Ruby / Java

The differences…

irb(main):002:0> author

```
=> {:name => “Jan Schulte”, :twitter=> “@schultyy” }
```

Gemeinsamkeiten

- Objektorientiert
- Plattformunabhängig
  - Linux
  - Windows
  - Mac OS X

Workflow

- Java
  - Edit - Compile - Run
    - Compiler
- Ruby
  - Edit - Run
  - Interpretiert

Typisierung

- Java
  - Statisch
  - Alle Typen müssen zur Übersetzungszeit bekannt sein
- Ruby
  - Dynamisch
  - Variablen haben keine Typen
  - Typüberprüfungen passieren zur Laufzeit

Syntax

Java

Ruby

Klassen

- Nur ein Konstruktor
  - `initialize`
- `Car.new` statt `new Car();`
- Instanzvariablen
  - @
  - Private by default
- `attr_accessor`

```ruby
class Car
  attr_accessor :name, :horse_powers

  def initialize(name, hp)
    @name = name
    @horse_powers = hp
  end
end
```

Methoden

- Geben automatisch den letzten Wert zurück
- Werden mit `def` und `end` umschlossen
- Klammern optional
- Default Argumente

```ruby
def say_hi
  "Ahoi!"
end

```

```ruby
def say_hi(name = "John")
  "Ahoi #{name}!"
end

```

Alles ist ein Objekt

```
5.times do
  puts "da schau her"
end
```

Hashes und Symbols

- Schlüssel-Werte Paare
- Symbols
  - Konstanten

```ruby
def to_hash
  {
    :name => @name,
    :manufacturer => @manufacturer,
    :horse_powers => @horse_powers
  }
end
```

Duck typing

>> Wenn es geht wie eine Ente und quakt wie eine Ente, dann ist es eine Ente

```ruby
class Duck
  def swim
    puts "Duck swims"
  end
end

class Goose
  def swim
    puts "Goose swims"
  end
end

def let_it_swim(duck) duck.swim end

let_it_swim(Duck.new)
let_it_swim(Goose.new)
```

> questions?