

## Ruby trunk - Bug #15596

### Kernel.warn without arguments should do the same as Kernel.warn(nil)

02/08/2019 01:05 PM - kke (Kimmo Lehto)

<b>Status:</b> Open	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> ruby 2.6.1p33 (2019-01-30 revision 66950) [x86_64-darwin18]	<b>Backport:</b> 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

#### Description

Kernel.warn without arguments does not print an empty line to \$stderr.

This is inconsistent with Kernel.puts and it feels weird, because it does not act like a regular Ruby method would (if it was written in Ruby instead of C, it would probably be defined like `def warn(msg = nil)` and calling it with or without nil as argument would make no difference)

#### Expected behavior:

```
irb(main):001:0> warn nil
=> nil
irb(main):002:0> warn
=> nil
```

#### Actual behavior:

```
irb(main):001:0> warn nil
=> nil
irb(main):002:0> warn
=> nil
```

#### History

##### #1 - 02/08/2019 02:42 PM - shevegen (Robert A. Heiler)

This is indeed a (to me) somewhat surprising behaviour; not that I guess many have encountered it (and it probably is quite minor, which may explain why that has not been reported before? Unless that was a more recent change perhaps).

The current documentation is at:

<https://ruby-doc.org/core-2.6.1/Kernel.html#method-i-warn>

If the behaviour is not a bug then I think it should be mentioned in the documentation at the least briefly to explain why it behaves that way.

If the behaviour is a bug, though, then I think I concur with Kimmo, but I really do not know any specifics. It did seem strange though, since I was not able to determine as to why `warn nil` would be treated differently than `warn` without arguments.

##### #2 - 02/08/2019 02:50 PM - nobu (Nobuyoshi Nakada)

It is a natural behavior.

This method prints multiple arguments to STDERR, per line.  
`warn nil` passes one argument and prints one line, whereas `warn` passes no argument and prints zero line.