Javascript

Short introduction

Shadi Lahham - Programmazione web - Frontend - Javascript

Language

Javascript

WHAT EXACTLY IS JAVASCRIPT?

- JavaScript is a programming language used to make web pages interactive
- Javascript is runs **client-side** (in the visitor's browser). Client-side code is code that is run on the user's computer
- It is responsible for the "behavior" of a Website.
- It is the third layer of the layer cake of standard web technologies in addition to HTML and CSS.

Javascript

WHAT CAN JAVASCRIPT DO?

JavaScript allows you to implement complex things on web pages such as:

- Updating website content (e.g. news updates)
- Interactive maps
- Drawing and animation
- Image galleries and lightboxes
- Full featured web applications
- Keep track of users with cookies
- Interactive elements like tabs, sliders and accordions

Node.js

WHAT IS NODE.JS?

- Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications.
- Node.js runs **server-side**. Server-side code on the other hand is run on the server, then its results are downloaded and displayed in the browser.
- Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Compiled and interpreted languages

Compiled and interpreted languages

Interpretation and compilation are properties of the implementation of a language

It’s not accurate to say that a language is interpreted or compiled because interpretation and compilation are both properties of the implementation of that particular language, and not a property of the language itself.

So, in theory, any language can be compiled or interpreted – it just depends on what the particular implementation that you are using does.

What exactly is compilation?

In a compiled implementation of a language, a compiler will translate the program directly into code that is specific to the target machine, which is also known as machine code – basically code that is specific to a given processor and operating system. Then the computer will run the machine code on its own.

What exactly is interpretation?

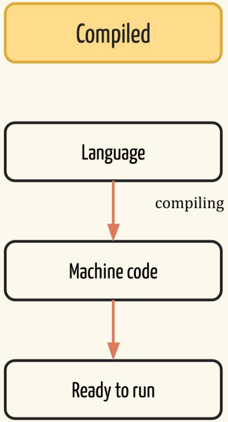
In an interpreted implementation of a language, the source code is not directly run by the target machine. What happens instead is that another program reads and then executes the original source code.

This other program is also known as the interpreter. The interpreter is usually written specifically for the native machine.

For example, consider the multiplier operation "*". If the interpreter sees this in your code, then at run time it would call its own definition of the multiplier function – maybe called "multiply(x,y)". Then that "multiply(x,y)" would execute the machine code's equivalent of the multiply instruction.

Too many words!

Compiled vs interpreted languages



Compiled

Language

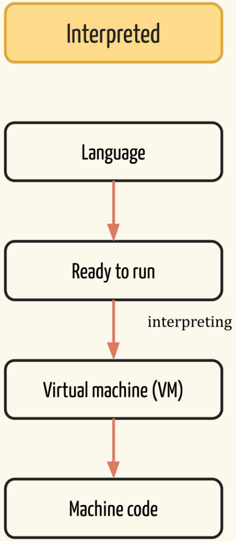
compiling

Machine code

Ready to run

Examples of compiled languages

C, C++, Fortran, Pascal



Interpreted

Language

Ready to run

interpreting

Virtual machine (VM)

Machine code

Examples of interpreted languages

Python, PHP, Ruby, Javascript

Comparison

Interpreted	**Compiled**
Requires interpreter	Requires compiler
Interpreted on the fly	Depends on platform
Platform independent	Slow compilation

Interpreted: Advantages and Disadvantages

Advantages

- Easy to learn and use
- More portable
- Allow complex tasks to be performed in relatively few steps
- Allow simple creation and editing in a variety of text editors
- Allow the addition of dynamic and interactive activities to web pages
- Editing and running of code is fast

Interpreted: Advantages and Disadvantages

Disadvantages

- Usually run quite slowly
- Limited access to low level and speed optimization code.
- Limited commands to run detailed operations on graphics.
- Limited access to the device

Compiled: Advantages and Disadvantages

Advantages

- Fast execution
- Optimised for the target hardware

Disadvantages

- Require a compiler
- Editing and deploying the code is a lot slower than interpreters

Your turn

1.Languages

- Make a list of all the programming languages that you know
- Classify the languages into the groups: compiled, interpreted, other
- For each language, explain why it is compile, interpreted or other
- Try to find additional programming languages and add them to the list

Create a folder named **01-languages**

Inside the folder create a **.txt** or **.doc** or **.md** file with your answers

Note: all files should be in kebab-case (italiano)

2.Levels

Read the following articles and write a short summary in Italian or English

Classification of Programming Languages

Levels of Programming Languages

Create a folder named **02-levels**

Inside the folder create a **.txt** or **.doc** or **.md** file with your answers

Note: all files should be in kebab-case (italiano)