



# Control Structure & Iterators

Anupam Sharma  
Sr Software Consultant  
961 123 0007  
[asharma@qwinix.io](mailto:asharma@qwinix.io)

# What is Control Structure?

Control structures allow programs to take different paths depending on different values and conditions.

Or, we can say that control structures decides the flow of execution of program.

And, yes its also plays big role in implementing business logics.

# What are Control Structure available in Ruby?

- Simple If with fixed condition
- If..end
- If..Else..end
- if..Elsif.. Else..end
- Simple Unless with fixed condition
- Unless..end
- Unless..Else..end
- Case statement

# Simple If with fixed condition

Syntax of Simple If

```
code if condition
```

E.g of Simple If

```
k = true
```

```
puts "Welcome to Qwinix" if k
```

```
k = false
```

```
puts "Oops no output" if k
```

# If..end

## Syntax of If..end

```
if condition  
  code  
end
```

## E.g of If..end

```
k = true  
If k  
  puts "Hello, your conditions is true"  
end
```

# If..Else..end

## Syntax of If..Else..end

```
if condition  
  code1  
else  
  code2  
end
```

## E.g of If..Else..end

```
k = true  
If k  
  puts "Hi, Since k is true"  
else  
  puts "Get lost"  
end
```

## Other way of writing If..else..end

```
k ? "Hi, Since k is true" : "Get lost"
```

Which is a.k.a.???



# Ternary operator

# If..elsif..else..end

## Syntax of If..elsif..else..end

```
if condition
  code1
elsif condition
  code2
else
  code3
end
```

## E.g of If..elsif..else..end

```
k = noon
If (k == 'noon')
  puts "Its lunch time"
elsif (k == 'mid_evening')
  puts "its snacks time"
else
  puts "keep your mouth shut"
end
```



# Simple Unless with fixed condition

Syntax of Simple Unless

code unless condition

E.g of Simple Unless

k = false

puts "Welcome to Qwinix" unless k

Or

puts "Welcome to Qwinix" if !k

# Unless..end

## Syntax of Unless..end

```
unless condition  
  code  
end
```

## E.g of Unless..end

```
k = false  
unless k  
  puts "Hello, your conditions is false"  
end
```

Or

```
if !k  
  puts "Hello, your conditions is false"  
end
```

# Unless..Else..end

## Syntax of Unless..Else..end

```
unless condition  
  code1  
else  
  code2  
end
```

## E.g of Unless..Else..end

```
k = false  
unless k  
  puts "Hi, Since k is false"  
else  
  puts "Get lost"  
end
```

## Other way of writing Unless..else..end

```
!k ? "Hi, Since k is true" : "Get lost"
```

# Case Statement

## Syntax of Case Statement

```
Case expr0  
When expr1, expr2  
    Stmt1  
When expr3, expr4  
    Stmt2  
else  
    Stmt3  
end
```

## E.g of Case Statement

```
age = 5  
case age  
when 0..2  
    puts "Baby"  
when 3..6  
    puts "Little Child"  
when 7..12  
    puts "Child"  
when 12..18  
    puts "Youth"  
else  
    puts "Adult"  
end
```

# That's All FoLKs!!!

Any questions or any doubt for clarification, regarding Control Structures..??

# What is Iterators?

Iterators are not an original concept with ruby. They are in common use in object-oriented languages.

The verb iterate means to do the same thing many times, So an iterator is something that does the same thing many times.



# What are the iterators in Ruby?

- Ruby While Statement
- Ruby While Modifier
- Ruby Until Statement
- Ruby Until Modifier
- Ruby For Statement
- Ruby Break Statement
- Ruby Next Statement
- Ruby Redo Statement
- Ruby Retry Statement

# Ruby While Statement

Syntax of while statement

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

E.g of while statement

```
p = 0
num = 5

while p < num do
  puts "Inside the loop p = #p"
  p += 1
end
```

# Ruby While Modifier

Syntax of while modifier

code while condition

Or

begin

code

end while conditional

E.g of while modifier

```
p = 0
```

```
num = 5
```

```
begin
```

```
  puts "Inside the loop p = #p"
```

```
  p += 1
```

```
end while p < num
```

# Ruby Until Statement

Syntax of until statement

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

E.g of until statement

```
p = 0
num = 5

until p > num do
  puts "Inside the loop p = #p"
  p += 1
end
```

# Ruby Until Modifier

Syntax of until modifier

code until condition

Or

begin

code

end until conditional

E.g of until modifier

```
p = 0
```

```
num = 5
```

```
begin
```

```
  puts "Inside the loop p = #p"
```

```
  p += 1
```

```
end while p > num
```

# Ruby For Statement

Syntax of for statement

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

E.g of for statement

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

For can also be written as below

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



# Ruby Break Statement

Syntax of break statement

```
break
```

E.g of break statement

```
for m in 0..5
```

```
  if m > 2 then
```

```
    break
```

```
  end
```

```
  puts "Value of local variable is #{m}"
```

```
end
```

# Ruby Next Statement

Syntax of next statement

```
next
```

E.g of next statement

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

# Ruby Redo Statement

Syntax of redo statement

```
redo
```

E.g of redo statement

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

# Ruby Retry Statement

Syntax of retry statement

```
retry
```

E.g of retry statement

```
for m in 0..5
```

```
  retry if m > 2
```

```
  puts "Value of local variable is #{m}"
```

```
end
```

**That's all FoLK's**

THANK YOU



HQ | North America  
9155 E Nichols Ave, Suite 300  
Centennial, CO 80112

[www.qwinix.io](http://www.qwinix.io) | 303 459 4310

f [facebook.com/qwinix](https://facebook.com/qwinix)    t [@Qwinix](https://twitter.com/Qwinix)    in [Qwinix Technologies](#)

