



Introduksjon til Ruby

Fagkveld

BEKK

Språket du bruker former
tankene og tankemåtene dine.

A detailed close-up of a green printed circuit board (PCB) with intricate silver-colored traces, numerous surface-mount components like resistors and capacitors, and several multi-pin connectors. The word "Java" is superimposed in white.

Java

A stylized, glowing blue brain with visible neural pathways, set against a solid black background. The word "Ruby" is superimposed in white.

Ruby



Java

“(...) We are the masters. They are the slaves.”



Ruby

Alt er objekter

Alt er objekter

```
"test".class # => String  
"test".length # => 4
```

Alt er objekter

- Også tall

```
-1.abs # => 1
```


Alt er objekter

- Også tall, booleans

```
-1.abs # => 1
```

```
true.class # => TrueClass
```


Alt er objekter

- Også tall, booleans og `nil` er objekter

```
-1.abs # => 1
```

```
v = nil  
v.nil? # => true
```

```
true.class # => TrueClass
```

Operatorer er metoder

```
1 + 1  
1.(+)(1)
```

```
x == 5  
x.==(5)
```

Streng

```
x = "foo"
```

```
y = "foo"
```

```
x == y
```

```
# => true
```

```
x.equal?(y)
```

```
# => false
```


Streng vs symbol

```
x = "foo"  
y = "foo"
```

```
x == y  
# => true  
x.equal?(y)  
# => false
```

```
x = :foo  
y = :foo
```

```
x == y  
# => true  
x.equal?(y)  
# => true
```

Streng vs symbol

```
x = "foo"  
y = "foo"
```

```
x == y  
# => true  
x.equal?(y)  
# => false
```

Mutable

```
x = :foo  
y = :foo
```

```
x == y  
# => true  
x.equal?(y)  
# => true
```

Immutable

If

```
if condition
  # do something
elsif other_condition
  # do something else
else
  # do something other
end
```


If

```
if condition
  # do something
elsif other_condition
  # do something else
else
  # do something other
end
```

```
x = 5 if condition
```

Unless

```
unless condition  
  # do something  
else  
  # do something else  
end
```

```
x = 5 unless condition
```

Array, Hash og Range

```
[1, 2, "test", true]
```

```
{ :navn => "Kim" }
```

```
(1..5)
```


Array, Hash og Range

```
[1, 2, "test", true].class  
# => Array
```

```
{ :navn => "Kim" }.class  
# => Hash
```

```
(1..5).class  
# => Range
```

Array, Hash og Range

```
(1..5).to_a  
# => [1, 2, 3, 4, 5]
```

Iteratorer

Iteratorer

```
a = [1,2,3]
a.each { |value|
  puts value
}
# 1
# 2
# 3
```

Iteratorer

```
a = [1,2,3]
a.each do |value|
  puts value
end
# 1
# 2
# 3
```

Iteratorer

```
3.times do
  puts "Ruby!"
end
# Ruby!
# Ruby!
# Ruby!
```

```
a = [1,2,3]
a.map do |value|
  value * 2
end
# => [2, 4, 6]
```

Iteratorer

```
h = { :x => 1, :y => "test" }  
h.each do |key, value|  
  puts "#{key} -> #{value}"  
end  
# x -> 1  
# y -> test
```


Lage egen metode

```
def hello_world  
  puts 'Hello UIA!'  
end
```

```
hello_world  
# => Hello UIA!
```

```
def weirdo(x)
  if x == 5
    "Bingo"
  else
    42
  end
end
```

```
puts weirdo(5) # => "Bingo"
puts weirdo("Ultimate?") # => 42
```

Noen konvensjoner

```
def scary!  
  # modifierer objektet  
end
```

```
def raining?  
  # returnerer boolean  
end
```

Klasser — instansmetode

```
class MyClass
  def hello(name)
    "Hi, #{name}"
  end
end

o = MyClass.new
puts o.hello("Bob")
# => Hi, Bob
```



```
class MyClass
  def self.hello(name)
    "Hi, #{name}"
  end
end

puts MyClass.hello("Bob")
# => Hi, Bob
```

Variabler

```
a = 1    # lokal
@a = 2    # instans
@@a = 3   # klasse
$a = 4    # global

puts a, @a, @@a, $a
# 1
# 2
# 3
# 4
```

Variabler

```
class User
  def name=(name)
    @name = name
  end

  def name
    @name
  end
end

user = User.new
user.name = "Kim"
puts user.name
```

Variabler

```
class User  
  attr_accessor :name  
end
```

```
user = User.new  
user.name = "Kim"  
puts user.name
```


Lek med Rubys egne objekter

Lek med Rubys egne objekter

```
class String
  def exclaimate
    "#{self}!"
  end
end

puts "Rop".exclaimate
# => Rop!
```

irb og .rb

- For enkel kodetesting er `irb` veldig bra. Åpne terminalen og skriv inn `irb`.
- Ellers kan du lagre filer med `.rb` og kjøre de med `ruby filnavn.rb`

Lage en webapp!

Lage en webapp!

Og nå blir det litt avansert ...



Vi trenger tre filer

Vi trenger tre filer

- To “kjedelige filer” for Heroku sin del

Vi trenger tre filer

- To “kjedelige filer” for Heroku sin del

```
# config.ru  
require './fagkveld'  
run Sinatra::Application
```

```
# Gemfile  
source 'http://rubygems.org'  
gem 'sinatra'
```

Livecoding

Rubydoc

<http://ruby-doc.org/>

<https://github.com/kjbekkelund/ruby-fagkveld>



BEKK

Kim Joar Bekkelund

Konsulent

kim.bekkelund@bekk.no

BEKK CONSULTING AS
SKUR 39, VIPPETANGEN. P.O. BOX 134 SENTRUM, 0102 OSLO, NORWAY. WWW.BEKK.NO