

Try out different examples on above topics by their own on ruby console

1. Block

```
irb(main):011* def greet
irb(main):012*   yield "Pranali"
irb(main):013> end
irb(main):014>
irb(main):015> greet { |name| puts "Hello, #{name}!" }
Hello, Pranali!
=> nil
```

2. Types of Arguments

1) Default argument:

```
dell@dell-Latitude-3440:~$ irb
irb(main):001> #Default Arguments
irb(main):002* def greet(name='Pranali',age,salary)
irb(main):003*   puts "Hello #{name}, you are #{age} years old #{salary}"
irb(main):004> end
=> :greet
irb(main):005> greet(23,200)
Hello Pranali, you are 23 years old 200
=> nil
```

2) Keyword argument:

```
irb(main):024> #Keyword Arguments
irb(main):025* def greet(name:,age:)
irb(main):026*   puts "hello #{name}, you are #{age} years old"
irb(main):027> end
irb(main):028> greet(name:'Pranali',age:23)
hello Pranali, you are 23 years old
=> nil
irb(main):029> █
```

3) Positional argument:

```

irb(main):009> #Positional Arguments
irb(main):010> def greet(name,age)
irb(main):011>   puts "hello #{name},you are #{age} years old"
irb(main):012> end
irb(main):013> greet("Pranali",23)
hello Pranali,you are 23 years old
=> nil
irb(main):014>

```

4)Variable-Length argument:

```

irb(main):006> #Variable-Length Arguments
irb(main):007> def sum(*numbers)
irb(main):008>   numbers.sum
irb(main):009> end
irb(main):010> puts sum(1,2,3,4,6)
16
=> nil

```

3.Lambda:

```

irb(main):016> #Basic Lambda
irb(main):017> greet=->(name){ "Hello,#{name}" }
irb(main):018> puts greet.call("Pranali")
Hello,Pranali
=> nil
irb(main):019> #Lambda with Argument Checking
irb(main):020> sum=->(a,b){a+b}
irb(main):021> puts sum.call(2,3)
5
=> nil
irb(main):022> puts sum.call(2)
(irb):20:in `block in <top (required)>': wrong number of arguments (given 1, expected 2) (ArgumentError)
    from (irb):22:in `<main>'
    from /var/lib/gems/3.0.0/gems/irb-1.15.2/exe/irb:9:in `<top (required)>'
    from /usr/local/bin/irb:25:in `load'
    from /usr/local/bin/irb:25:in `<main>'
irb(main):023> # Lambda with return
irb(main):024> def test_lambda
irb(main):025>   greet=->{return "GoodBye"}
irb(main):026>   result=greet.call
irb(main):027>   puts "This won't be printed!"
irb(main):028>   return result
irb(main):029> end
irb(main):030> puts test_lambda
This won't be printed!
GoodBye
=> nil

```

4.Proc:

```

dell@dell-Latitude-3440:~$ irb
irb(main):001> #Basic Proc
irb(main):002> greet=Proc.new{|name| "Hello,#{name}"}
irb(main):003> puts greet.call("Pranali")
Hello,Pranali
=> nil
irb(main):004> #Proc with Argument checking
irb(main):005> sum=Proc.new{|a,b| a+b}
irb(main):006> puts sum.call(5,6)
11
=> nil
irb(main):007> puts sum.call(5)
(irb):5:in `+': nil can't be coerced into Integer (TypeError)
    from (irb):5:in `block in <top (required)>'
    from (irb):7:in `<main>'
    from /var/lib/gems/3.0.0/gems/irb-1.15.2/exe/irb:9:in `<top (required)>'
    from /usr/local/bin/irb:25:in `load'
    from /usr/local/bin/irb:25:in `<main>'
irb(main):008> #Proc with return
irb(main):009* def test_proc
irb(main):010*   greet = Proc.new { return "Goodbye" }
irb(main):011*   result = greet.call
irb(main):012*   puts "This will never be printed"
irb(main):013*   return result
irb(main):014> end
irb(main):015> puts test_proc
Goodbye
=> nil
irb(main):016> █

```

5>Returns:

```

irb(main):005* def add(a,b)
irb(main):006*   a+b
irb(main):007> end
irb(main):008> puts add(2,3)
5
=> nil
irb(main):009>

```