Functional Programming
Lecture 13: FP in the Real World

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science
FEE, Czech Technical University in Prague

viliam.lisy@fel.cvut.cz

Mixed paradigm languages

Functional programming is great

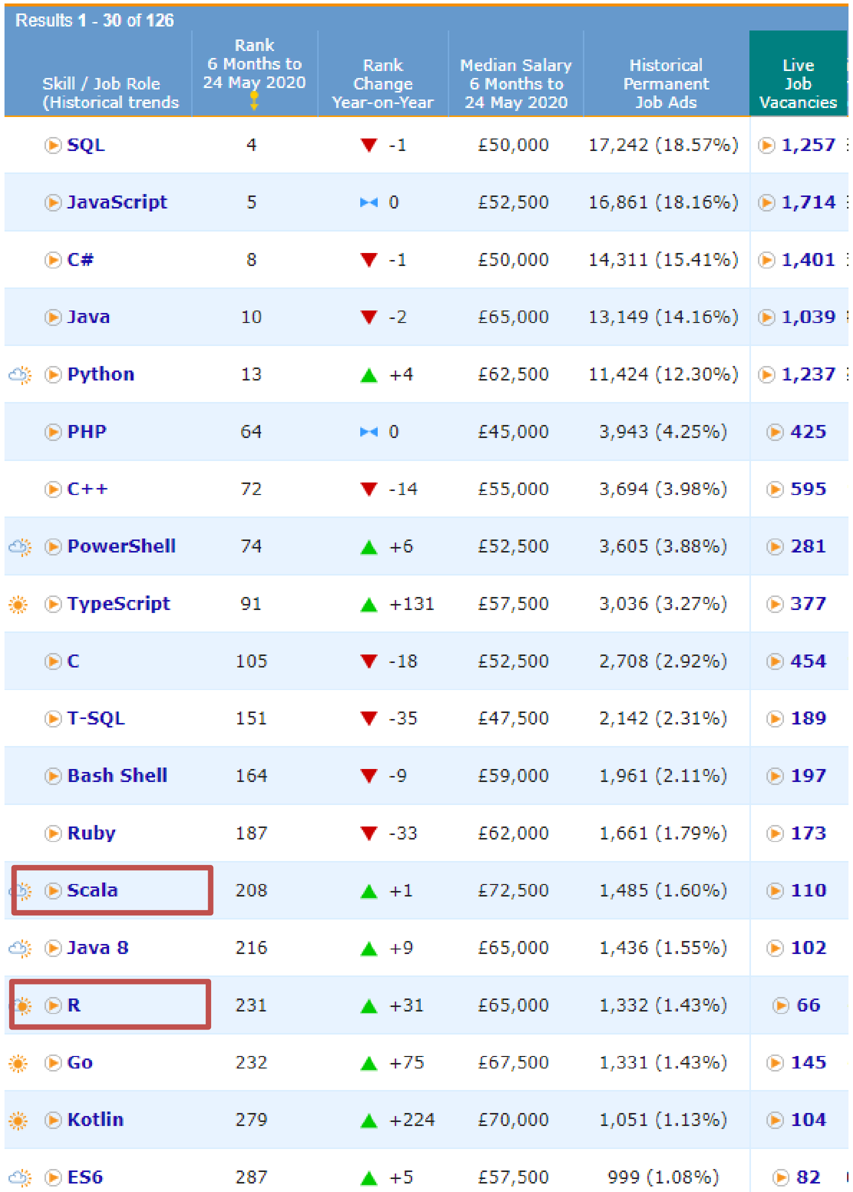
- easy parallelism and concurrency
- referential transparency, encapsulation
- compact declarative code

Imperative programming is great

- more convenient I/O
- better performance in certain tasks

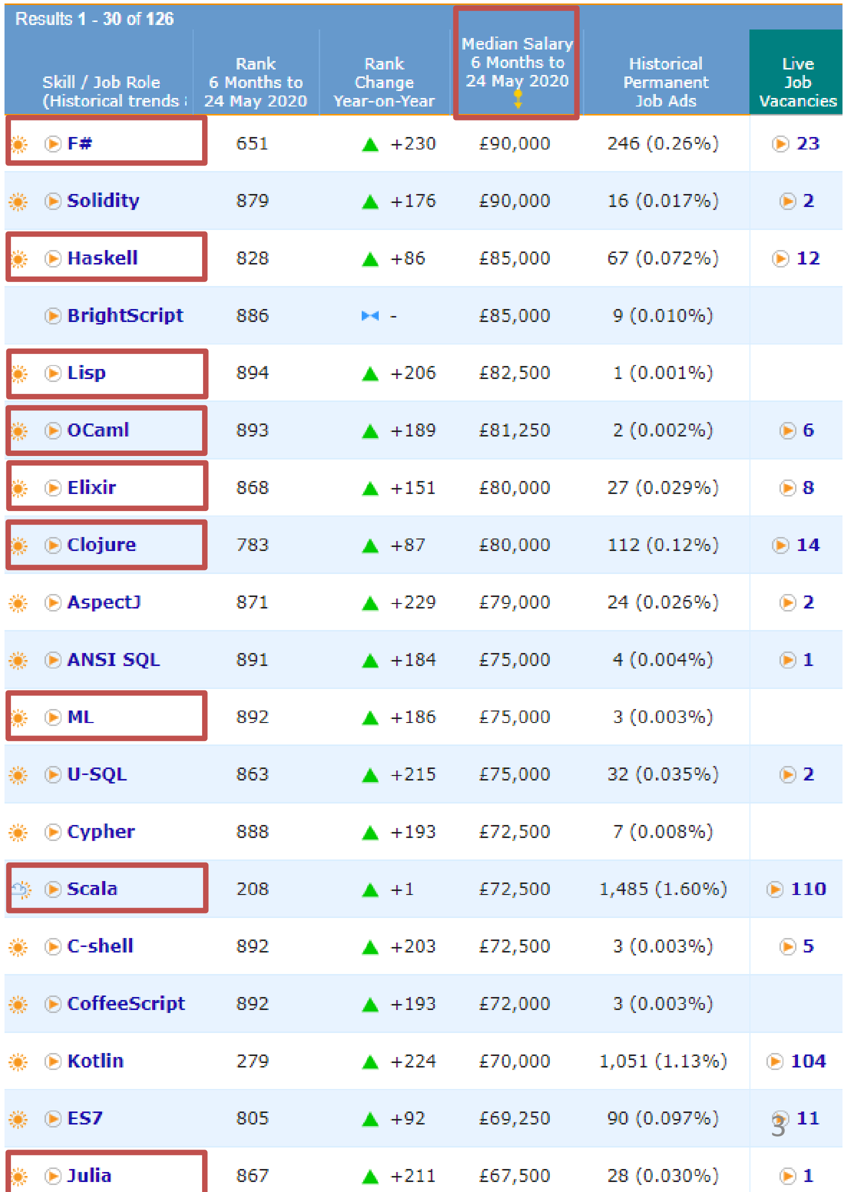
There is no reason not to combine paradigms

UK Job Market (May 2020)



Results 1 - 30 of 126

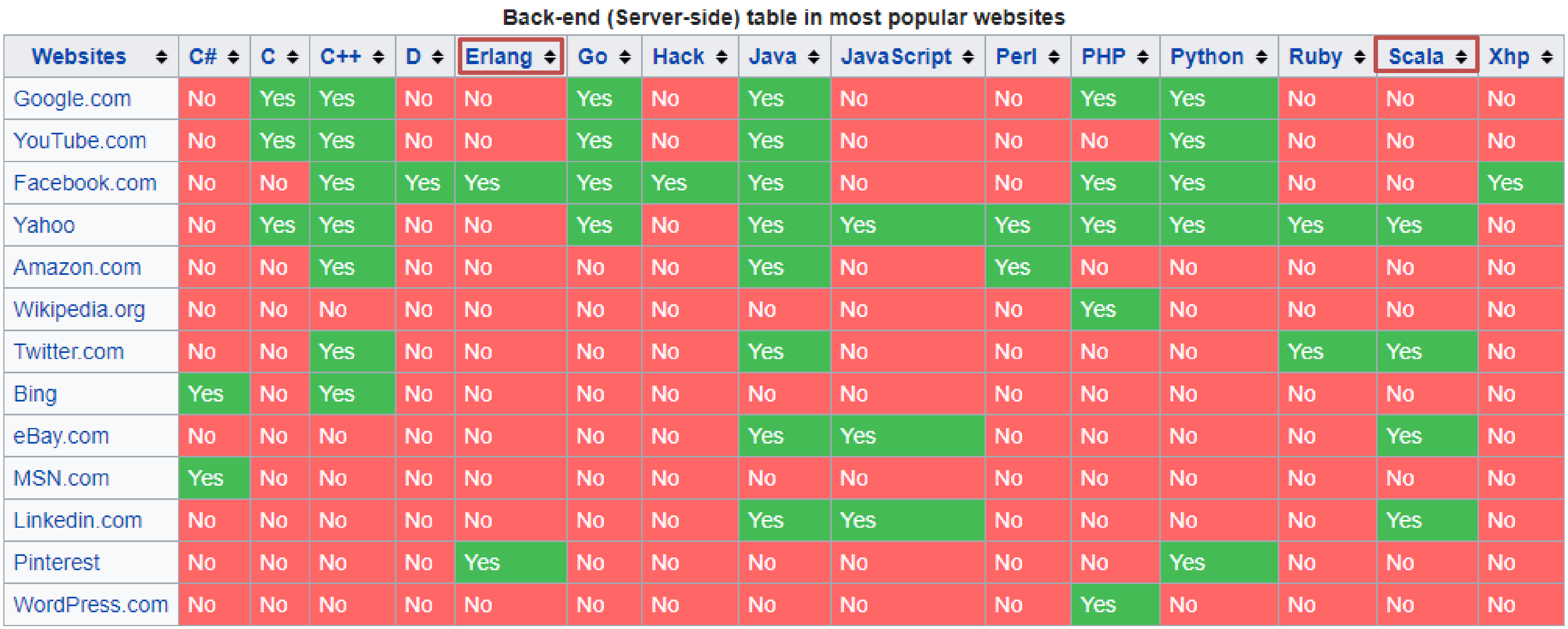
Skill / Job Role (Historical trends	Rank 6 Months to 24 May 2020	Rank Change Year-on-Year	Median Salary 6 Months to 24 May 2020	Historical Permanent Job Ads	Live Job Vacancies
SQL	4	▼ -1	£50,000	17,242 (18.57%)	1,257
JavaScript	5	⧓ 0	£52,500	16,861 (18.16%)	1,714
C#	8	▼ -1	£50,000	14,311 (15.41%)	1,401
Java	10	▼ -2	£65,000	13,149 (14.16%)	1,039
Python	13	▲ +4	£62,500	11,424 (12.30%)	1,237
PHP	64	⧓ 0	£45,000	3,943 (4.25%)	425
C++	72	▼ -14	£55,000	3,694 (3.98%)	595
PowerShell	74	▲ +6	£52,500	3,605 (3.88%)	281
TypeScript	91	▲ +131	£57,500	3,036 (3.27%)	377
C	105	▼ -18	£52,500	2,708 (2.92%)	454
T-SQL	151	▼ -35	£47,500	2,142 (2.31%)	189
Bash Shell	164	▼ -9	£59,000	1,961 (2.11%)	197
Ruby	187	▼ -33	£62,000	1,661 (1.79%)	173
Scala	208	▲ +1	£72,500	1,485 (1.60%)	110
Java 8	216	▲ +9	£65,000	1,436 (1.55%)	102
R	231	▲ +31	£65,000	1,332 (1.43%)	66
Go	232	▲ +75	£67,500	1,331 (1.43%)	145
Kotlin	279	▲ +224	£70,000	1,051 (1.13%)	104
ES6	287	▲ +5	£57,500	999 (1.08%)	82



Results 1 - 30 of 126

Skill / Job Role (Historical trends	Rank 6 Months to 24 May 2020	Rank Change Year-on-Year	Median Salary 6 Months to 24 May 2020	Historical Permanent Job Ads	Live Job Vacancies
F#	651	▲ +230	£90,000	246 (0.26%)	23
Solidity	879	▲ +176	£90,000	16 (0.017%)	2
Haskell	828	▲ +86	£85,000	67 (0.072%)	12
BrightScript	886	⧓ -	£85,000	9 (0.010%)	
Lisp	894	▲ +206	£82,500	1 (0.001%)	
OCaml	893	▲ +189	£81,250	2 (0.002%)	6
Elixir	868	▲ +151	£80,000	27 (0.029%)	8
Clojure	783	▲ +87	£80,000	112 (0.12%)	14
AspectJ	871	▲ +229	£79,000	24 (0.026%)	2
ANSI SQL	891	▲ +184	£75,000	4 (0.004%)	1
ML	892	▲ +186	£75,000	3 (0.003%)	
U-SQL	863	▲ +215	£75,000	32 (0.035%)	2
Cypher	888	▲ +193	£72,500	7 (0.008%)	
Scala	208	▲ +1	£72,500	1,485 (1.60%)	110
C-shell	892	▲ +203	£72,500	3 (0.003%)	5
CoffeeScript	892	▲ +193	£72,000	3 (0.003%)	
Kotlin	279	▲ +224	£70,000	1,051 (1.13%)	104
ES7	805	▲ +92	£69,250	90 (0.097%)	11
Julia	867	▲ +211	£67,500	28 (0.030%)	1

Most popular websites



Back-end (Server-side) table in most popular websites

Websites	C#	C	C++	D	Erlang	Go	Hack	Java	JavaScript	Perl	PHP	Python	Ruby	Scala	Xhp
Google.com	No	Yes	Yes	No	No	Yes	No	Yes	No	No	Yes	Yes	No	No	No
YouTube.com	No	Yes	Yes	No	No	Yes	No	Yes	No	No	No	Yes	No	No	No
Facebook.com	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes
Yahoo	No	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Amazon.com	No	No	Yes	No	No	No	No	Yes	No	Yes	No	No	No	No	No
Wikipedia.org	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No
Twitter.com	No	No	Yes	No	No	No	No	Yes	No	No	No	No	Yes	Yes	No
Bing	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
eBay.com	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes	No
MSN.com	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Linkedin.com	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes	No
Pinterest	No	No	No	No	Yes	No	No	No	No	No	No	Yes	No	No	No
WordPress.com	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No

Source: Wikipedia

Scala

Quite popular with industry

Multi-paradigm language

- simple parallelism/concurrency
- able to build enterprise solutions

Runs on JVM

Scala vs. Haskell

- Adam Szlachta's slides

Is Java 8 a Functional Language?

Based on:

https://jlordiales.me/2014/11/01/overview-java-8/

Functional language

- first class functions
- higher order functions
- **pure functions** (referential transparency)
- recursion
- closures
- currying and partial application

First class functions

Previously, you could pass only classes in Java

```
File[] directories = new File(".").listFiles(new FileFilter() {
    @Override
    public boolean accept(File pathname) {
      return pathname.isDirectory();
    }
});
```

Java 8 has the concept of method reference

```
File[] directories = new File(".").listFiles(File::isDirectory);
```

Lambdas

Sometimes we want a single-purpose function

```
File[] csvFiles = new File(".").listFiles(new FileFilter() {
    @Override
    public boolean accept(File pathname) {
      return pathname.getAbsolutePath().endsWith("csv");
    }
});
```

Java 8 has lambda functions for that

```
File[] csvFiles = new File(".")
  .listFiles(pathname -> pathname.getAbsolutePath().endsWith("csv"));
```

Streams

We want a list of adult users grouped by sex

```
public Map<Sex, List<User>> groupUsers(List<User> allUsers) {
  Map<Sex, List<User>> result = new HashMap<>();
  for (User user : allUsers) {
    if (user.getAge() >= 18) {
      List<User> currentUsers = result.get(user.getSex());
      if (currentUsers == null) {
          currentUsers = new ArrayList<>();
          result.put(user.getSex(),currentUsers);}
      currentUsers.add(user);
    }}
  return result;}
```

Streams

In Java 8, we can use higher order functions

```
public Map<Sex, List<User>> groupUsers(List<User> allUsers) {
  return allUsers
    .parallelStream()
    .filter(user -> user.getAge() >= 18)
    .collect(groupingBy(User::getSex));
}
```

Declarative style (and lazy)

- easier to understand
- easier to parallelize

Is Java 8 a Functional Language?

Functional language

first class functions	Yes
higher order functions	Yes
pure functions (referential transparency)	No
recursion	No tail recursion optimization by default
closures	Only values, variables become final
currying and partial application	Yes

No, but it provides many of the nice FP features

FP aspect in mainstream languages

	First class functions	Higher order functions	Lambda	Closures	List comprehensions	Referential transparency	Currying/partial application	Data immutability	Pattern matching	Lazy evaluation
Haskell	+	+	+	+	+	+	+	+	+	+
Java 8	(+)	+	+	+/-	-	-	(+)	(+)	-	(+)
C++14	+	+	+	+	-	-	(+)	(+)	(+)	(+)
Python	+	+	+	+	+	-	+	(+)	(+/-)	(+)
JavaScript	+	+	+	+	+	-	+	(+)	(+/-)	(+)
MATLAB	+	+	+	+	-	-	+	(+)	-	(+)

Erlang

Haskell – complex types + concurrency support

- Immutable data
- Pattern matching
- Functional programming
- Distributed
- Fault-tolerant

Map Reduce

Distributed parallel big data processing inspired by functional programming

- John Hughes's slides

Lisp for Scripting in SW Tools

- Emacs: extensible text editor
- AutoCAD: technical drawing software
- Gimp: gnu image manipulation program

Gimp

User scripts in: ~/.gimp-2.8/scripts

Register the function by

- script-fu-register
- script-fu-menu-register
- Filters → Script-Fu → Refresh Scripts

See example source code in a separate file.

(example form Gimp documentation)

TAKE-AWAYS FROM FP

Declarative programming

- write what should be done and leave how to the **optimizer**
  - particularly interesting in distributed setting
- easier to understand, no need to go back from how to what

Minimizing Side Effects

- reusability
- predictability
- concurrency
- lower mental load (modularity/encapsulation)

It is easier than it seems!

Immutability

You can use it in any programming language to

- ease parallelization
- avoid defensive copying
- avoid bugs in hashmaps / sets
- consistent state even with exceptions
- allows easier caching

It is not as inefficient as it seems!

Recursion

- Many problems are naturally recursive
  - easier to understand / analyze
  - less code, less bugs
  - combines well with immutability

- A great universal tool

Exam

Remote test

- recording screen, camera, mic.
- may be asked to explain the solution orally

Schedule

- 40 min test
  - anything hard to evaluate by programming
- 15 min break
- 3h of programming at computers (>50% points)
  - ~2 Haskell and ~2 Scheme tasks
  - upload system

Dates (tentative): 3.6. 9:00; …