

Implicit Class

```
In [1]: abstract class Shape
case class Square(s:Int) extends Shape
case class Rectangle(l:Int, b:Int) extends Shape
```

```
defined class Shape
defined class Square
defined class Rectangle
```

```
In [2]: implicit class RectangleHelper(l:Int){
  def x(b:Int)=Rectangle(l,b)//def xx(b:Int)=Rectangle(l,b)
}
```

```
defined class RectangleHelper
```

```
In [4]: val r= 5 x 4 //implicit conversion
// same as RectangleHelper(5).x(4)
```

```
r: Rectangle = Rectangle(5, 4)
```

```
In [5]: val r = RectangleHelper(5).x(4)
```

```
r: Rectangle = Rectangle(5, 4)
```

```
In [6]: case class ErrorPrompt(p:String)
def logError(msg:String,prompt:ErrorPrompt)={
  println(prompt.p + msg)
}
```

```
defined class ErrorPrompt
defined function logError
```

```
In [7]: logError("Hello",ErrorPrompt("$> "))
```

```
$> Hello
```

```
In [8]: implicit val prompt:ErrorPrompt=ErrorPrompt("hello> ")
```

```
prompt: ErrorPrompt = ErrorPrompt("hello> ")
```

```
In [8]: logError("Implicits are nice")
```

```
Main.scala:29: not enough arguments for method logError: (msg: String,
prompt: cmd8.this.$ref$cmd5.ErrorPrompt)Unit.
Unspecified value parameter prompt.
logError("Implicits are nice")
      ^
```

```
In [10]: def logError(msg:String)(implicit prompt:ErrorPrompt)={
        println(prompt.p + msg)
      }
```

```
defined function logError
```

```
In [11]: logError("Implicits are nice")
```

```
hello> Implicits are nice
```

```
In [12]: println(implicitly[ErrorPrompt].p)
```

```
hello>
```

```
In [13]: implicit val prompt2:ErrorPrompt=ErrorPrompt("$> ")
```

```
prompt2: ErrorPrompt = ErrorPrompt("$> ")
```

```
In [13]: logError("Implicits are nice, but could be cofusing if there are so many")
```

```
Main.scala:31: ambiguous implicit values:
  both value prompt in class $user of type => cmd7.INSTANCE.$ref$cmd5.E
rrorPrompt
  and value prompt2 in class $user of type => cmd12.INSTANCE.$ref$cmd5.
ErrorPrompt
  match expected type cmd9.INSTANCE.$ref$cmd5.ErrorPrompt
logError("Implicits are nice, but could be cofusing if there are so ma
ny")
      ^
```

```
In [14]: object MyPreferences{ //use implicit inside an object n import whichever
    def logError(msg:String)(implicit prompt:ErrorPrompt)={
        println(prompt.p + msg)
    }
}
import MyPreferences._
```

defined object MyPreferences

Predef

predef in intellij cntrl+b -> predef codes r included in all scala codes by default

```
In [ ]: Safaribooksonline.com
```