



## Rubys Core-Klassen

# Von Zahlen, Zeichen und Wahrheiten

---

- Ganze Zahlen: Integer
- Fließkommazahlen: Float
- Listen: Array
- Reihen: Range
- Zeichenketten: String
- Symbole/Atoms: Symbol
- Dictionary/Map: Hash
- Wahrheitswerte: true, false
- "Nichts": nil

# Core-Klassen: Zahlen

---

Ganze Zahlen und Fließkommazahlen:

```
42, 13.37
```

Ganzzahldivision und Fließkommazahlen:

```
5 / 2 # => 2  
5.0 / 2 # => 2.5  
5 / 2.0 # => 2.5
```

<https://ruby-doc.org/core-2.5.1/Integer.html>  
<https://ruby-doc.org/core-2.5.1/Float.html>

# Core-Klassen: Listen

Liste mit verschiedenen Dingen:

```
things = [1, 2, "three", nil]
```

Zugriff auf bestimmtes Element der Liste:

```
things[0] # => 1  
things[2] # => "three"
```

Element anhängen und verändern:

```
things << 5  
things[0] = "First!"  
things # => ["First!", 2, "three", nil, 5]
```

# Core-Klassen: Listen

Eine Teilliste mit drei Elementen ab Index zwei auslesen:

```
things = [1, 2, 3, 4, 5, 6, 7]  
things[2, 3] # => [3, 4, 5]
```

Oder mit einer Reihe:

```
things[2..4] # => [3, 4, 5]
```

Oder Teile verändern:

```
things[2..4] = ["a", "b"]  
things # => [1, 2, "a", "b", 6, 7]
```

# Core-Klassen: Zeichen

## Zeichenketten und Einsetzungen (Interpolation):

```
first_name = "Franz"  
greeting = "Hallo #{first_name}!"  
# => "Hallo Franz!"
```

## Zeichenketten anhängen:

```
greeting << " Hallo Jan!"  
greeting  
# => "Hallo Franz! Hallo Jan!"
```

# Core-Klassen: Zeichen

Zeichenketten verhalten sich wie Listen von einzelnen Zeichen:

```
sentence = "Hallo Welt aus Ruby!"  
sentence[0] # => "H"  
sentence[0..9] # => "Hallo Welt"  
  
sentence[1] = "e"  
sentence[6..13] = "world from"  
sentence  
# => "Hello world from Ruby!"
```

# Core-Klassen: Symbole

Ähnlich zu einer Zeichenkette, aber unveränderlich:

```
:my_symbol  
# => :my_symbol
```



Symbole werden oft für interne Bezeichner verwendet, z.B. für Argumente oder Schlüssel in einem Hash.



# Core-Klassen: Hash

## Verzeichnis mit Schlüssel-Wert-Paaren:

```
languages = { :de => "Deutsch", :en => "Englisch" }  
languages[:de] # => "Deutsch"  
  
languages[:fr] = "Französisch"  
languages  
# => {:de=>"Deutsch", :en=>"Englisch", :fr=>"Französisch"}
```

## Kurzform mit Symbolen als Schlüssel:

```
{ de: "Deutsch", en: "Englisch" }
```

# Core-Klassen: Hash

Schlüssel und Werte ausgeben:

```
languages.each do |code, language|  
  puts "#{language} (Code: #{code})"  
end  
# Deutsch (Code: de)  
# Englisch (Code: en)  
# Französisch (Code: fr)
```

Viele Methoden von Listen auch für Hash-Objekte verfügbar:

```
languages.select do |code, lang|  
  code == :de || code == :en  
end  
# => {:de=>"Deutsch", :en=>"Englisch"}
```

# Core-Klassen: Hash

Schlüssel und Werte können beliebige Objekte sein:

```
things = {  
  0          => 'A number',  
  'letters'  => 'A string',  
  Object.new => 'An object'  
}  
  
things.each { |key, value| puts key.class }  
# Integer  
# String  
# Object
```



## Rubys Core-Klassen