

# Special Methods

Carl Friedrich Bolz, David Schneider  
Dynamische Programmiersprachen  
Heinrich-Heine-Universität Düsseldorf  
Sommersemester 2010

# Messages and Syntax

- ▶ in Python, some messages sends are syntactically obvious:

```
obj.message(arg1, arg2)
```

- ▶ some are not:

```
a + b
```

# Smalltalk

absolute syntactical uniformity:

```
obj a b c
```

```
a + b * c
```

```
obj a:  arg1 with:  arg2
```

# Special Methods

- ▶ map non-syntactical messages to special names
- ▶ examples seen so far: `__init__`, `__getattr__`,  
`__setattr__`

# Reading and Writing Items

- ▶ `a[b]` equivalent to `a.__getitem__(b)`
- ▶ `a[b] = c` equivalent to `a.__setitem__(b, c)`

# Arithmetic Operators

`--add--`, `--mul--`, ...

# Comparison

`--eq--`, `--lt--`, `--cmp--`, ...

# Call

`a(b, c, d)` equivalent to `a.__call__(b, c, d)`

How is the inherent circularity resolved?