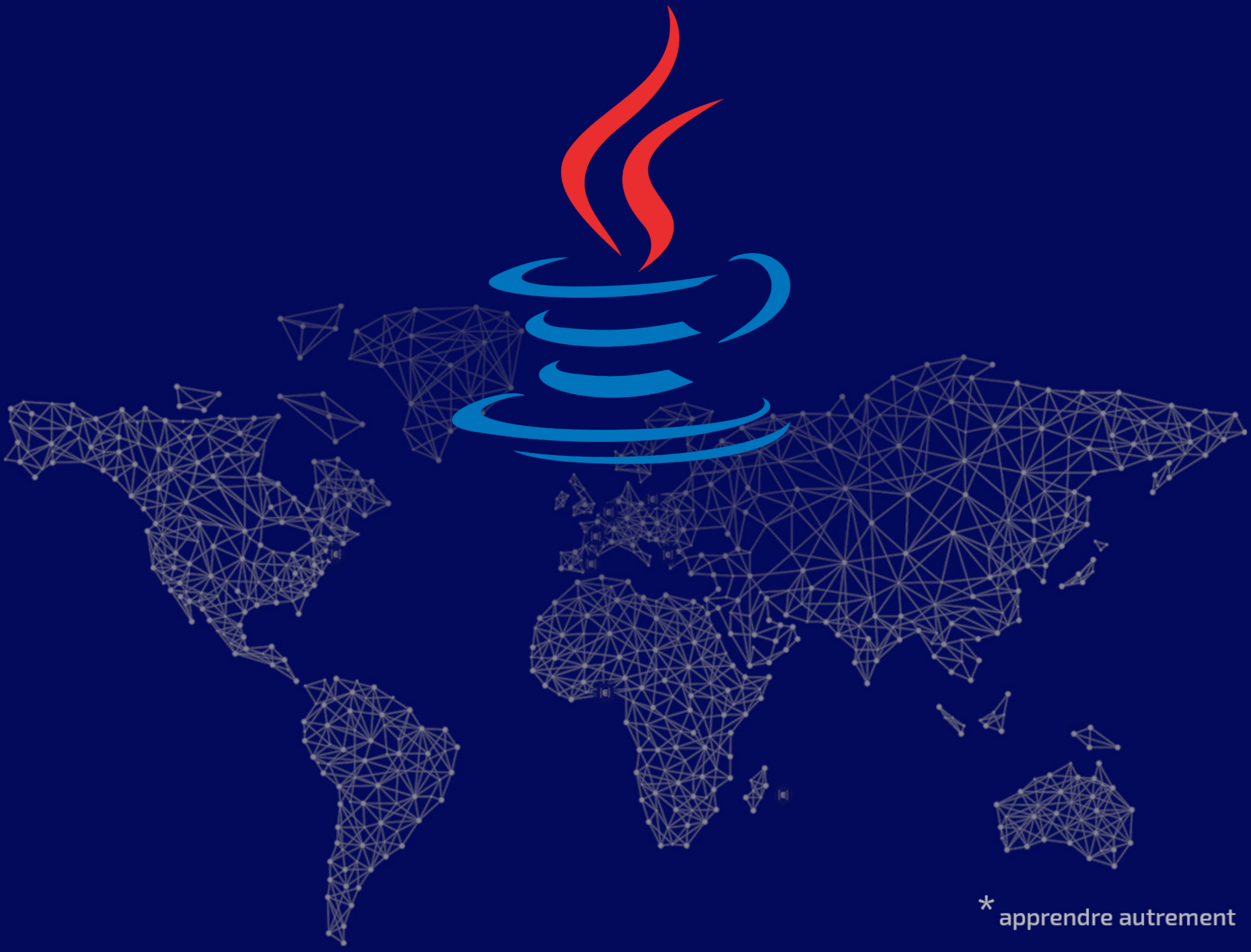
{EPITECH} LEARN DIFFERENT*

JAVA SEMINAR

DAY 01 - HELLO WORLD



*apprendre autrement

JAVA SEMINAR

To run this pool, you need Java to be installed on your computer.
We are here talking about **JSE**, that you could easily find on Internet.

In Java, every line of code must be included into a class.
There must also be one file by class (and one class by file) named the same way.
Here is a very first example of Java code (necessary in a file name "HelloWorld.java"):

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Then this code is compiled with

```
Terminal
T-JAV-500> javac HelloWorld.java
```

and executed:

```
Terminal
T-JAV-500> java HelloWorld
```

Sure this code snippet displays "Hello World!" when you execute it.

Have you seen the "main" function? Why is it special?

For all the following exercises, your functions must be included in a class named the same way as the file to turn-in.
For instance if the file to turn-in is `Ex00.java` then the class must be as follow:

```java
class Ex00 {
    public static void myMethod() {
    }
}
```

Exercise 01

Delivery: `./ex_01/Ex01.java`
Restrictions: `System.out.print` is mandatory, and must be used only once.
Prototype: `public static void myConcat(String str1, String str2);`

Create a `myConcat` function that takes two parameters and displays the first parameter followed by a space followed by the second parameter.

```
Terminal
T-JAV-500> Ex01.myConcat(‘‘Hello’’, ‘‘world’’);
Hello world
```

Exercise 02

Delivery: `./ex_02/Ex02.java`
Restrictions: the `for` keyword is mandatory
Prototype: `public static String getAngryDog(int nbr);`

Write a function that takes a `nbr` integer as parameter and returns `nbr` times the word “woof”, followed by a newline.

```
Terminal
T-JAV-500> Ex02.getAngryDog(3);
woofwoofwoof
```

Exercise 03

Delivery: `./ex_03/Ex03.java`
Restrictions: the `for` keyword is mandatory.
Prototype: `public static void printArray(ArrayList<String> myArray);`

Write a function that takes a `myArray` ArrayList as parameter and displays each of its value followed by a newline.

```
Terminal
T-JAV-500> Ex03.printArray([Volvo, BMW, Ford, Mazda]);
Volvo
BMW
Ford
Mazda
```

Exercise 04

Delivery: `./ex_04/Ex04.java`
Restrictions: the `switch` keyword is mandatory.
Prototype: `public static void printMovieFromNbr(int nbr);`

Write a function that takes an integer as parameter and displays

- ✓ `The Three Brothers` when the value is 3 ;
- ✓ `The Sixth Sense` when the value is 6 ;
- ✓ `The Number 23` when the value is 23 ;
- ✓ `28 Days Later` when the value is 28 ;
- ✓ `I don't know.` otherwise.

A new line must be printed after each display.

```
Terminal
T-JAV-500> Ex04.printMovieFromNbr(23);
The Number 23
```

Exercise 05

Delivery: `./ex_05/Ex05.java`
Prototype: `public static ArrayList<String> myGetArgs(String... var);`

Write a function that takes as parameter a variable number of arguments and returns these arguments in an array.

Exercise 06

Delivery `./ex_06/Ex06.java`
Prototype: `public static void sequence(int nbr);`

Write a function that outputs the following sequence to the nth iteration.
Here is the beginning of the sequence:

1
11
21
1211

111221
312211

The first iteration is 0.
Don't print anything when the number is negative.

```
Terminal
T-JAV-500> sequence(0);
1
```

```
Terminal
T-JAV-500> sequence(3);
1
11
21
1211
```

v 3.1

*apprendre autrement