SmallTalk

Giovanni Carvelli
Stephen Kim

SmallTalk History

Emerged from ARPA funded research

Created based on a bet

Software powering the Dynabook

An educational programming language, easy readability

What is SmallTalk?

Lots of “SmallTalk” languages

- SmallTalk-71
- SmallTalk-80 (most widely used iteration)
- GNU SmallTalk (modern implementation of SmallTalk-80)
- Squeak - free and open source SmallTalk programming system

Influenced By

- Simula

Influenced

- Virtually every OO language that came after it.
- AppleScript, CLOS, Java, Python, Objective-C

SmallTalk Language: Intro

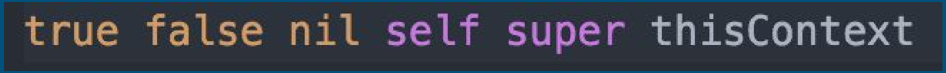
Dynamically Typed, Object-Oriented

Interpreted and Compiled

SmallTalk Language: Basics

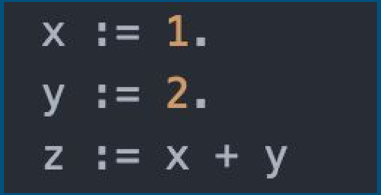
Everything’s an object (even the code itself!)

Only six reserved keywords



```
true false nil self super thisContext
```

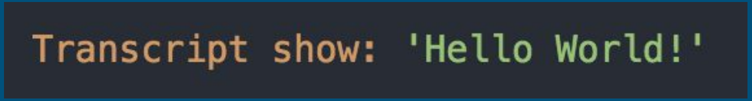
Period separated (not delimited)



```
x := 1.
y := 2.
z := x + y
```

SmallTalk Language: Basics

All operations are done with messages

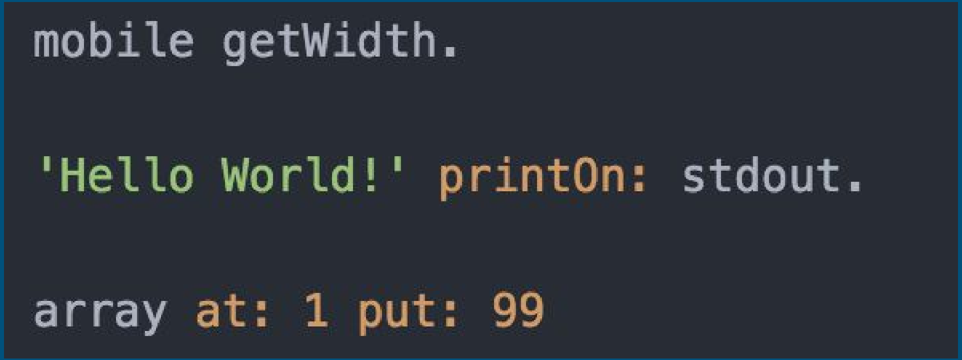


```
Transcript show: 'Hello World!'
```

SmallTalk Language: Messages

“Messages” instead of Functions

Messages can take parameters



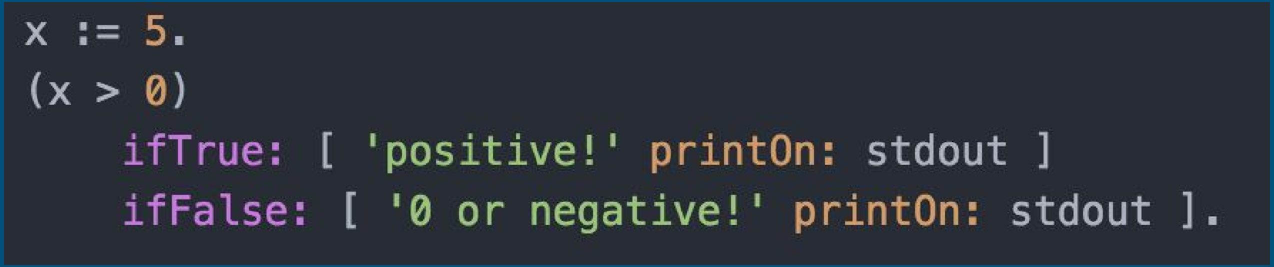
```
mobile getWidth.

'Hello World!' printOn: stdout.

array at: 1 put: 99
```

SmallTalk Language: Messages

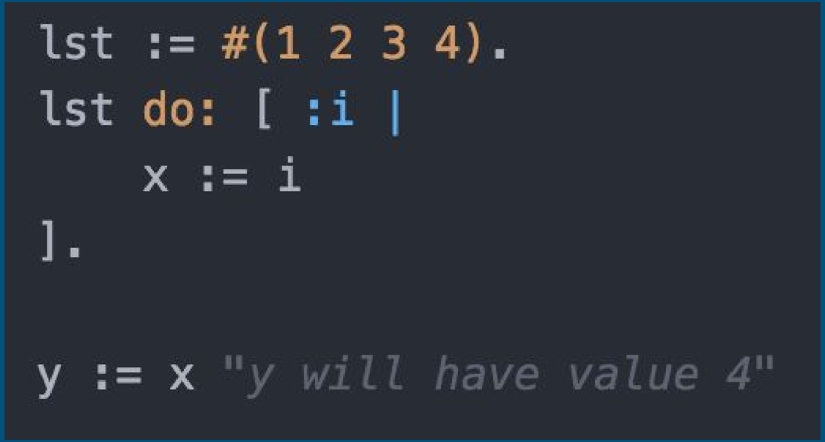
Control flow is done with messages



```
x := 5.
(x > 0)
    ifTrue: [ 'positive!' printOn: stdout ]
    ifFalse: [ '0 or negative!' printOn: stdout ].
```

SmallTalk Language: Scope

Dynamic most of the time

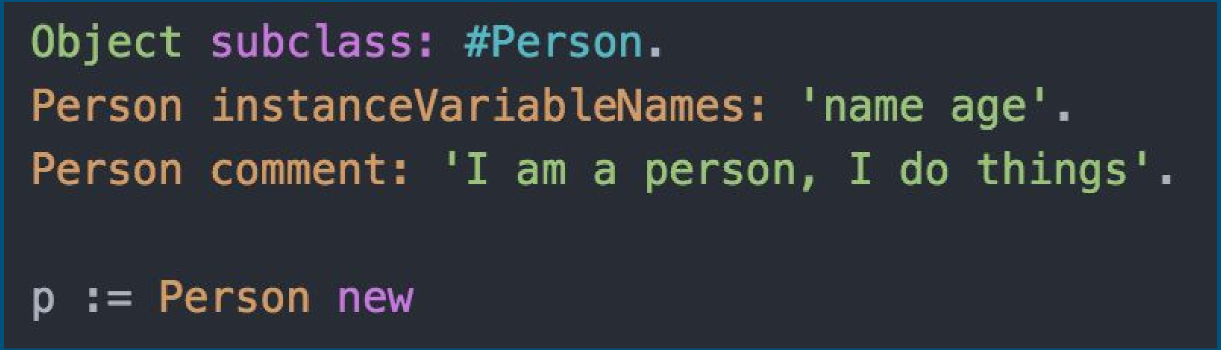


```
lst := #(1 2 3 4).
lst do: [ :i |
    x := i
].

y := x "y will have value 4"
```

SmallTalk Language: Classes

Classes can be defined with messages

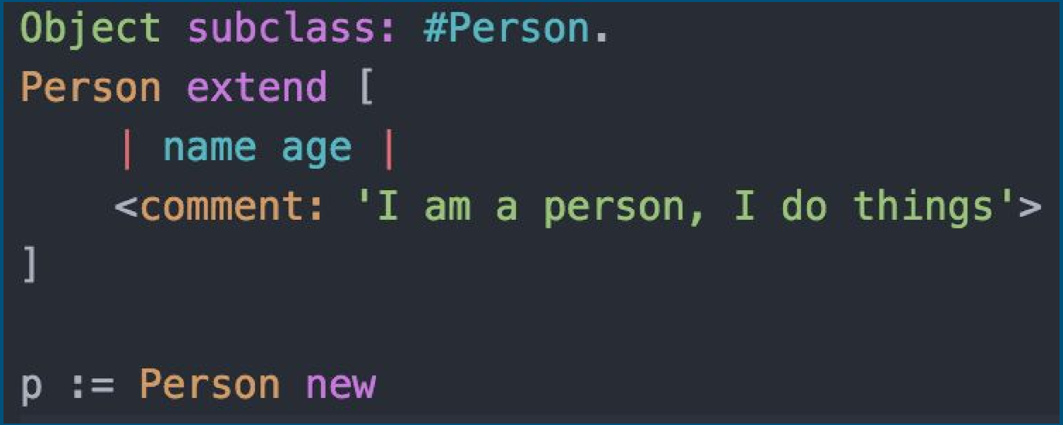


```
Object subclass: #Person.
Person instanceVariableNames: 'name age'.
Person comment: 'I am a person, I do things'.

p := Person new
```

SmallTalk Language: Classes

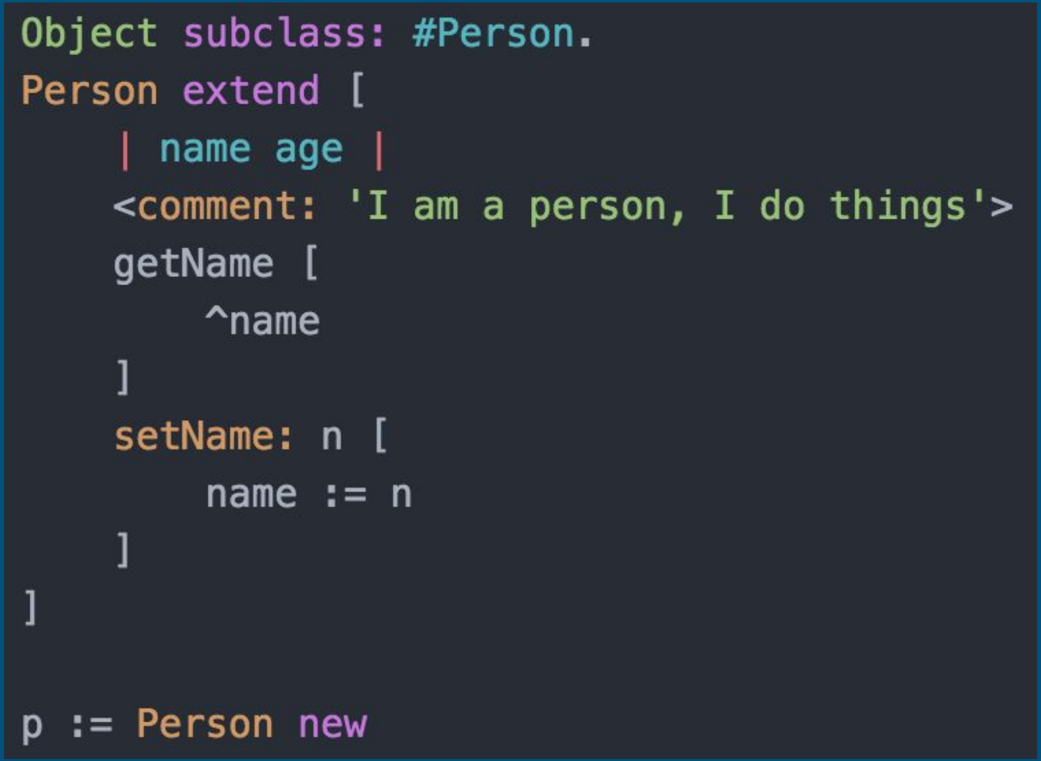
GNU SmallTalk offers a better option



```
Object subclass: #Person.
Person extend [
    | name age |
    <comment: 'I am a person, I do things'>
]

p := Person new
```

SmallTalk Language: Methods



```
Object subclass: #Person.
Person extend [
    | name age |
    <comment: 'I am a person, I do things'>
    getName [
        ^name
    ]
    setName: n [
        name := n
    ]
]

p := Person new
```

SmallTalk Language: Class Initialization

And demo!

SmallTalk: Applications

Originally created to make a computer that was entirely OOP.

As an OS

Education

SmallTalk: Anti-applications

Did not catch on because it was more like an operating system at the time. This meant it was competing with Apple, Microsoft, IBM, etc for market share.

At the time of deployment, it had a memory footprint of a couple megabytes. This was a deal breaker because memory in computers of the time were measured in kilobytes

SmallTalk: Impressions

Pure OO language

Fun to write

Q/A

Sources

http://web.cecs.pdx.edu/~harry/musings/SmalltalkOverview.html

http://gagne.homedns.org/~tgagne/contrib/EarlyHistoryST.html

https://www.gnu.org/software/smalltalk/manual/gst.html#Invocation