Daten in Ruby

Wichtige Elementare Daten

Der Nil-Wert(Typ Nil)

```
nil
```

Logische Wahrheitswerte(Typ Bool)

```
true
false
```

Integerzahlen (Typ Int)

```
0
1
123
-1
-123
```

Gleitkommazahlen (Typ Float)

```
0.0
1.0
123.0
-1.0
-123.0
```

Texte, Namen, Intervalle

Texte (Typ String)

```
"" ''
```

```
"HAW Hamburg", 'HAW Hamburg'
```

Namen (Typ Symbol)

```
:"HAW Hamburg"
```

```
:HAW
```

Integer-Intervalle (Typ IntRange)

```
0..9
```

```
-5..-1
```

Zusammengesetzte Daten

Strukturen (Typ Point, Rect, ...)

```
Point[0,0]

Rect[0..3,2..7]
```

Sequenzen (Typ Seq, List, Array)

```
List[]

List[1,2,3]

List[3,2,1]

List[1,2,1,3]

List[1,"cat",3.14]

Array[]

Array[1,2,3]

Array[3,2,1]

Array[1,2,1,3]

Array[1,"cat",3.14]

[1,"cat",3.14]
```

Spezielle zusammengesetzte Daten

- Weil man sie häufig braucht, sind sie schon fest eingebaut
- Diese Strukturen verwenden **intern** oft Arrays

Mengen (Typ Set)

```
Set[1,2,3]

Set[3,2,1]

Set["Emil","Otto"]
```

Die Reihenfolge ist unerheblich

```
(Set[1,2,3] == Set[3,2,1])      -> true
```

Duplikate sind nicht erlaubt und werden ignoriert

```
(Set[1,2,3,1,2] == Set[1,2,3])  -> true
```

```
Set[3,2,1]
```

Tabellierte Funktionen oder Abbildungen (Typ Map)

- In Java z.B. heißt der Typ Map und es gibt die Implementationen HashMap und TreeMap
- In Ruby gibt es nur eine Implementation mit dem Namen Hash
- der Name Hash impliziert eine spezielle Implementationstechnik von vielen möglichen und ist leider sehr unglücklich gewählt

```
Hash[true  => false,
     false => true]
```

```
Hash[[false,false] => false,
     [true,false]  => false,
     [false,true]  => false,
     [true,true]   => true]]
```

```
{true => false,
 false => true}
```

```
{[false,false] => false,
 [true,false]  => false,
 [false,true]  => false,
 [true,true]  => true]}
```

```
{"cello"    => "string",
 "clarinet" => "woodwind",
 "drum"     => "percussion",
 "oboe"     => "woodwind",
 "trumpet"  => "brass",
 "violin"   => "string"}
```

Wichtige vordefinierte Standardklassen

```
Object

  NilClass
  TrueClass
  FalseClass

  String
  Symbol

  Numeric
    Float
    Integer
      Bignum
      Fixnum

  Range
  List
  Array
  Hash

  Module
    Class
  Method
  Proc
```

Legende

C = Comparable

E = Enumerable

I = Indexable

Typen, Klassen, Typprädikate etc.

Spec Type	*Ruby Class*	*Ruby Pred*	*C*	*E*	*I*	*+*	*Conversion*
		Basic Types					
Any, Obj	Object	any?, obj?					
Nil	Nil	nil?					
Bool	---	bool?					
		Number Types					
Num	Numeric	num?	C			+	to_i, to_f
Int	Integer	int?	C			+	to_i, to_f
Nat	---	nat?	C			+	to_i, to_f
Float	Float	float?	C			+	to_i, to_f
		Text Types					
Text	---	text?			I	+	to_a, to_sym
String	String	string?	C		I	+	to_a, to_sym
Sym	Symbol	sym?	C		I	+	to_a, to_sym, to_proc
		Sequence Types					
Seq	---	seq?		E		+	to_a, to_set
Array	Array	array?		E	I	+	to_a, to_set
List	List	list?		E		+	to_a, to_set
Range	Range	range?				+	
		Set Types					
Set	Set	set?		E		+	to_a, to_set
Hash	Hash	hash?		E	I	+	to_a, to_set
		Mixin Types					
Comp	Comparable	comp?	C				
Enum	Enumerable	enum?		E			to_a, to_set
		Program Types					
Module	Module	module?					
Class	Class	class?					
Method	Method	method?					to_proc
Proc	Proc	proc?					to_proc
Binding	Binding	binding?					

Konstruktion konkreter Datenobjekte einer Klasse

Konstruktion durch Literale (nur für wenige vordefinierte Klassen)

- 123, -7.34, "HAW" , :HAW, true, nil, …

Durch Konstruktionsoperationen (sonst)

- List[1,2,1], Set[1,2,3], …