Experiences Unlimited

RAMBLINGS OF A DEVELOPER

HOME › SCALA › NULL, NULL, NIL, NOTHING, NONE, AND UNIT IN SCALA

Null, null, Nil, Nothing, None, and Unit in Scala

BY **MOHAMED SANAULLA** *on* **JULY 12, 2009** • (**10**)

Null– Its a Trait.
null– Its an instance of Null- Similar to Java null.

Nil– Represents an emptry List of anything of zero length. Its not that it refers to nothing but it refers to List which has no contents.

Nothing is a Trait. Its a subtype of everything. But not superclass of anything. There are no instances of Nothing.

None– Used to represent a sensible return value. Just to avoid null pointer exception. Option has exactly 2 subclasses- Some and None. None signifies no result from the method.

Unit– Type of method that doesn't return a value of anys sort.

> *Note: Any is supertype of AnyRef and AnyVal. AnyRef is the supertype of all the reference classes (like String, List, Iterable) in scala. AnyVal is the supertype of all the value classes (like Int, Float, Double, Byte, Short..). Null is a subtype of all the reference classes. null is its only instance. Nothing is subtype of every other type i.e of reference and value classes.*

> *Think- AnyRef == Object in Java.*

The above is a breif summary of a wonderful post by Matt Malone where he has explained all these concepts in depth with examples. Read the blog post **here**.

10 replies

Matt

July 23, 2009 • 8:19 pm

Glad you enjoyed the article. Thanks for the shout-out.

★ Like

> ### i82much
>
> **July 3, 2010 • 3:15 am**
>
> Thanks for this. I got my Nil and null mixed up and caused some confusion with why I couldn't instantiate some objects as Nil.
>
> ★ Like

egg

July 22, 2011 • 12:37 am

fun chatting about nothing. Don't forget the object _scala.xml.Null_. I was stuck for some time, mistaking it for null. Its used in scala.xml.MetaData which in turn is used for xml attributes.

Scala is fun!

★ Like

DS

October 31, 2011 • 8:58 pm

You have written something that is someway wrong :
"Unit- Type of method that doesn't return a value of anys sort."
Unit is used like void, but is slightly different. As in PF all functions MUST return a value.
Unit is a TYPE (sub class of AnyVal) that has only one value : (), and () is the (useless) value 😉that is returned by functions of type Unit.

★ Like

> ### Mohamed Sanaulla

October 31, 2011 • 9:52 pm

It was in a way conveying the same information. I didnt wan't to explain more as I was trying to link a well explained article.

Thanks a lot for bringing this up. It makes sense to add some more information

★ Like

Raúl Raja (@raulraja)

August 26, 2013 • 5:06 pm

Here is a link to some comments and clarifications on twitter related to this post **https://twitter.com/itemstate/status/371957550012317696**

★ Like

Mohamed Sanaulla

August 26, 2013 • 7:14 pm

Thanks for sharing and also pointing me to the clarification. I have replied to the tweet. I will look into this and update the post as the content is not my original one, instead a summary of the blog post linked towards the end of the post.

★ Like

Mohamed Sanaulla

August 26, 2013 • 7:41 pm

Clarification for the tweet mentioned in your comment:

1. From the documentation of Unit:

```
/** `Unit` is a subtype of [[scala.AnyVal]]. There is only one value of type
* `Unit`, `()`, and it is not represented by any object in the underlying
* runtime system. A method with return type `Unit` is analogous to a Java
* method which is declared `void`.
*/
```

From what I stated about Unit: It says: return type of the Method that doesn't return value of any sorts. The intention here is to state that there is no value

> backed/contained with in it.
>
> 2. This post doesn’t establish any context about how these types are used is because this post is a summary of an interesting post I found (which I have also linked towards the end of my post) where in there are examples which show how these types are used.
>
> ★ Like

Arleigh Dickerson

June 14, 2014 • 9:15 pm

Thanks

★ Like

Trackbacks

1. **Scala: The Successor to the Throne « Metaphysical Developer**