

# Gurken zum Salat

Behaviour Driven Development  
mit Cucumber

# Cucumber



Behaviour Driven Development  
with elegance and joy

- **Dan North:**

BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology.

It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters

- **Erweiterung des Test Driven**

**Developments:**

**Testfälle in natürlicher Sprache, die Nicht-Entwickler verstehen können**

- **JBehave → RBehave → RSpec →  
Cucumber**

- Implementiert in Ruby
- Ruby, Java, .Net und Flex
- Web-Applikationen
  - Ruby on Rails (Webrat)
  - beliebige Frameworks (Selenium, Watir)
- DSL
- Natürliche Sprache
  - > 30 Sprachen
- Outside-In-Development
- Dokumentation
- Tests

```
gem install cucumber
```

**Scenario:** Happy JUG-MeNue attendees

**Given** someone comes today

**When** Andreas talks about cucumber

**Then** you should gain new knowledge

**And** no-one should fall asleep

**And** no rotten eggs should be thrown

**Feature:** Hello World Feature

**In Order** to ensure that my  
installation works

**As a** Developer

**I want** to run a quick Cucumber test

**Scenario:** Hello World Scenario

**Given** The Action is Hello

**When** The Subject is World

**Then** The Greeting is Hello, World

```
cucumber hello_world.feature
```

```
Given /^The Action is Hello$/ do
  pending
end
```

```
When /^The Subject is World$/ do
  pending
end
```

```
Then /^The Greeting is Hello, World$/ do
  pending
end
```

```
Given /^The Action is ([A-z]*)$/ do |action|  
  @action = action  
end
```

```
When /^The Subject is World$/ do  
  pending  
end
```

```
Then /^The Greeting is Hello, World$/ do  
  pending  
end
```

```
Given /^The Action is ([A-z]*)$/ do |action|  
  @action = action  
end
```

```
When /^The Subject is ([A-z]*)$/ do |subject|  
  @subject = subject  
end
```

```
Then /^The Greeting is Hello, World$/ do  
  pending  
end
```

```
Given /^The Action is ([A-z]*)$/ do |action|
  @action = action
end

When /^The Subject is ([A-z]*)$/ do |subject|
  @subject = subject
end

Then /^The Greeting is (.*)$/ do |greeting|
  greeting.should == @action + ", " + @subject
end
```

- Viele Testfälle?
- Scenario Outlines

**Scenario Outline:** Hello World Scenario

**Given** The Action is **<action>**

**When** The Subject is **<subject>**

**Then** The Greeting is **<greeting>**

**Examples:**

<b>action</b>	<b>subject</b>	<b>greeting</b>
Hello	World	Hello, World
Hello	Andreas	Hello, Andreas
Hello	Darling	Hello Darling
Hello	JUG-MeNue	Hello, JUG-MeNue

- Viele Daten?
- Tabellen

**Feature-File:**

**Given** the following people exist:

<b>name</b>	<b>email</b>	<b>phone</b>
Aslak	aslak@email.com	123
Joe	joe@email.com	234
Bryan	bryan@email.org	456

**Step-Definitions-File:**

```
Given /the following people exist:/ do |people_table|
  people_table.hashes.each do |hash|
    puts hash["name"]
    ...
  end
end
```

**Und jetzt mal mit Rails**

The Pragmatic  
Programmers

# The RSpec Book

Behaviour-Driven Development  
with RSpec, Cucumber,  
and Friends

*David Chelmsky*  
with *Dave Astels,*  
*Zach Dennis,*  
*Aslak Helleøy,*  
*Bryan Helmkamp,*  
and *Dan North*

*Foreword by Robert C. Martin*  
(*Uncle Bob*)

*Edited by Jacquelyn Carter*

The Facets  of Ruby Series

- The Secret Ninja Cucumber Scrolls  
<http://cuke4ninja.com>

Freies Buch über Cucumber, behandelt auch Cucumber und Java

**Vielen Dank!**

**[andreas.schubert@mathema.de](mailto:andreas.schubert@mathema.de)**