

2.7: Klassen

Truthy und falsy

In Ruby können alle Objekte als Wahrheitswerte verwendet werden, z.B. in Bedingungen von Kontrollstrukturen oder Schleifen.

Dabei werden alle Objekte außer `nil` und `false` als wahr behandelt - in Ruby nennt man diese Werte **truthy**. `nil` und `false` werden als **falsy** bezeichnet.

Relevant ist dieser Unterschied vor allem in drei Fällen:

- Kontrollstrukturen:

```
if obj.method?
  # Ausgeführt, wenn #method? nicht nil oder false zurückgibt
end
```

- Blöcke, die Wahrheitswerte zurückgeben:

```
[1, false, nil, Object.new, true, :symbol].select do |object|
  object
end # => [1, #<Object:0x...>, true, :symbol]
```

- Negation (mit `!` oder `not`):

```
not Object.new # => false
not 0          # => false
not nil        # => true
not false      # => true
```

Kurzform für einfache Blöcke

Für Blöcke, die einfach nur auf ihrem ersten Parameter eine Methode aufrufen, gibt es in Ruby eine Kurzform:

```
["a", "aa", "aaa"].map { |str| str.length } # => [1, 2, 3]
["a", "aa", "aaa"].map(&:length)           # => [1, 2, 3]
```

Dabei wird das Symbol `:length` mit dem `&`-Operator in einen Block umgewandelt und als Block-Argument an die Methode `map` übergeben.