Chennaipy

Feb 2015

Hy!


Shrayas
Logic Soft

A

Lisp powered

Python

A

Python powered

Lisp

Lisp

Lisp?


John McCarthy
1958

Polish Notation

1+1

INFIX

1+1

+ 1 1

POLISH

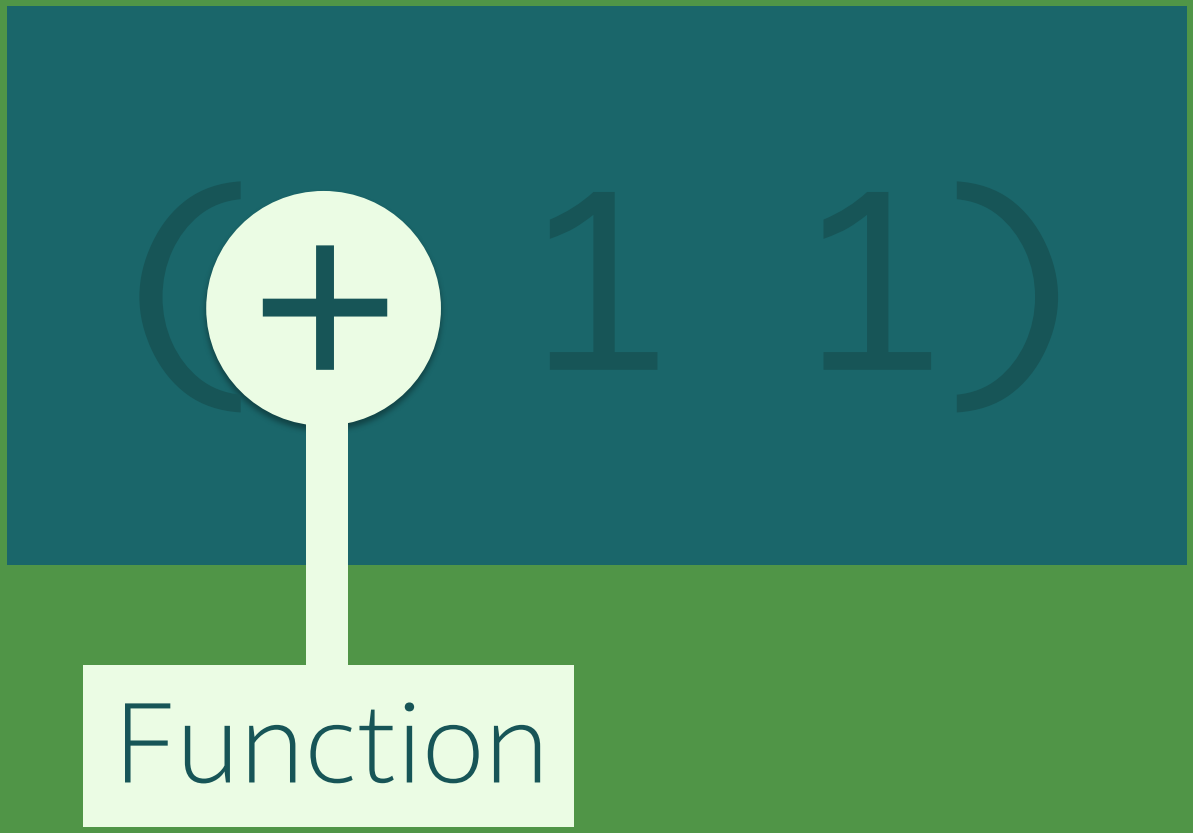
+ 1 1

```
(+ 1 1)
```

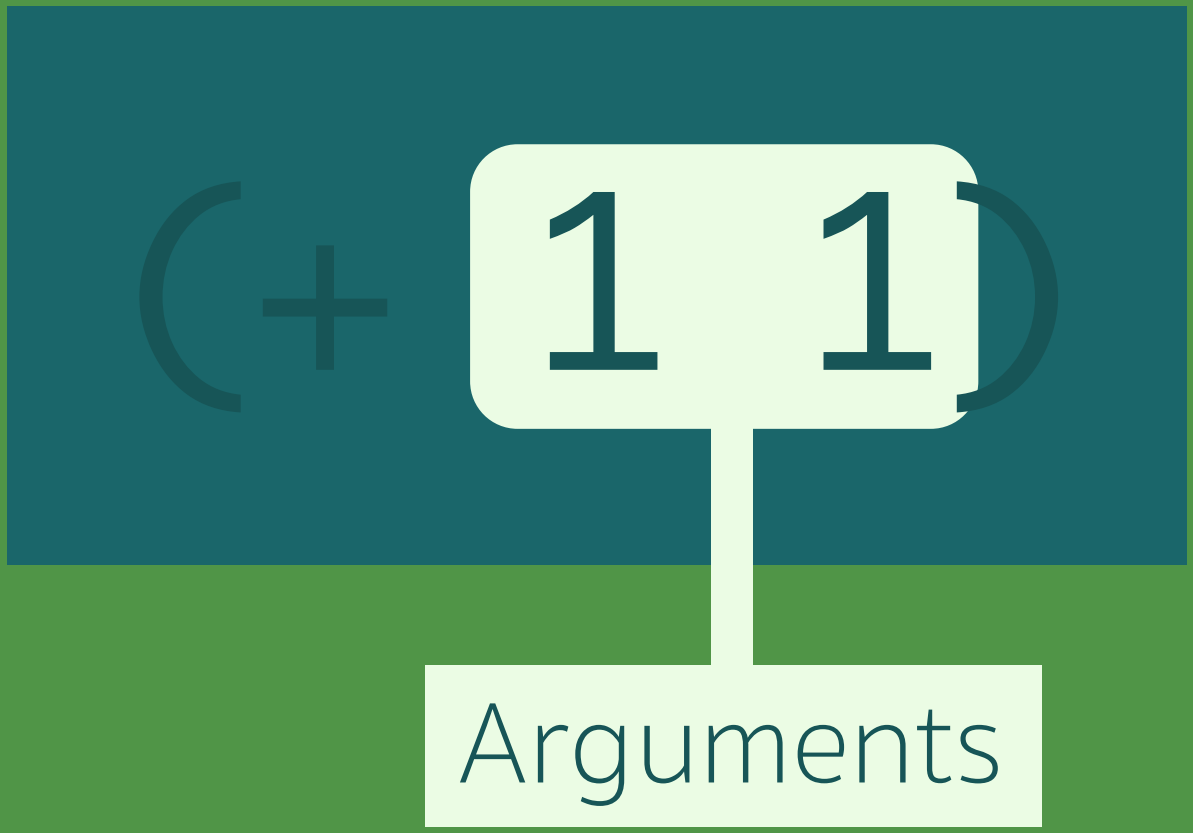
S-EXPRESSIONS

```
(+ 1 1)
```

S-EXPRESSIONS



S-EXPRESSIONS



Variable Arity!

Infix

```
1 + 2 + 3 + 4 + 5
```

Infix

```
1 + 2 + 3 + 4 + 5
```

S-Expressions

```
(+ 1 2 3 4 5)
```

Dialects

Dialects

Arc, AutoLISP, Clojure, Common Lisp, Emacs Lisp, EuLisp, Franz Lisp, Interlisp, ISLISP, LeLisp, LFE, Maclisp, MDL, Newlisp, NIL, Picolisp, Portable Standard Lisp, Racket, Scheme, SKILL, Spice Lisp, T, XLISP, Zetalisp

Dialects

Arc, AutoLISP, **Clojure**, **Common Lisp**, **Emacs Lisp**, EuLisp, Franz Lisp, Interlisp, ISLISP, LeLisp, LFE, Maclisp, MDL, Newlisp, NIL, Picolisp, Portable Standard Lisp, **Racket**, **Scheme**, SKILL, Spice Lisp, T, XLISP, Zetalisp

hylang.org

@paultag et al.

Getting & Running

```
$ pip install hy
$ hy
```

Lisp

Lisp

in Python

Why?

1

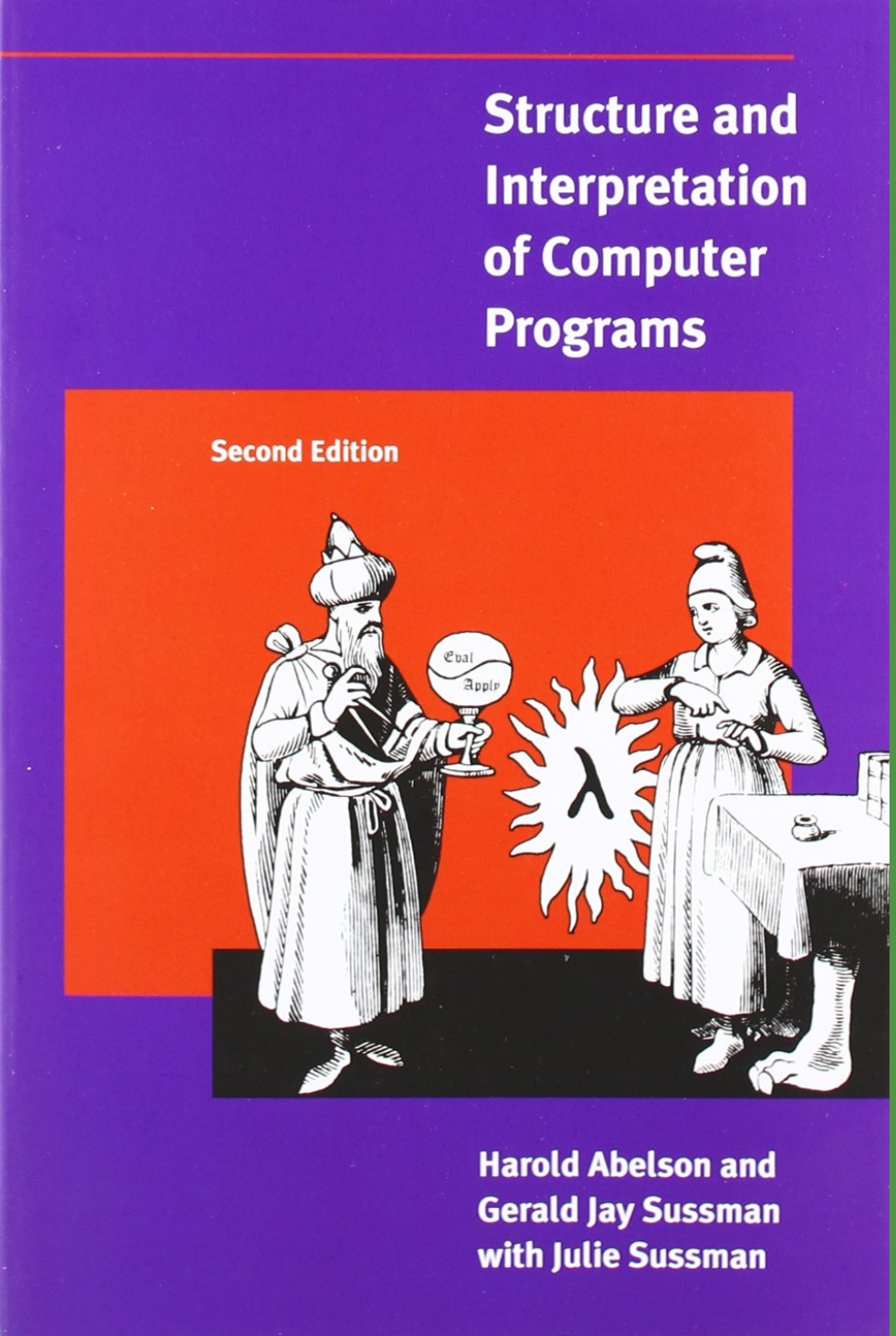
Pythonic Lisp

2

Absurd Powers!!!1

3

Learn Lisp


Structure and
Interpretation
of Computer
Programs
Second Edition
Eval
Appl
λ
Harold Abelson and
Gerald Jay Sussman
with Julie Sussman

SICP

4

Great Py Interop

Code!

Hello, python

```python
def say_hello(name):
    print "Hello", name

if __name__ == "__main__":
    say_hello("Python")
```

Hello, python

python

```python
def say_hello(name):
    print "Hello", name

if __name__ == "__main__":
    say_hello("Python")
```

hylang

```
(defn say_hello [name]
  (print "Hello" name))

(if (= __name__ "__main__")
  (say_hello "python"))
```

Data Structures

```python
>>> [1,2,3]
>>> {"foo": 1, "bar": 2, "baz": 3}
>>> (1,2,3)
```

Data Structures

python

```
>>> [1,2,3]
>>> {"foo": 1, "bar": 2, "baz": 3}
>>> (1,2,3)
```

hylang

```
=> [1 2 3]
=> {"foo" 1 "bar" 2 "baz" 3}
=> (, 1 2 3)
```

Fn on Objects

```python
>>> "Hello World".upper()
”HELLO WORLD"
```

Fn on Objects

```
>>> "Hello World".upper()
”HELLO WORLD"
```

python

```
⇒(.upper "Hello world")
"HELLO WORLD"
```

hylang

Conditionals

python

```
if (foo > 5):
    print "foo more than 5!"
else:
    print "foo less than 5!"
```

Conditionals

python

```
if (foo > 5):
    print "foo more than 5!"
else:
    print "foo less than 5!"
```

hylang

```
(if (> foo 5)
  (print "foo more than 5!")
  (print "foo less than 5!"))
```

Conditionals (more)

python

```
if (foo > 7):
  print "Too much"
elif (foo < 7):
  print "Too less"
else:
  print "ok :)"
```

Conditionals (more)

hylang

```
(cond
  [(> foo 7)
    (print "Too much")]
  [(< foo 7)
    (print "Too less")]
  [true
    (print "ok :)")]])
```

Loops

```python
for i in xrange(10):
  print i
```

Loops

```python
for i in xrange(10):
  print i
```

python

```hylang
(for [i (xrange 10)]
  (print i))
```

hylang

Hy import Py

hylang

```
(import os)

(if (not (os.path.isdir "/foo/bar"))
    (os.mkdir "/foo/bar"))
```

Interop!

```
(defn square [x]
  "returns square of x"
  (* x x))
```

hylang
(hymath.hy)

Interop!

hylang
(hymath.hy)

```
(defn square [x]
  "returns square of x"
  (* x x))
```

python
(usemath.py)

```
import hy
import hymath

print hymath.square(8)
```

Recap

Recap

1. hylang is a pythonic lisp

Recap

1. hylang is a pythonic lisp
2. Has great interop with python

Recap

1. hylang is a pythonic lisp
2. Has great interop with python
3. Great excuse to learn Lisp

Recap

1. hylang is a pythonic lisp
2. Has great interop with python
3. Great excuse to learn Lisp
4. A new language

Fin

@shrayasr