

SWEETUGG

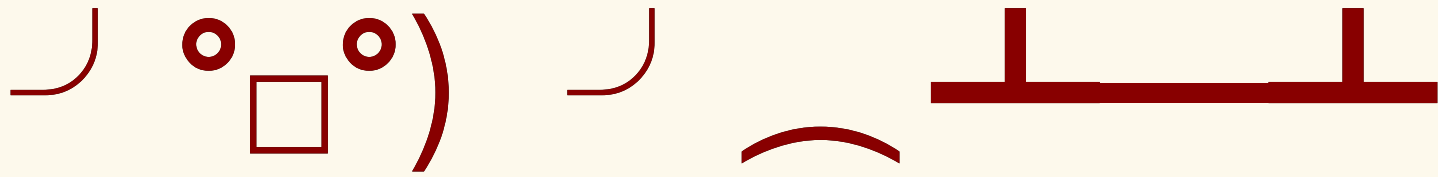
WHAT IS NEXT?

**(A FRIENDLY GUIDE TO CHOOSING YOUR
NEXT LANGUAGE)**

FEBRUARY 2018

@silverspoon

Disclaimer:



IRISH-ISMS AHEAD

CRAIC, EEJIT, ETC ARE COMMON CHILD-FRIENDLY IRISH SLANG

- Eejit – an idiot or a fool, but more often it's used in an affectionate (yet still mocking!) manner.
- Grand: average, not too bad, definitely not great.
- Cop on - common sense
- Give out - complaint

THIS IS BACON



DELPHI

VB.NET

C# (SOME JS WHEN JQUERY WAS NEW, SOME JAVA)

F# / C#

SCALA

REFERENTIAL TRANSPARENCY

LESS MUTABLE STATE!!

NO EXCEPTIONS FOR FLOW CONTROL!!

.. AND MORE

BACON'S FRIENDS FELT AWKWARD



Bacon McPig
@bacon

 Follow

Last night I wrote [#java](#) for the first time after moving to [#fp](#). OMG! I had to write so many lines of code to get something done! 😄



2



13



@silverspoon

BACON'S FP

- Typed FP
- FP everywhere
- Aspiring to purity / Total functions

@silverspoon

PROBLEMS WITH FP AS BACON UNDERSTANDS IT

- Dependency management
- Type tetris
- Complicated concepts

... is it worth it?... is it the best way?

@silverspoon

BACON DREAMS OF WELL STRUCTURED PROGRAMS

Well-structured software is easy to write and to debug, and provides a collection of modules that can be reused to reduce future programming costs. [Why FP matters. John Hughes]

@silverspoon

MEET 00000




- C++
- C#
- Works with Bacon
- Shipping is everything
- Starting to do some functional programming

"FUNCTIONAL PROGRAMMING HAS EMERGED SINCE THE MID-2000S AS AN ATTRACTIVE BASIS FOR SOFTWARE CONSTRUCTION. ONE REASON IS THE INCREASING IMPORTANCE OF PARALLELISM AND DISTRIBUTION IN COMPUTING." ODERSKY, ROMPF APRIL 2014

**"...ESPECIALLY ITS (SCALA) FOCUS ON PRAGMATIC
CHOICES THAT UNIFY TRADITIONALLY DISPARATE
PROGRAMMING-LANGUAGE PHILOSOPHIES (SUCH AS
OBJECT-ORIENTED AND FUNCTIONAL PROGRAMMING)**

ODERSKY, ROMPF APRIL 2014

A close-up photograph of a white owl's face. The owl has large, round, yellow eyes with dark pupils and a dark, pointed beak. The background is a textured, light-colored surface.

**SOLID LOOKS A LOT LIKE FP WHEN YOU
SQUINT**

FROM THE PL DESIGNERS

Scala is very much about better component oriented programming for the Java platform. Although we do a good job of object oriented programming which is very nice in F#, we haven't thought to make fundamental improvements at the component level, in a sense. We are quite happy to say "You are making components? OK, make it a .NET component". — Don Syme - March 2009

"...[Scala] focus on pragmatic choices that unify traditionally disparate programming-language philosophies (such as object-oriented and functional programming). The key lesson is these philosophies need not be contradictory in practice. — [Odersky, Rompf - April 2014]

*Regarding functional and object-oriented programming, one fundamental choice is where to define pieces of functionality (...)
...and Scala gives programmers the choice.* — [Odersky, Rompf - April 2014]

Choice also involves responsibility, and in many cases novice Scala programmers need guidance to develop an intuitive sense of how to structure programs effectively. — [Odersky, Rompf - April 2014]

A photograph of two stuffed animals, a white chick on the left and an orange bear on the right, sitting on a dark surface covered with numerous colorful Easter eggs. The eggs feature various patterns, including polka dots, stripes, and floral designs. A semi-transparent white rectangular box is centered over the image, containing text.

When Oooo and Bacon talk, they often disagree and call each other names

WE BUILD SYSTEMS WITH:

LANGUAGE(S)

TOOLS: LIBRARIES, FRAMEWORKS

CONTEXT: USERS AND COMMUNITY

BELIEFS MATTER

PARADIGMS

A PROGRAMMING PARADIGM

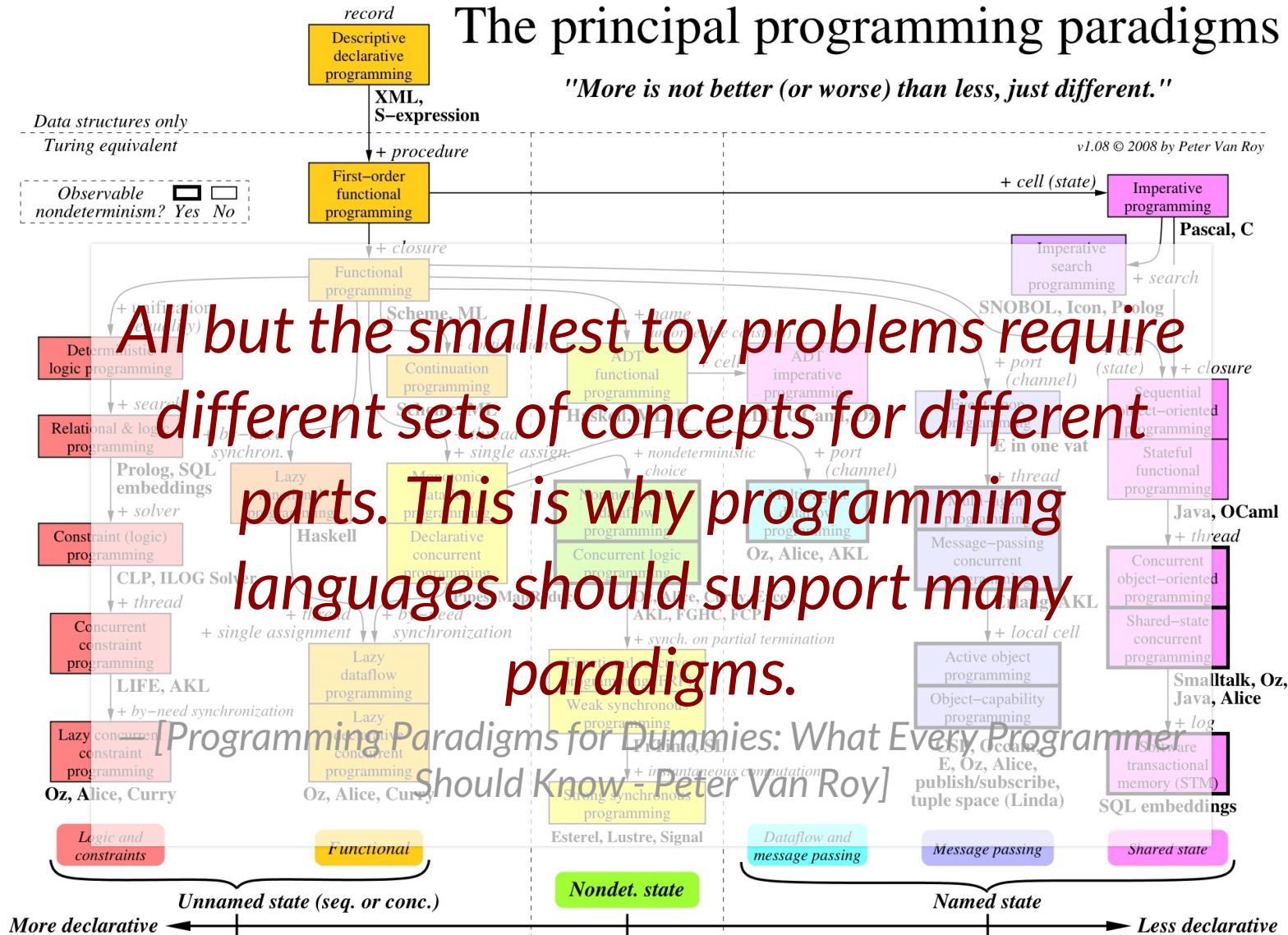
...is an approach to programming a computer based on a mathematical theory or a coherent set of principles.

— [Programming Paradigms for Dummies: What Every Programmer Should Know - Peter Van Roy]

The principal programming paradigms

"More is not better (or worse) than less, just different."

v1.08 © 2008 by Peter Van Roy




All but the smallest toy problems require different sets of concepts for different parts. This is why programming languages should support many paradigms.

— [Programming Paradigms for Dummies: What Every Programmer Should Know - Peter Van Roy]

A language should ideally support many concepts in a well-factored way, so that the programmer can choose the right concepts whenever they are needed without being encumbered by the others.

— [Programming Paradigms for Dummies: What Every Programmer Should Know - Peter Van Roy]



*...it is certainly not true that there is one
“best” paradigm*

— [Programming Paradigms for Dummies: What Every Programmer
Should Know - Peter Van Roy]

If the need for pervasive modifications manifests itself, we can take this as a sign that there is a new concept waiting to be discovered.

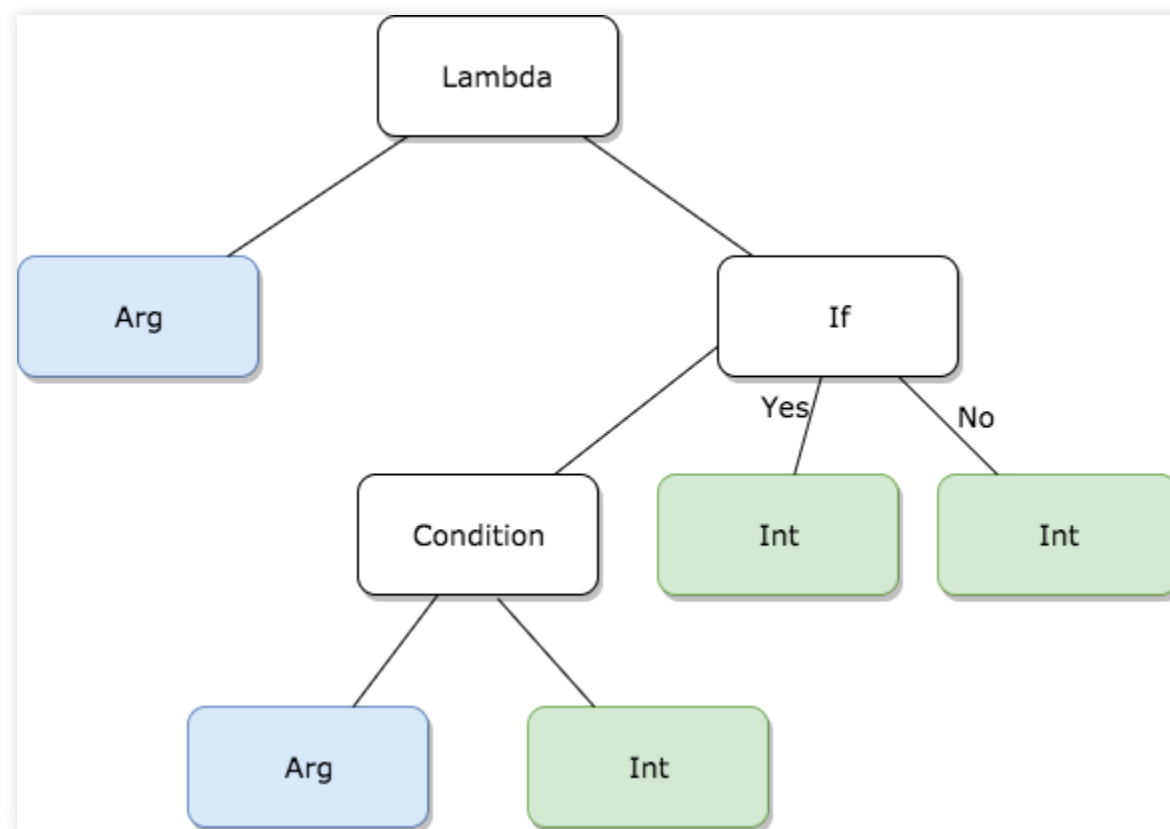
— [Programming Paradigms for Dummies: What Every Programmer Should Know - Peter Van Roy]

TYPE INFERENCE

```
1: let myfn bla =  
2:     if bla = 0 then 0 else 42
```

```
1: // int -> int
2: let myfn bla =
3:     if bla = 0 then 0 else 42
```

```
1: let simpleFuncAst = LAMBDA(ARG "bla",  
2:                        IF(EQUAL(  
3:                            ARG("bla"), INT(0)),  
4:                            INT(0),  
5:                            INT(42))  
6:
```



```
1:  (Unbound "🐱_1" === Unbound "🐱_2")
2:  (Unbound "🐱_2" === Bound "INT")
3:  (Unbound "🐱_3" === Bound "BOOL")
4:  (Unbound "🐱_4" === Bound "INT")
5:  (Unbound "🐱_4" === Unbound "🐱_6")
6:  (Unbound "🐱_5" === Bound "INT")
7:  (Unbound "🐱_5" === Unbound "🐱_6")
8:  (Unbound "🐱_7" === Unbound "🐱_1" -> Unbound "🐱_6")
```


RESULTS

```
1:      [ ( "🐱_2", "INT" );  
2:      ( "🐱_3", "BOOL" );  
3:      ( "🐱_4", "INT" );  
4:      ( "🐱_5", "INT" );  
5:      ( "🐱_1", "INT" );  
6:      ( "🐱_6", "INT" );  
7:      ( "🐱_7", "INT->INT" ) ]
```

A PARADIGM SHIFTS

"a proliferation of compelling articulations, the willingness to try anything, the expression of explicit discontent, the recourse to philosophy and to debate over fundamentals"

— Kuhn, Thomas S.. *The Structure of Scientific Revolutions*.

Many languages adding features generally associated with functional programming:

- lambdas
- functional data structures
- pattern matching, etc

C++, Java, C#

The decision to reject one paradigm is always simultaneously the decision to accept another, and the judgment leading to that decision involves the comparison of both paradigms with nature and with each other.

— Kuhn, Thomas S.. *The Structure of Scientific Revolutions*.

It is, I think, particularly in periods of acknowledged crisis that scientists have turned to philosophical analysis as a device for unlocking the riddles of their field. Scientists have not generally needed or wanted to be philosophers.

— Kuhn, Thomas S.. The Structure of Scientific Revolutions.

"... two scientific schools disagree about what is a problem and what a solution, they will inevitably talk through each other when debating the relative merits of their respective paradigms."

— Kuhn, Thomas S.. The Structure of Scientific Revolutions.

"He argued that competing paradigms are "incommensurable": that is to say, there exists no objective way of assessing their relative merits."

— Kuhn, Thomas S.. The Structure of Scientific Revolutions.

ARE WE SCIENTISTS?

Almost always the people who achieve these fundamental inventions of a new paradigm have been either very young or very new to the field whose paradigm they change .

— Kuhn, Thomas S.. The Structure of Scientific Revolutions.

A group of people, including men and women in various uniforms and civilian attire, are gathered around a long, dark wooden table in a dimly lit room. The room has a high ceiling with exposed concrete beams and several small, square skylights. The lighting is dramatic, with strong highlights and deep shadows. A large, semi-transparent white rectangular box is overlaid on the center of the image, containing the text "ALL THIS HAS HAPPENED BEFORE AND IT WILL HAPPEN AGAIN" in a bold, red, sans-serif font. The text is arranged in two lines, with "ALL THIS HAS HAPPENED BEFORE AND IT" on the top line and "WILL HAPPEN AGAIN" on the bottom line. The people are looking in various directions, some towards the camera and others away. The overall mood is serious and intense.

**ALL THIS HAS HAPPENED BEFORE AND IT
WILL HAPPEN AGAIN**



A fluffy cat with long brown and white fur is the central focus, wearing a colorful party hat. It is sitting on a white tablecloth. In front of the cat are two small, round cakes made of brown food, each on a white plate and surrounded by dry cat food. Each cake has a single lit blue candle. Several other colorful party hats are scattered around the cat on the table. The background is slightly blurred, showing a kitchen area.

**PEOPLE ARE PART OF THE CONTEXT, MAKE THEM PART OF
YOUR CONTEXT**

A black and white cat is the central focus, sitting upright and looking towards the camera. It wears a red and gold striped party hat with a yellow star on top. In its mouth, it holds a string of colorful, multi-colored beads. The cat is surrounded by a dense shower of multi-colored confetti (red, green, blue, yellow, pink) that appears to be falling from above. The background is a plain, light-colored wall. A small, dark, circular object is visible on the wall to the right of the cat. The overall scene is festive and celebratory.

STUDYING THE PAST YIELDS INTERESTING RESULTS.

The image features a sunset or sunrise scene with a warm orange and yellow glow on the horizon. In the foreground, there are dark silhouettes. On the left, a cowboy wearing a hat is riding a horse, facing right. On the right side, three cats are walking or running towards the right. The background is a clear sky transitioning from blue at the top to orange near the horizon.

**CHANGING BELIEFS IS A PERSONAL
JOURNEY.**

THANKS TO:

ROSS MCKINLAY

CHRIS MEIKLEJOHN

EDWIN BRADY

JUAN MANUEL SERRANO

TOMAS PETRICEK

AND OTHERS



THANK YOU

ANDREA MAGNORSKY

@SILVERSPoon

SOURCES | REFERENCES

PAPERS

- Programming Paradigms for Dummies: What Every Programmer Should Know - Peter Van Roy
- The paradigms of programming
- The next 700 programming languages by Peter Landin
- Why Functional Programming Matters by John Hughes
- Joe Armstrong Thesis

ARTICLES, POSTS, VIDEOS

- A punchcard ate my programme by Walt Mankowski
- Clojure spec
- Lenses in F#
- F# Don Syme
- Programming paradigm
- The expression problem

IMAGES

- Cats with hats [link](#)