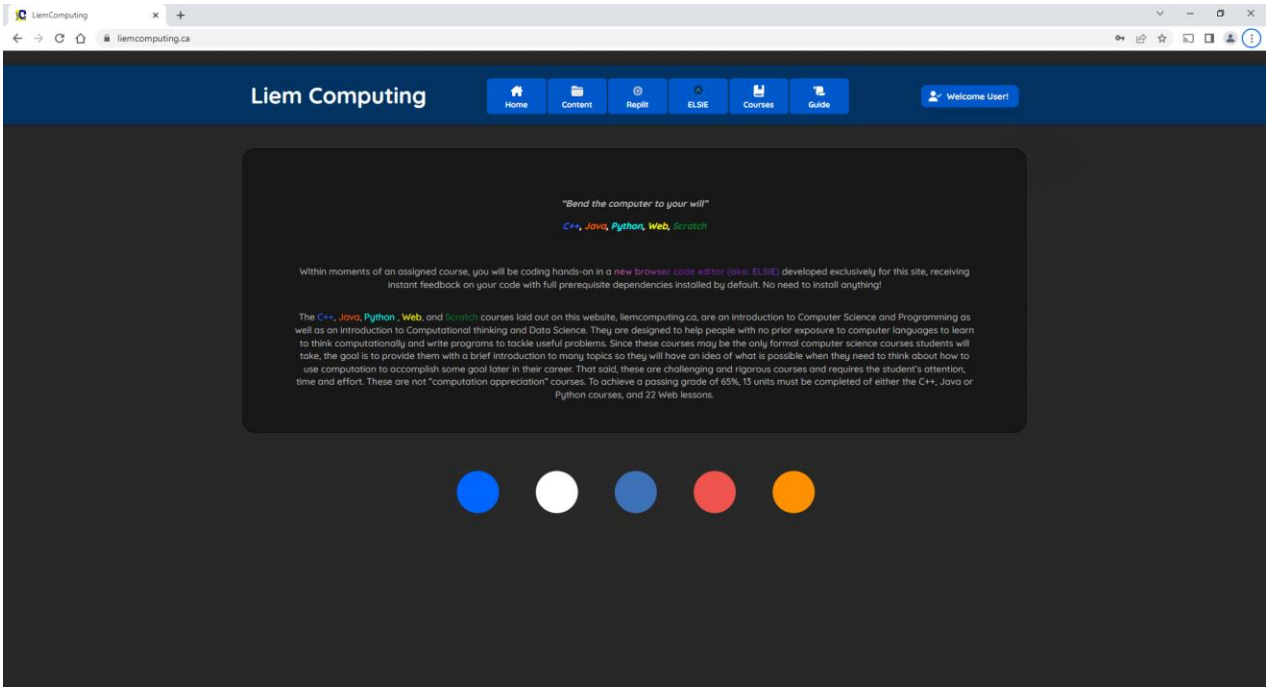
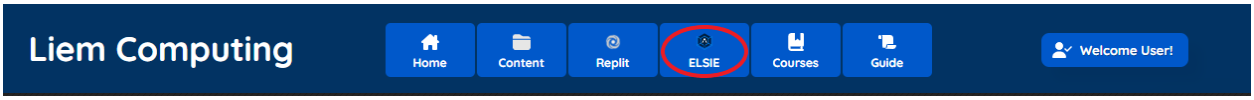
Java coding using ELSIE

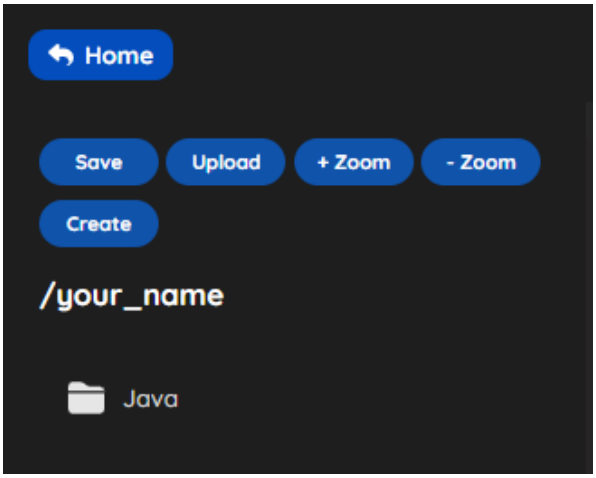
This is a getting-started lesson teaching you how to use ELSIE for the purpose of coding in Java. To begin, you'll need to open chrome and sign into liemcomputing.ca

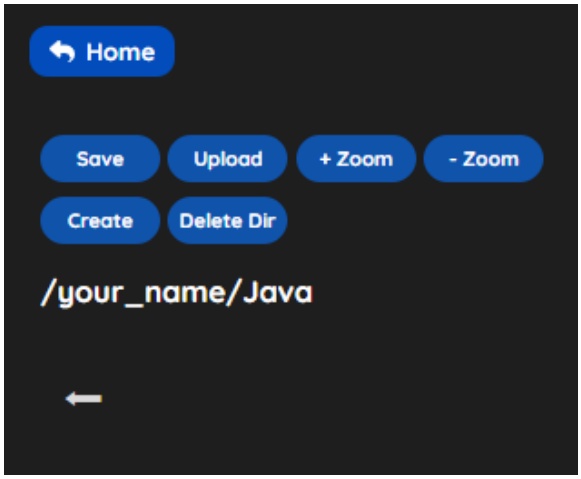


Next, click the ELSIE button found in the top toolbar near the search bar.



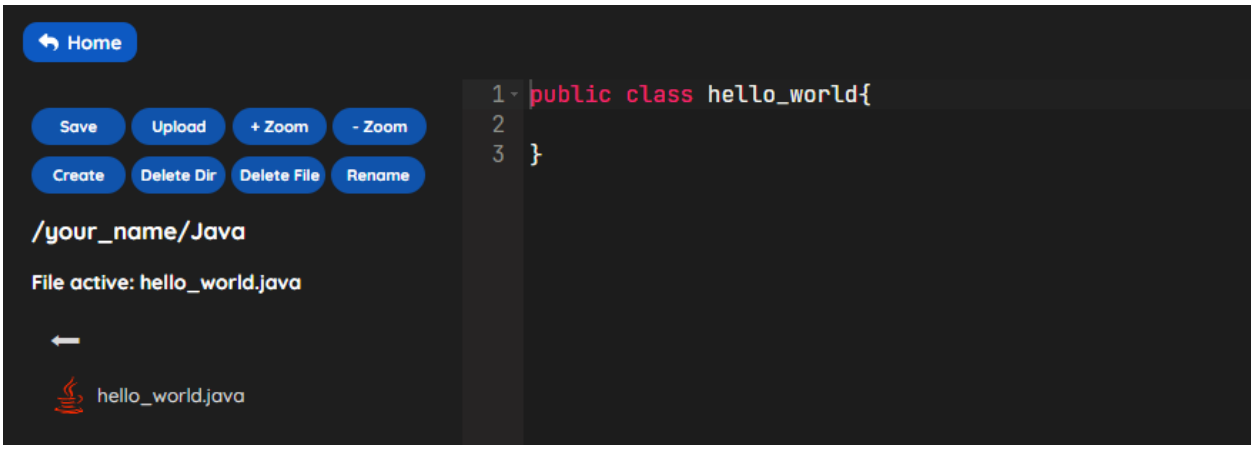
Great! Now ELSIE should have been opened. We're going to create a new folder called "Java," in which you'll store all your lessons. Once you've created it, enter the folder.





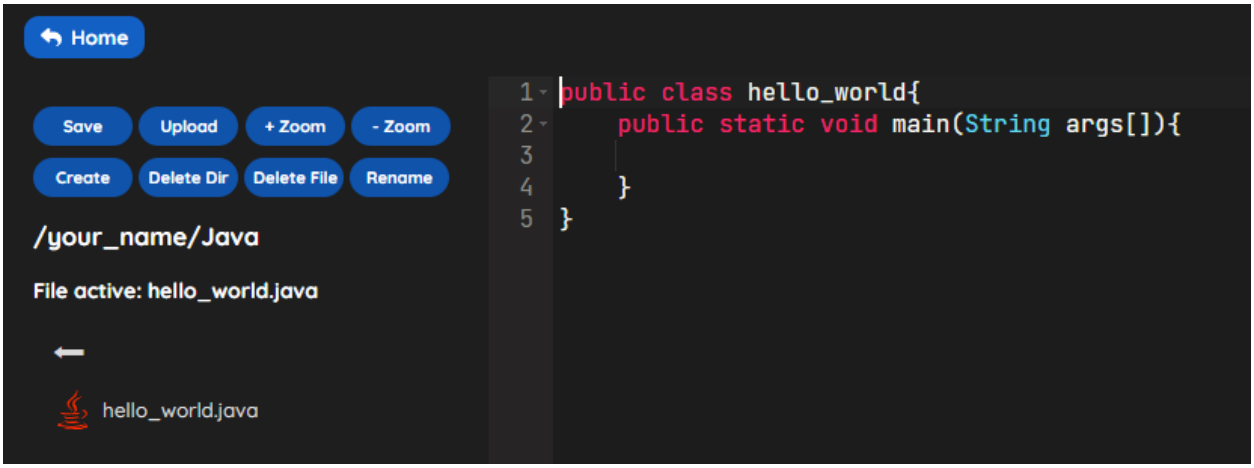
Java programs have the .java file extension, so to tell ELSIE that we want to code in java, we'll need to create files called [name].java. Unlike most other languages, names of files actually matter to the code itself. Lucky for us, ELSIE will auto build this skeleton so we don't need to worry too much about that.

Create a file called "hello_world.java". Some of the code should already be filled in, leaving you with a file looking like the image.



'public' tells ELSIE that this file should be visible to other files (this is for more advanced coding), and 'class' is telling ELSIE that 'hello_world' should be compiled to a .class file. This file is written in binary so ELSIE will hide it for us.

Next, we can build the main method that ELSIE will run. Copy the following code:



Once again, 'public' will allow the Java Runtime Environment (JRE) to access this method and execute the code found within it. The static argument is a bit more complex. 'Static' allows the Java Virtual Machine (JVM) to load the class into its memory without creating an *instance* of the class first. It's alright if what I just said passed over your head; you'll understand it with time.

Every java method will return something. Putting the argument ‘void’ allows this method to not return anything, because as soon as this method is over the code terminates. In other words, there’s no reason to return data so java won’t let you.

When Java programs run, they look for the ‘main’ method. If you wrote ‘myMain’ instead, then the class won’t compile because the JVM can’t find what it’s looking for. String args[] are for formatting the arguments you write in this method.

Ok, now that the wordy explanations are out of the way we can continue writing code. This final bit is what causes an output to the console log.

System.out.println(); is pretty self-explanitory. This line of command is telling the *system* to *output* to the console, *printing* the *line* “Hello!” When you run this command, it should give you the following output:

Great! You just wrote your first Java program! Continue to Lesson 1 to improve your Java coding skills.