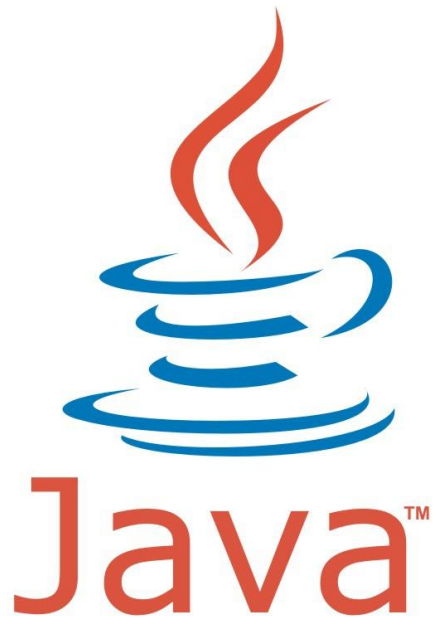


# Programación funcional en java



```
float fitnnes = 0;  
Set<String> types = ret.stream().map(e -> e. fst).collect(Collectors.toSet());  
fitnnes += types.size()/totalTypes ;  
Set<String> sizes = ret.stream().map(e -> e. snd).collect(Collectors.toSet());  
fitnnes += sizes.size()/totalSizes ;
```



**KEEP  
CALM  
BUT  
JAVA  
SUCKS**

# Programación Funcional En Python



# ¿Que es FP?

En ciencias de la computación, la programación funcional es un paradigma de programación declarativa basado en el uso de funciones matemáticas, en contraste con la programación imperativa, que enfatiza los cambios de estado mediante la mutación de variables. La programación funcional tiene sus raíces en el cálculo lambda, un sistema formal desarrollado en los años 1930 para investigar la definición de función, la aplicación de las funciones y la recursión. Muchos lenguajes de programación funcionales pueden ser vistos como elaboraciones del cálculo lambda.

# Características de FP

- Funciones son ciudadanos de primer orden
- Datos inmutables
- Funciones puras (no side effects)
- Recursividad y recursividad de cola (no loops)
- Evaluación perezosa

Iteradores, secuencias, pattern matching , etc.

# Ejemplos de Programación Funcional

Interprete de BrainFuck en Haskell

# Python lo cumple?

- Funciones ciudadanos de primer orden -> Sí
- Recursion.... -> Sí (pero leeeeeeeeeeeeeeeeeento)
- Evaluación perezosa -> Sí, pero con matices
- ....

Luego, ¿Es Python un lenguaje funcional?

NO, pero soporta un estilo de programación funcional.

# High Order Functions (Búcles fuera!)

- `map` -> aplica una función sobre una colección
- `reduce`-> genera un valor aplicando una función al par (valor\_acumulado, siguiente valor de la colección)
- `filter` -> filtra elementos en colecciones
- `comprehensions` -> `map` + `filter`



# Reduce POWER

Los otros operadores se pueden crear desde reduce

```
myMap = lambda f, l : reduce(lambda x, y: x+[f(y)], l, [])
```

```
myFilter = lambda cond, l : reduce(lambda x, y: x+[y] if cond(y) else x, l, [])
```

El problema radica en la creación de objetos en cada llamada a la función, no está optimizado en python. Aunque se puede Hackear (pero es muy feo).

# Evaluación perezosa

-En Python los Generator Expressions son perezosos: ¿Que son?

-Built-in generators :

- map
- zip
- casi todo en itertools....

# Generadores y Corrutinas

Syntax:

- (x for x in @inserta\_tu\_coleccion\_y/o\_generador\_favorit@)
- yield -> Cómo return pero guardando el estado de la función.
- yield from -> yield que delega en otro generador o iterador.

# Currying and Function composition

- $f(x, y, z) = f(x, g(y, h(z)))$
- $f(x) \leftarrow g(x) \leftarrow h(x) = f(g(h(x)))$

# Ejemplo final

## STRING MASK

### CHALLENGE DESCRIPTION:

You've got a binary code which has to be buried among words in order to unconsciously pass the cipher. Create a program that would cover the word with a binary mask. If, while covering, a letter finds itself as 1, you have to change its register to the upper one, if it's 0, leave it as it is. Words are always in lower case and in the same row with the binary mask.

### INPUT SAMPLE:

The first argument is a path to a file. Each row contains a test case with a word and a binary code separated with space, inside of it. The length of each word is equal to the length of the binary code.

For example:

```
1 hello 11001
2 world 10000
3 cba 111
```

### OUTPUT SAMPLE:

Print the encrypted words without binary code.

For example:

```
1 Hello
2 World
3 CBA
```

# Documentación de consulta

<http://www.oreilly.com/programming/free/functional-programming-python.csp>

[https://en.wikipedia.org/wiki/Lambda\\_calculus](https://en.wikipedia.org/wiki/Lambda_calculus)

<http://kachayev.github.io/talks/uapycon2012/#/49>

<https://docs.python.org/2/howto/functional.html>

<http://stackoverflow.com/questions/1017621/why-isnt-python-very-good-for-functional-programming>

<https://www.linkedin.com/pulse/python-functional-programming-geison-flores>

[https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_funcional](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional)

<http://stackoverflow.com/questions/20535342/lazy-evaluation-python>

<https://mathieularose.com/function-composition-in-python/>

¿Preguntas?

