neueda

Programming Languages

Objectives

- Compiled vs Interpreted languages
- Java & Bytecode compiled languages
- Object Oriented vs Functional Programming
- Choosing a programming language

Compiled vs Interpreted Languages

- We can categorise most programming languages as either **compiled** languages, or **interpreted** languages.

Compiled

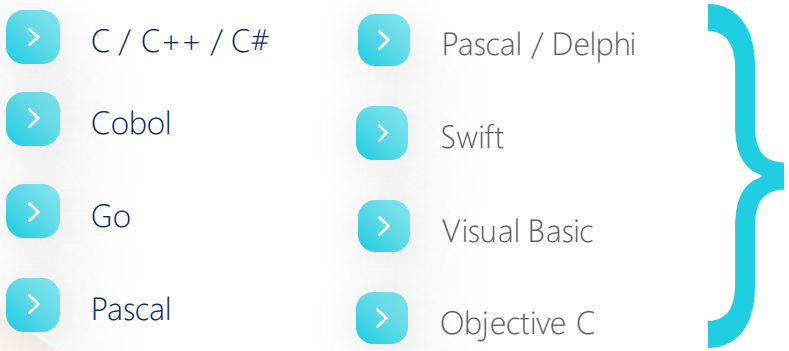
- C / C++ / C#
- Cobol
- Go
- Pascal
- Pascal / Delphi
- Swift
- Visual Basic
- Objective C

Interpreted

- JavaScript
- Perl
- PHP
- MATLAB
- Python
- R
- Visual Basic for Applications (VBA)
- Ruby

Compiled languages

- With compiled languages, we need to take the code that we write and convert it into a format ready for the operating system to be able to run, an executable file. This conversion process is called compiling.



Run faster than Interpreted

Different versions for each O/S

Interpreted languages

- With interpreted languages, we need additional software to actually run the code we write. For example Javascript code needs a browser to run, VBA needs Microsoft Office, PHP needs a web server. This additional software interprets our code / is called the interpreter.

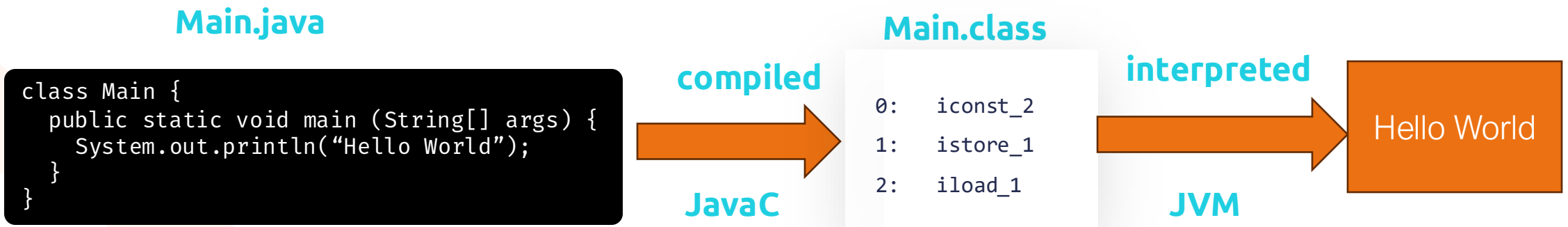
- With modern applications deployed to servers, we are no longer concerned about giving our customers the code – this wasn't the case when software was always downloaded / provided on physical media!

The Java Challenge

- The Java programming language aimed to solve a disconnect between these two approaches
- The developers of Java wanted to
  - Develop the same application but for different hardware (originally embedded into TV cable boxes)
  - Keep their intellectual property private by not giving away their source code

The Virtual Machine Approach

- The solution to the problem was to create a hybrid approach
- Java can **compile** the code, but not to a machine readable format – it is compiled to an optimised, standard structure called **bytecode**
- The bytecode needs an **interpreter** to run – this is called the **Java Virtual Machine**



Bytecode compiled languages

- Java is a bytecode compiled language.
- Other languages have since been developed that can also be compiled to bytecode and run on the JVM

Compiled

- C / C++ / C#
- Cobol
- Go
- Pascal
- Pascal / Delphi
- Swift
- Visual Basic
- Objective C

Bytecode Compiled

- Java
- Scala
- Groovy
- Clojure
- Kotlin

Interpreted

- JavaScript
- Perl
- PHP
- MATLAB
- Python
- R
- Ruby
- Visual Basic for Applications (VBA)

Object Oriented vs Functional

Object Oriented (OO) and Functional Programming (FP) are different paradigms

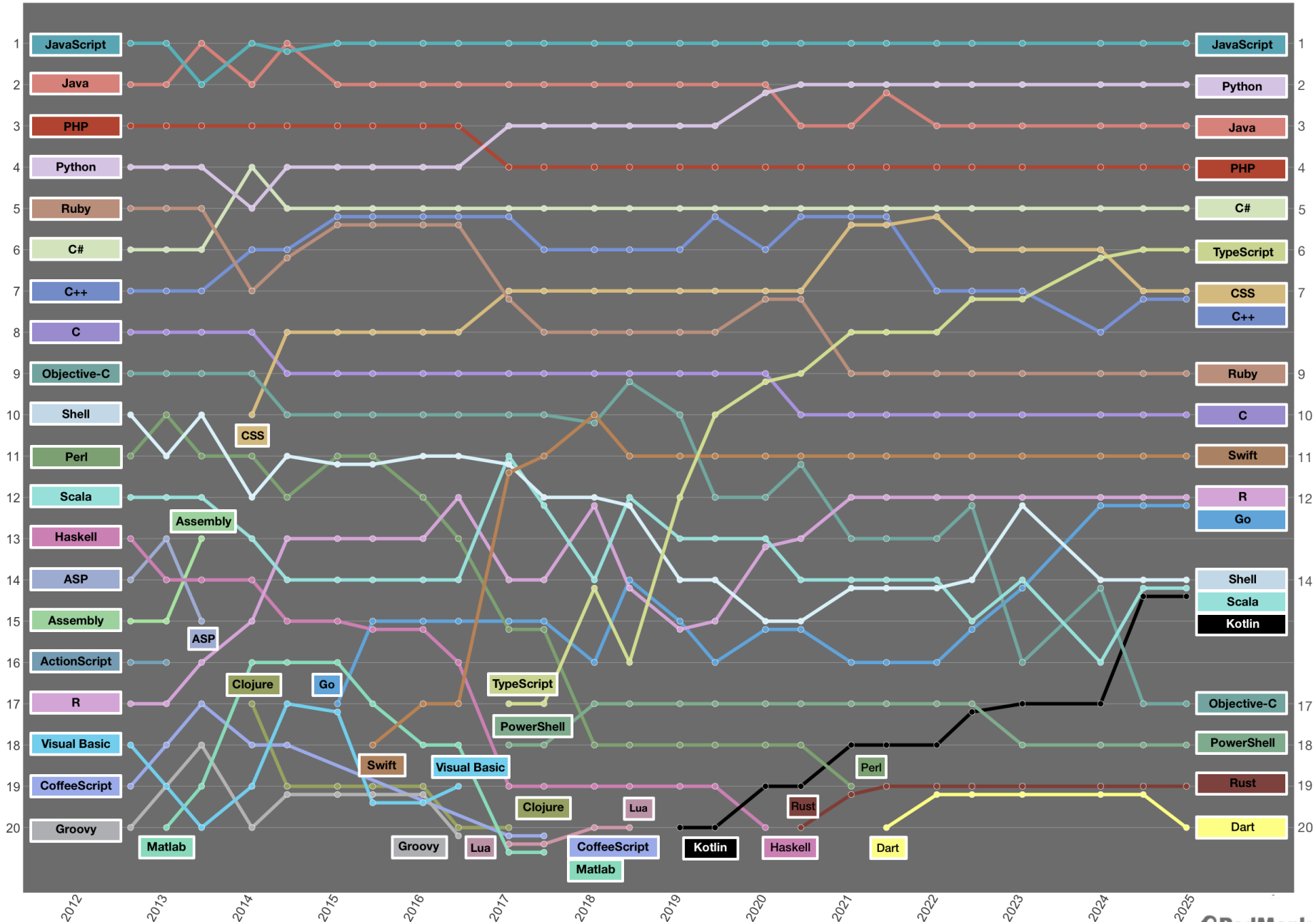
- OO languages are typically used to model real world concepts
  - Classes define objects with **properties** and **behaviors.**
  - Classes are the building blocks of the applications – we can create multiple **instances** of a class
- Functional programming is very well suited to data manipulation tasks
  - **Functions** are the building blocks of the applications
  - Data is **immutable**
- Other paradigms exist too, including prototyping, procedural and logic

Choosing a programming language

- Many tasks can be undertaken in multiple programming languages
- Factors which organizations consider when selecting a language include:
  - Speed of development
  - Suitability of the language to the task
  - Availability of staff with knowledge and skills to support and maintain

RedMonk Language Rankings

September 2012 - December 2024



neueda

Task: Choose favorite programming language

- Go to http://helloworldcollection.de/
- Find your most interesting programming language / 找到你最感兴趣的编程语言
- Discuss as a team per table / 根据表格作为一个团队进行讨论
- Present your favorite language per table / 在每张桌子上展示您最喜欢的语言
  - Name of programming language / 编程语言名称
  - Why is it your favourite? / 为什么它是你的最爱？

Summary

- Compiled vs Interpreted languages
- Java & Bytecode compiled languages
- Object Oriented vs Functional Programming
- Choosing a programming language