TOUR OF SCALA

DEFAULT PARAMETER VALUES

Scala provides the ability to give parameters default values that can be used to allow a caller to omit those parameters.

```
def log(message: String, level: String = "INFO") = println(s"$level: $message")

log("System starting")  // prints INFO: System starting
log("User not found", "WARNING")  // prints WARNING: User not found
```

The parameter `level` has a default value so it is optional. On the last line, the argument `"WARNING"` overrides the default argument `"INFO"`. Where you might do overloaded methods in Java, you can use methods with optional parameters to achieve the same effect. However, if the caller omits an argument, any following arguments must be named.

```
class Point(val x: Double = 0, val y: Double = 0)

val point1 = new Point(y = 1)
```

Here we have to say `y = 1`.

Note that default parameters in Scala are not optional when called from Java code:

```
// Point.scala
class Point(val x: Double = 0, val y: Double = 0)
```

```
// Main.java
public class Main {
    public static void main(String[] args) {
        Point point = new Point(1);  // does not compile
    }
}
```

← previous next →

Contributors to this page:

ckipp01 manishbansal8843 mlachkar ashawley dongxuwang heathermiller