```
println("");
// error: expected function, tuple struct or tuple variant,
// found macro `println`
// did you mean `println!(...)`? (notice the trailing `!`)
```

Another case where this error is emitted is when a value is expected, but something else is found:

```
pub mod a {
    pub const I: i32 = 1;
}

fn h1() -> i32 {
    a.I
    //~^ ERROR expected value, found module `a`
    // did you mean `a::I`?
}
```