

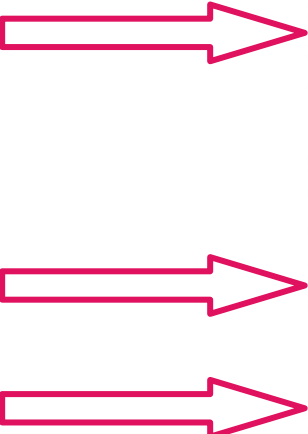
Based on "*Debugging in Ruby (IDE => puts => ...?)*"
presented 05 June 2019@CSRug Ruby Meetup (first Wednesday in the month)

TERMINAL FRIENDLY DEBUGGING IN RUBY

=) K

PUTS

```
1 def boo(a)
2   puts a.inspect # OR "p a" ;o)
3   puts "boo"
4 end
5
6 puts 'Boo - before'
7 boo('t')
8 puts 'Boo - after'
```

Three pink arrows are drawn on the left side of the code, pointing to specific lines. The first arrow points to line 2, the second to line 6, and the third to line 8.

IRB (PRY)



```
1 def boo(a)
2   require 'irb'; binding.irb
3   puts "boo"
4 end
5
6 boo('t')
```

TRACER

```
ruby -rtracer trace_simple.rb
```

```
#0:.../kernel_require.rb:54:Kernel:<:      return gem_original_require(path)
#0:trace_simple.rb:1::-: def boo(a)
#0:trace_simple.rb:4::-: boo('t')
#0:trace_simple.rb:1:Object:>: def boo(a)
#0:trace_simple.rb:2:Object:-:   puts "boo"
boo
#0:trace_simple.rb:3:Object:<: end
```

```
ruby trace_add_tracer.rb
```

```
#0:trace_add_tracer.rb:7::-: Tracer.on { boo('t') }
#0:trace_add_tracer.rb:3:Object:>: def boo(a)
#0:trace_add_tracer.rb:4:Object:-:   puts "boo"
boo
#0:trace_add_tracer.rb:5:Object:<: end
```

```
1 require 'tracer'
2
3 def boo(a)
4   puts "boo"
5 end
6
7 Tracer.on { boo('t') }
```



DEBUG[GER] (+BYEBUG)

ruby debug_add_debug.rb

```
debug_add_debug.rb:3: puts "boo"
(rdb:1) p a
"t"
(rdb:1) l
[-2, 7] in debug_add_debug.rb
  1 def boo(a)
  2 require 'debug'
=> 3 puts "boo"
  4 end
  5
  6 boo('t')
(rdb:1) c
boo
```

ruby -rdebug debug_simple.rb

```
debug_simple.rb:1: def boo(a)
(rdb:1) l
[-4, 5] in debug_simple.rb
=> 1 def boo(a)
  2 puts "boo"
  3 end
  4
  5 boo('t')
(rdb:1) break 2
Set breakpoint 1 at debug_simple.rb:2
(rdb:1) c
Breakpoint 1, boo at debug_simple.rb:2
debug_simple.rb:2: puts "boo"
(rdb:1) p a
"t"
```

BYEBUG

--no-stop

.byebugrc [~/, \$PWD/]

MY_BUG.BASH & CUSTOM VIM COMMAND

```
1 #!/bin/bash
2 file=${1%:*}
3 break_point=$1
4
5 echo "break ${break_point}" > .byebugrc
6
7 byebug --no-stop ${file}
8
9 if [ -f .byebugrc ]; then
10     rm .byebugrc
11 fi
```

```
:command MyBug execute "!my_bug.bash " .shellescape(expand('%'), 1).":".shellescape(line('.'), 1)
```


THANKS



@otaznik_net



https://github.com/otaznik-net/debug_rb