Ruby master - Feature #11530

unicode planes

09/15/2015 10:16 PM - eike.rb (Eike Dierks)

Status:	Feedback	
Priority:	Normal	
Assignee:		
Target version:		

Description

Back then, there was ASCII, 7bit.

We are somehow still stuck to this.

All the parsing still is stuck in that old 7bit world.
While there are so many nice symbols in unicode
that we could put to use to make our code shine.

ruby2 does allow for the use of unicode characters throughout,
but it does not yet differentiate the use of the unicode planes.

I'd like to suggest that some planes of the unicode space
should be reserved from the use as indentifiers.

I'd like to suggest that all characters from the plane of mathematical operators
should be reserved, and should not be parsed as identifiers.

This might also apply to the uppercase greek letters,
which are commonly used in mathematical formulae.

This would be no problem, just a function:
Σ(from:0, to:k){|i| i*2}

I'd like to suggest to reserve the binary operators for future use:
let me give an example:
a ∩ b # intersect
a ∪ b # union

History

#1 - 09/16/2015 02:06 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Status changed from Open to Feedback

Eike Dierks wrote:

> This might also apply to the uppercase greek letters,
> which are commonly used in mathematical formulae.
>
> This would be no problem, just a function:
> Σ(from:0, to:k){|i| i*2}

I can't get your point why it is no problem.

And Unicode defines mathematical symbols separately from Greek letters.
e.g., U+2211;N-ARY SUMMATION, ∑

#2 - 09/21/2015 05:46 AM - duerst (Martin Dürst)

Eike Dierks wrote:

> Back then, there was ASCII, 7bit.

> ruby2 does allow for the use of unicode characters throughout,

but it does not yet differentiate the use of the unicode planes.

I'd like to suggest that some planes of the unicode space
should be reserved from the use as indentifiers.

There are exactly 17 planes in Unicode (the BMP and 16 planes that need surrogate pairs in UTF-16/4 bytes in UTF-8), see
https://en.wikipedia.org/wiki/Plane_%28Unicode%29. The majority of these planes is still completely empty.

What you seem to meen are not planes. In some cases, it may be blocks (see https://en.wikipedia.org/wiki/Unicode_block), but in other cases, one
would have to decide character-by-character.

The main reason this hasn't been done (yet?) is that while such symbols may be great to look at (if they are supported in the relevant fonts), they
aren't easy to input for most programmers.