



Iterators and Loops



Outline

1. While, Until loops
2. The Ruby each iterator
3. The "for in" loop in Ruby
4. Select, map, inject



1. Loop - While

Syntax

```
while conditional [do]
  code
end
```

OR

```
begin
  code
end while conditional
```

Example

```
i = 0
while i < 10 do
  puts("Inside the loop i = #{i}" )
  i +=1
end
```

```
i = 0
begin
  puts("Inside the loop i = #{i}" )
  i +=1
end while i < 10
```



1. Loop - Until (cont.)

Syntax

```
until conditional [do]
  code
end
```

OR

```
begin
  code
end until conditional
```

Example

```
i = 0
until i > 10 do
  puts("Inside the loop i = #{i}" )
  i +=1
end
```

```
i = 0
begin
  puts("Inside the loop i = #{i}" )
  i +=1
end until i > 10
```



2. The Ruby each iterator

```
collection.each{|item| #something}
```

OR

```
collection.each do |item|  
  # something  
end
```

Example:

```
[123, 345, 678, 910].each{|item| item*2}
```

OR

```
[123, 345, 678, 910].each do |item|  
  item*2  
end
```



3. Loop - For

Syntax

```
for variable [, variable ...] in expression [do]  
  code  
end
```

```
(expression).each do |variable[, variable...]|  
  code  
end
```

Example

```
for i in 0..5  
  puts "Value of local variable is #{i}"  
end
```

```
(0..5).each do |i|  
  puts "Value of local variable is #{i}"  
end
```



3. Loop - Excute in loop (cont.)

Break

```
for i in 0..5
  if i > 2 then
    break
  end
  puts "Value of local variable is #{i}"
end
```

Next

```
for i in 0..5
  if i < 2 then
    next
  end
  puts "Value of local variable is #{i}"
end
```



3. Loop - Excute in loop (cont.)

Redo

```
for i in 0..5
  if i < 2 then
    puts "Value of local variable is #{i}"
    redo
  end
end
```

Retry

```
begin
  do_something # exception raised
rescue
  # handles error
  retry # restart from beginning
end

for i in 1..5
  retry if some_condition # restart from i == 1
end
```




4. Select, map, inject

Select method

```
> my_array = [1,2,3,4,5,6,7,8,100]
> my_array.select{|item| item%2==0 }
=> [2,4,6,8,100]

> my_hash = {"Joe" => "male", "Jim" =>
"male", "Patty" => "female"}
> my_hash.select{|name, gender| gender
== "male" }
=> {"Joe" => "male", "Jim" => "male"}
```

Map method

```
> my_array = [1,2,3,4,5,6,7,8,100]
> my_array.map{|item| item*2}
```

```
> @users = User.all
> @users.map{|user| user.email}
```

Inject method

```
> my_array = [1,2,3,4,5,6,7,8,100]
> my_array.inject(0){|running_total,
item| running_total + item}
=> 120
```



References

- ❖ <http://ruby-doc.org/>
- ❖ <https://www.tutorialspoint.com/>



Thank you for listening!