Subject: Re: !null values in arrays
Posted by penteado on Thu, 16 Dec 2010 15:39:54 GMT
View Forum Message <> Reply to Message

On Dec 16, 1:14 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:
> Purely for perspective into the issue, ruby handles operations with null values thusly:
>
> irb(main):001:0> 3 + nil
> TypeError: nil can't be coerced into Fixnum
> from (irb):1:in `+'
> from (irb):1
>
> and python similarly:
>
>>>> 3 + None
>
> Traceback (most recent call last):
> File "<stdin>", line 1, in ?
> TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'

Both appropriate. !null and NaN are not the same, and are not the same as 0. In fact, their lack in more primitive languages is an important hassle.

0 is not, intrinsically, the same as missing data. In particular, most mathematical operations on NaNs (say, adding a value to a NaN) should always return NaN to make sense. If an array has missing data, data cannot be magically created on the missing elements because, say, a scalar was added to the array.

NaN is special in the sense that it is a floating point value, part of the IEEE 754 standard, and properly recognized by hardware and other software. !null is a language construct, which necessarily varies among languages, as they vary in what constitutes a variable. It is an undefined variable, so one can assign to it, but not operate on it, the same way that one cannot operate on an undefined variable, but can create it by assignment.