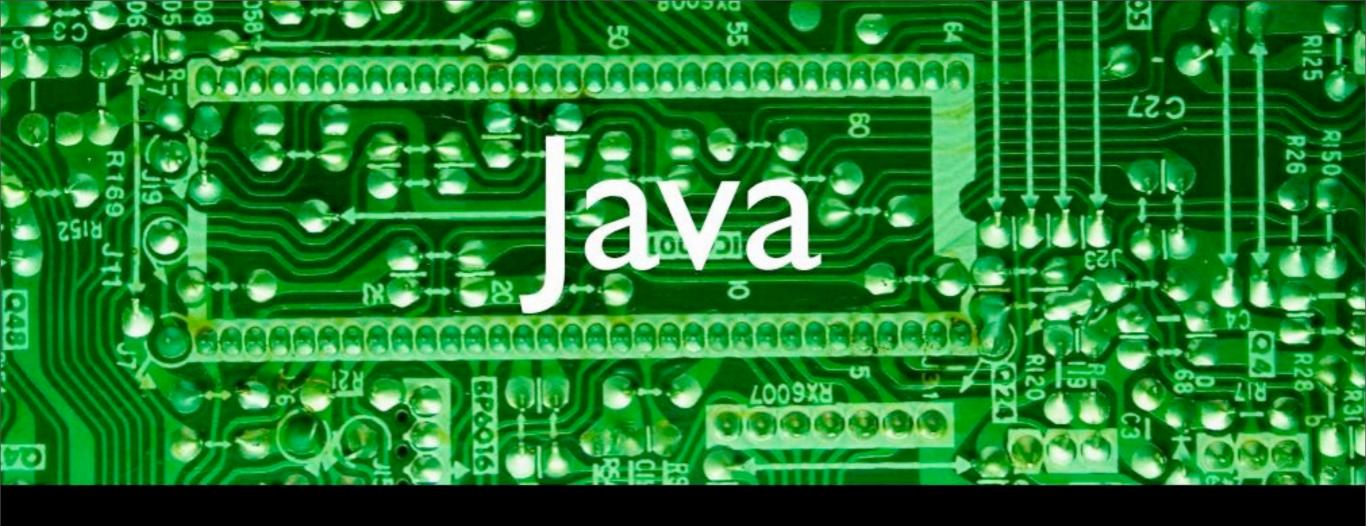
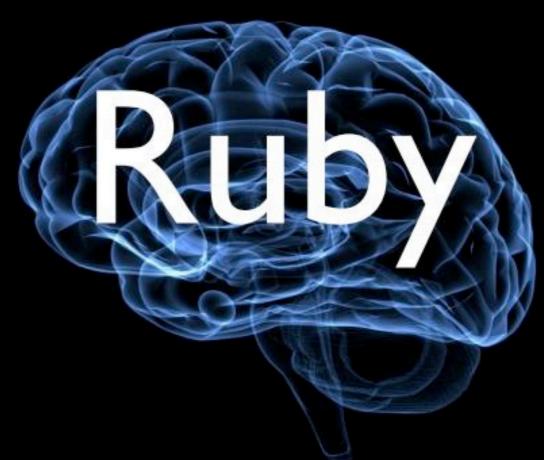


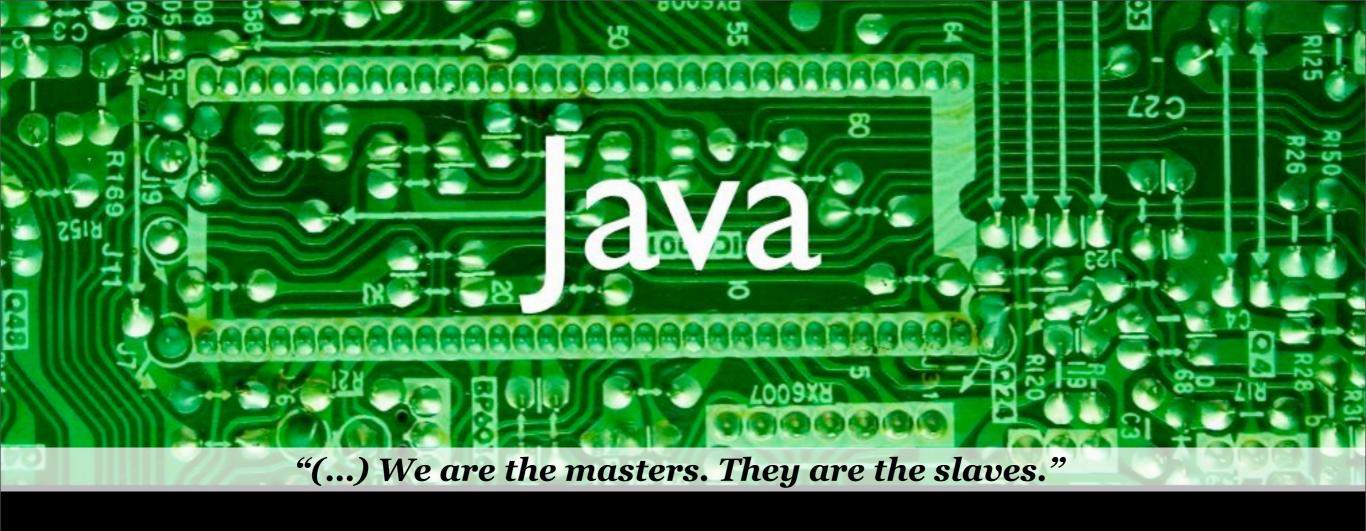
Introduksjon til Ruby

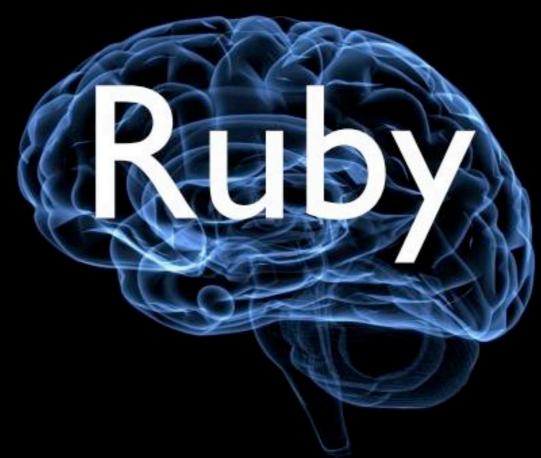
Fagkveld

Språket du bruker former tankene og tankemåtene dine.









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

Også tall

$$-1.abs # => 1$$

Også tall, booleans

$$-1.abs # => 1$$

true.class # => TrueClass

Også tall, booleans og nil er objekter

$$-1.abs # => 1$$

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

true.class # => TrueClass

Operatorer er metoder

$$x == 5$$
 $x == (5)$

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

```
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

```
a = [1,2,3]
a.each { Ivalue!
  puts value
```

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

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

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

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

Lage egen metode

```
def hello_world
  puts 'Hello UIA!'
end
hello_world
# => Hello UIA!
```

Metodesignatur

```
def weirdo(x)
  if x == 5
    "Bingo"
  else
    42
  end
end
puts weirdo(5) # => "Bingo"
puts weirdo("Ultimate?") # => 42
```

Noen konvensjoner

```
def scary!
 # modifiserer 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
```

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 exclamate
    "#{self}!"
  end
end
puts "Rop".exclamate
# => 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

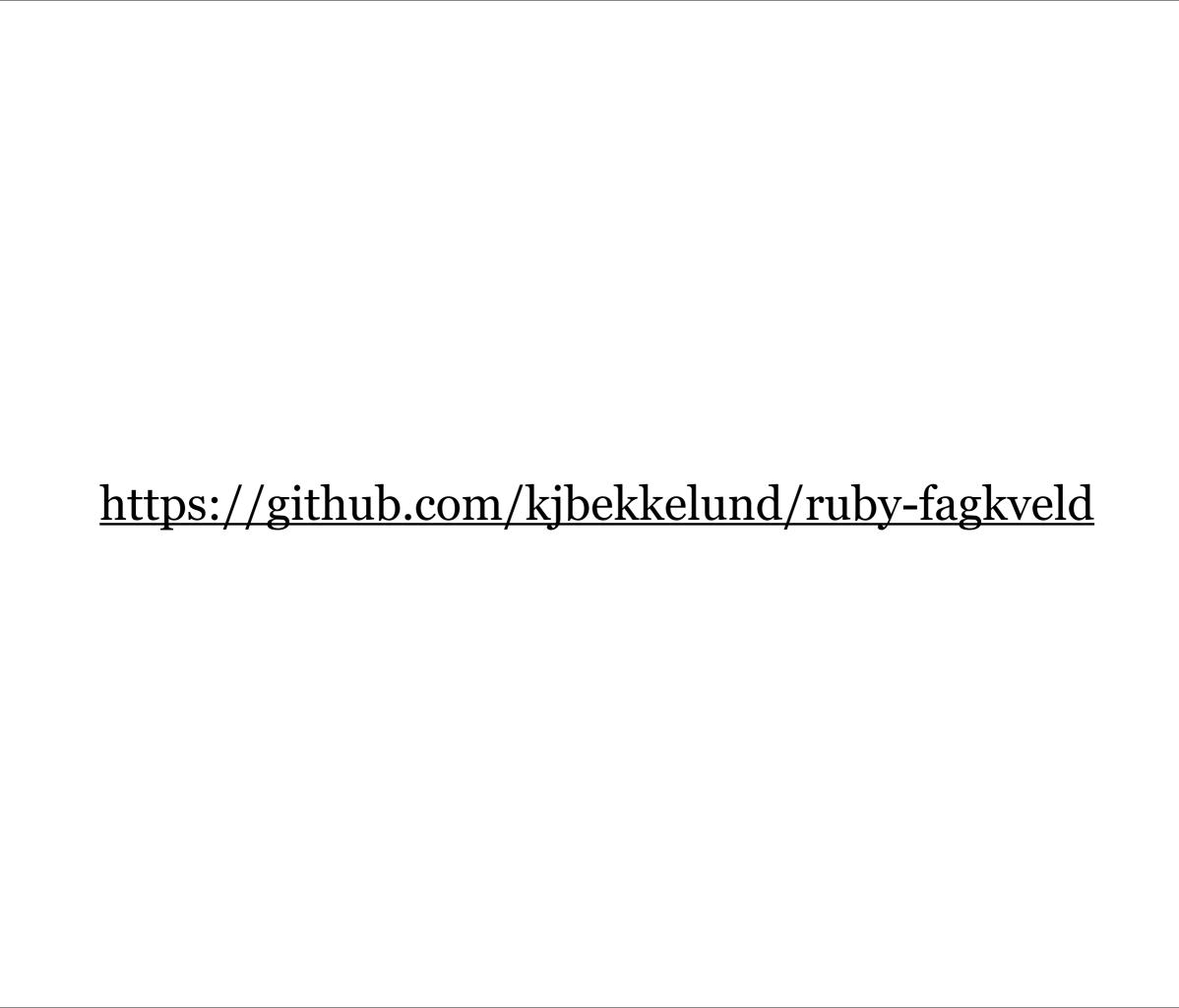
```
# config.ru
require './fagkveld'
run Sinatra::Application
```

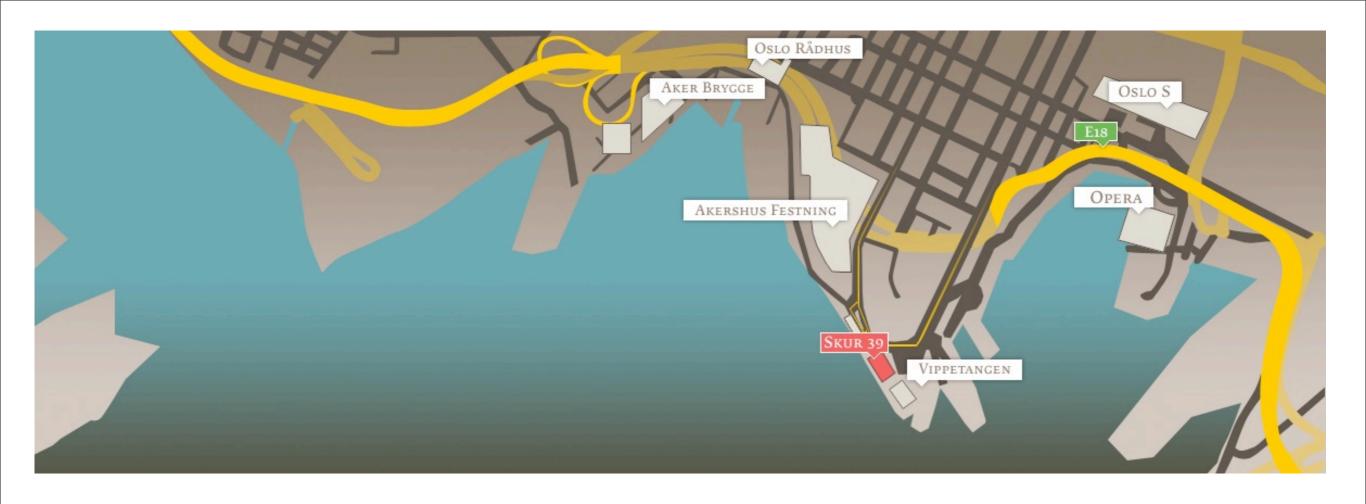
```
# Gemfile
source 'http://rubygems.org'
gem 'sinatra'
```

Livekoding

Rubydoc

http://ruby-doc.org/





3EKK

Kim Joar Bekkelund

Konsulent <u>kim.bekkelund@bekk.no</u>

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