



Die Programmiersprache Ruby

Die Geschichte von Ruby

- Veröffentlicht 1995 von Yukihiro “Matz” Matsumoto
- Balance zwischen funktionaler und imperativer Programmierung

Ruby is simple in appearance, but is very complex inside, just like our human body.

Ruby sieht von außen einfach aus, ist von innen aber sehr komplex, genau wie unser menschlicher Körper.

Yukihiro “Matz” Matsumoto
<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/2773>

Theorie: Skriptsprache

- Kein Compiler
- Durch Interpreter direkt ausführbar:
~\$ ruby mein_script.rb

Theorie: Objektorientierte Sprache

- Alles ist ein Objekt:

```
42.class # => Integer  
13.37.class # => Float  
"Zeichenkette".class # => String  
[1, 2, 3].class # => Array  
true.class # => TrueClass  
false.class # => FalseClass
```

- Alle Operatoren sind Methoden:

```
5 + 2 # => 7  
5.+(2) # => 7
```

Theorie: Blöcke

- Code zur späteren Ausführung
- Erlaubt funktionale Konzepte:

```
[1, 2, 3].each do |i|  
  puts i  
end  
# Gibt "1", "2", "3" aus
```

```
(1..10).select {|i| i.odd? }  
# => [1, 3, 5, 7, 9]
```

Theorie: Alles ist ein Ausdruck

- Alles ist ein Ausdruck und hat einen Rückgabewert

```
variable = (three = 3)
# => 3

greater_5 = three > 5
# => false

message = if greater_5
  "#{variable} is greater five."
else
  "#{variable} is less or equal five."
end
# => "3 is less or equal five."
```



Alles hat einen Rückgabewert.

Theorie: Rückgabewert

- Der Rückgabewert ist implizit der letzte Ausdruck

```
def add(a, b)
  result = a + b
  result
end
# => :add

add(4, 9)
# => 13
```



Der Rückgabewert eines Codestücks ist **immer** das Ergebnis des letzten Ausdrucks. Das Schlüsselwort `return` ist optional.

Die interaktive Ruby-Konsole: IRB

Führt direkt Ruby-Anweisungen aus und gibt Ergebnis zurück:

```
> irb  
2.5.1 :001 > 5 + 5  
=> 10
```

Rückgabewert

Auszuführender Befehl

```
2.5.1 :002 > puts "Hallo Welt"  
Hallo Welt  
=> nil
```

Rückgabewert

Ausgabe des Programms

```
2.5.1 :003 >
```




Die Programmiersprache Ruby