Ruby master - Bug #16842

`inspect` prints the UTF-8 character U+0085 (NEXT LINE) verbatim even though it is not printable

05/09/2020 02:35 PM - sawa (Tsuyoshi Sawada)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

```ruby
ruby -v:
ruby 2.8.0dev (2020-05-09T13:24:57Z
master 889b0fe46f) [x86_64-linux]
```

<table>
<thead>
<tr>
<th>Backport:</th>
<th>2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN</th>
</tr>
</thead>
</table>

**Description**

The UTF-8 character U+0085 (NEXT LINE) is not printable, but inspect prints the character verbatim (within double quotation):

```ruby
0x85.chr(Encoding::UTF_8).match?(/\p{print}/) # => false
0x85.chr(Encoding::UTF_8).inspect #=> "\n"
```

My understanding is that non-printable characters are not printed verbatim with inspect:

```ruby
"\n".match?(/\p{print}/) # => false
"\n".inspect #=> "\n"
```

while printable characters are:

```ruby
"a".match?(/\p{print}/) # => true
"a".inspect # => "a"
```

I ran the following script, and found that U+0085 is the only character within the range U+0000 to U+FFFF that behaves like this.

```ruby
def verbatim?(char)
  !char.inspect.start_with?('%r{"\n[a-z]}
end

def printable?(char)
  char.match?(/\p{print}/)
end

(0x0000..0xffff).each do |i|
  begin
    char = i.chr(Encoding::UTF_8)
  rescue RangeError
    next
  end
  puts '
#x' % i unless verbatim?(char) == printable?(char)
end
```

Powered by TCPDF (www.tcpdf.org)