

present

# Multiple Dispatch

---

## Erstmal *single dispatch* 😊

---

ich\_bin (generic function with 1 method)

- `function ich_bin(x::Number)`
- `"eine Zahl"`
- `end`

ich\_bin (generic function with 2 methods)

- `function ich_bin(x::String)`
- `"ein String"`
- `end`

ich\_bin (generic function with 3 methods)

- `function ich_bin(x::Integer)`
- `"eine Ganzzahl"`
- `end`

ich\_bin (generic function with 4 methods)

- `function ich_bin(x)`
- `"ein UFO"`
- `end`

"eine Ganzzahl"

- `ich\_bin(4)`

"eine Zahl"

- `ich\_bin(2.3)`

"ein String"

- `ich\_bin("hallo")`

"ein UFO"

- `ich\_bin([2,3,4])`

Und nun *multiple dispatch* mit 🦊 und 🐔

---

# Tierische Typen

```
• abstract type Animal end
```

```
• struct Fox <: Animal  
•   name::String  
•   weight::Int32  
• end
```

```
• struct Chicken <: Animal  
•   name::String  
•   weight::Int32  
• end
```

## ... die sich nicht immer vertragen

```
• #trouble(a1::Animal, a2::Animal)
```

trouble (generic function with 1 method)

```
• trouble(a1::Fox, a2::Fox) = false
```

trouble (generic function with 2 methods)

```
• trouble(a1::Chicken, a2::Chicken) = false
```

trouble (generic function with 3 methods)

```
• trouble(a1::Fox, a2::Chicken) = true
```

trouble (generic function with 4 methods)

```
• trouble(a1::Chicken, a2::Fox) = trouble(a2, a1)
```

## Wer kann mit wem?

```
anton = Fox("Anton the smart fox", 4500)
```

```
• anton = Fox("Anton the smart fox", 4500)
```

```
jim = Fox("Little Jim", 1800)
```

```
• jim = Fox("Little Jim", 1800)
```

```
jenny = Chicken("Jenny the funny chicken", 3400)
```

```
• jenny = Chicken("Jenny the funny chicken", 3400)
```

false

```
• trouble(anton, jim)
```

true

```
• trouble(anton, jenny)
```

false

- `trouble(jim, anton)`