

#55  
Smarthead  
ruby tutorial

## ⇒ Proc and lambda

What is so special about Proc?

\* in ruby everything is an object except blocks and methods.

\* Blocks are not objects in ruby.

\* In order to convert this block <sup>✓</sup> to an object we are using this in proc class.

\* Proc class can turn Block <sup>✓</sup> into an object by wrapping blocks in it's instance.

\* Once we turn our block into an object, then whatever operation we can do on a normal object, all those operations we can perform on these blocks objects too. And also when we convert our block object

into a normal object - with the help of this proc class, and this turns our blocks into a first class function all same

✓  $\text{proc object} = \text{normal object} = \text{block object}$

This turns our block into a first class function and then we can perform all sorts of things with blocks as we can do with a normal object

4 ways to create a proc object

1) Using Proc.new.

there is Proc class and proc object

This is a syntax that is used to wrap a block into the instance of the proc class.

- 2) Using the proc method in kernel module
- 3) Using the kernel lambda methods
- 4) Implicit way  $\swarrow$  using yield statement  
 $\searrow$  using concept (&BLOCK)