

```
human> load programmer --simple
```

Loading kernel module for programmer device with
`IMPERATIVE_SYS_MODULE` (deprecated)

```
human> Take the number from 1 to 10 and square them,  
then keep the squares less than 15.
```

```
Syntax error. Before or after non-existent ';'.
```

```
human>
```

```
human> exec panda
```

```
panda>
```



HOW WOULD A PANDA PROGRAM COMPUTERS?

2016-09-17 Jonas S Karlsson
Strangeloop, St. Louis

History of Programming

- physical wires
- bits
- symbols
- computation (functions)
- text processing
- structure (procedures)
- objects
- communication, client/server
- publish/subscribe
- concurrent data flow processing

History as Languages

Electrical (physical wires), Machine code (hex),
ASM (symbols), Fortran (1954), Lisp (1957),
COBOL (1959), Algol (1958), PL/I (1964), APL
(1964), Simula (1967), Pascal (1968), C
(1969), C++ (1983), erlang (1986), perl (1987),
Haskell (1990), Java (1991), python (1991),
Javascript (1994), Ruby (1995), Rx, LINQ
(2012), golang, moztart/oz, elm (2012)

Panda - Why YAFPL?

C A R Hoare about Algol:

"Here is a language so far ahead of its time that it was not only an improvement of predecessors, but also on nearly all successors."

Criteria for success of a Programming Language



Google: computer languages and facial hair

The Future of Programming?

Star Trek: "Computer, calculate the ..."

Minority Report: Hand-waving the cloud

Matrix: Fly/submerge in the data simulation

What we have now?

- drag-drop
- auto-gen templates
- IntelliJ, Eclipse - tab-tab-tab fills in declaration
- interactive in the cloud (Khans Academy)

What do I envision?

Direct data manipulation

Don't name random stuff

Incremental programming

Touch programming on tablet

Requires

- Higher level
- Less syntax
- See what you get

Microsoft's Future (is here)

Filtering

In the following example, we use the Generate operator to create a simple observable sequence of numbers. The Generate operator has several overloads. In our example, it takes an initial state (0 in our example), a conditional function to terminate (fewer than 10 times), an iterator (+1), a result selector (a square function of the current value), and print out only those smaller than 15 using the Where and Select operators.

Microsoft RX - reactive

```
IObservable<int> seq =  
    Observable.Generate(0, i => i < 10, i => i + 1, i => i * i);  
IObservable<int> source = from n in seq  
    where n < 5  
    select n;  
source.Subscribe(x => {Console.WriteLine(x);}); // output is 0, 1, 4, 9  
Console.ReadKey();
```

C - simple program - lot'sa syntax!

Loop over the integer numbers 0 to 9, square them, write out the squares that are less than 15.

```
-- edit squares.c
void main() {
    for(int i = 0; i < 10; i++) {
        int s = i*i;
        if (s < 15) printf("%d ", s);
    }
}
-- gcc squares.c && ./a.out
```

perl, python - other syntax

```
for($i in 0..9) {  
    $s = $i * $i;  
    print $s if $s < 15;  
}
```

```
for(i in xrange(0, 9))  
    s = i*i  
    if s < 15  
        print s
```

Moore's Law of Programming?

If the number of transistors double every year, we will create verbose languages like Java and double the amount we need to type to accomplish the same thing as the year before.

The Essence

0..9

square

< 15

The Essence

0..9

square

< 15

Generate

Transform

Select

Features of the Language of the Future

- Focus on sets or streams of data
- Design above a single device
- Data flows, it's alive
- Program are services - they keep running
- Reacts to changes, new data

Panda

- No NULLS! (or nil ;)
- No lists, no cons!

- No loops
- No ifs

- No map
- No variables

Panda Demo

1. simple function call
2. chaining functions
3. how not to loop
4. how not to "if"

1. web stuff
2. fluff
3. html

Haskell

“The Panda standard library [seems to] contain unusual functions which appear contrived specifically for the number-filtering examples such as:”

```
1..10.odd.sqr.lt(50)
```

Haskell II

```
sqr = map (join (*))
```

```
od  = filter odd
```

```
It  = filter .(>)
```

“Then I could express the Panda example in Haskell as”

```
od . It 50 $ sqr [1..10]
```

Pandaz haz...

Demo

Reactive Streamed Evaluation

- Reactive
 - Concurrent
 - Streamed-evaluated
 - object-oriented functional based
 - distributed language system
-
- It avoids obsessing about types
 - math, and recursions

Expressions are cheap lightweight concurrent processes. Like Erlang, these together with global messaging forms the basis of distributed computing platform.

Computationally Complete?

Goals:

1. Simple should be simple
2. Complex should be possible

Current languages:

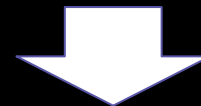
1. Simple is still complex
2. Complex is normal

Implementation

Panda



PandaLOG



Javascript

Datalog

(set a 1)

(set b 10)

(iota 1 10 i)

(odd i)

Generated JS - non-blocking

'1..10.odd.sqr'.typed.js

```
var _n14 = 1;
var _n15 = 10;
Q.push(0, "OUT__iota_bbf__integer_integer_integer",
  $PF.iota_bbf__integer_integer_integer(_n14,_n15, Q,
    function(_v193) {
      Q.push(1, "LOOP__iota_bbf__integer_integer_integer", function(_v193){return function
OUT__LOOP__iota_bbf__integer_integer_integer() {
  var _v194 = $PF.odd_bf__integer_0(_v193);
  if (_v194=== undefined||_v194=== false||_v194===null) return;
  var _v195 = $PF.sqr_bf__integer_integer(_v194);
  $PF.out_typed(out, "bf", "Any,NONE", "integer", { vars: { } }, _v195);
}}(_v193));
return true; }));
```

Generated JS - blocking

```
sum{{IOTA(1,10).odd.sqr}}'.typed.js
```

```
var _n22 = 1;
var _n23 = 10;
var _v350 = [0]; /// CCC
var _v351 = _v350;
loop1: for(var _v352 = _n22; _v352 <= _n23; _v352++){
  var _v353 = $PF.odd_bf__integer_0(_v352);
  if (_v353=== undefined||_v353=== false||_v353===null) continue loop1;
  var _v354 = $PF.sqr_bf__integer_integer(_v353);
  _v351[0] += _v354; /// DDD
}
Q.push(0, "end", function(_v350,_v351){return function OUT__end() {
  var _v356 = _v351;
  var _v357 = _v356[0]; /// EEE
  $PF.out_typed(out, "bf", "Any,NONE", "integer", { vars: { } } , _v357);
}}(_v350,_v351));
```

<<< sequences... >>>

fib

prime

UI Experiments

- <http://pandalang.org/pandagold.html>
- panda-style
- "IDE"
- Still typing...
- Turtle

Q & A

Any Pandas here?