Ruby - Feature #14255

Deprecate $SAFE support in ERB

12/28/2017 05:35 AM - k0kubun (Takashi Kokubun)

Status:	Closed	
Priority:	Normal	
Assignee:	k0kubun (Takashi Kokubun)	
Target version:	2.6	

Description

See https://bugs.ruby-lang.org/issues/14250.

Proc-level $SAFE change does no longer make sense. This should be deprecated and then removed. But the problem is that the position of safe_level argument is in the middle of argument list of ERB#initialize.

So to remove it safely, we should make them keyword arguments and deprecate the current non-keyword-argument interface. Having both interfaces would be backward-compatible because all of arguments don't take {}.

before

```
ERB.new("<%= 'foo' %>", 1, '%', '_erbout')
```

after

```
ERB.new("<%= 'foo' %>", 1, '%', '_erbout') # print deprecation in 2.6, and removed in 2.7
ERB.new("<%= 'foo' %>", trim_mode: '%', eoutvar: '_erbout') # no safe_level support from first
```

History

#1 - 12/28/2017 05:36 AM - k0kubun (Takashi Kokubun)

- Subject changed from Deprecate $SAFE support from ERB to Deprecate $SAFE support in ERB

#2 - 12/28/2017 05:37 AM - k0kubun (Takashi Kokubun)

- Related to Feature #14250: Make `$SAFE` process global state and allow to set 0 again added

#3 - 12/28/2017 05:53 AM - k0kubun (Takashi Kokubun)

- Status changed from Open to Closed

Changed from ruby-dev to ruby-core https://bugs.ruby-lang.org/issues/14256

#4 - 12/28/2017 05:53 AM - k0kubun (Takashi Kokubun)

- Related to deleted (Feature #14250: Make `$SAFE` process global state and allow to set 0 again)