Document Version: V1.1
This content is licensed under : **CC BY NC 4.0**

This document is licensed under the Creative Commons Attribution-Non Commercial 4.0 International License. You are free to share and redistribute it in any medium or format, adapt it, and build upon it, provided you give appropriate credit, do not use it for commercial purposes, and distribute your contributions under the same license.

You must not monetize this work, sell it, or use it for any commercial purpose.

Thought you looked a copyright notice! Well, there's more to it than that. If you want to know what you can and can't do with this work, see the full license in the LICENSE file in the root folder of this project.

By forwarding this, you are agreeing to every word written on this page. Even the ones you didn't read.

Latest Updates!

How long does this whole thing takes to read : 20min

Start here!!!!

Before we start... (skippable)

Why do we ask in order to do things?
Think of it like this, you and your friends are making a sandwich, you need to give explicit instructions to them (and) but you need to give explicit instructions to them in order for them to work.

This is basically your relationship between you and your computer, as a programmer: **You ask to let the computer know what you want.**

There are many ways to communicate with your computer, you can write machine code, which is close to "talk" to your computer, like English, there is "dialects" and "accents" to English, there is ways your computer would not understand what the book are you talking. This is like a book author trying to talk to you in a different language, or using too many slang and period ending with zero expressions.

So what is "grammar" in coding?
Syntax, yes, I have no idea what this word means, but it's the rules you need to follow when talking through.
Oh yes, capitalization/cases? matters!

How do my computer understand my code?
Well..... our computer understand 1s and 0s.
screw 1s, grab time.

Right Disclaimer :
Terms that are shown in here may not fully reflect the real world, words or certain terminology are modified for easy understanding

More extensive topic can be covered if there is a lot of feedback.

This does not cover usage of Intellisense

Exercise are not planned to make, but a 'maybe'

This document is bound to have grammatical mistakes, GPT is if pros ready don't understand the language

What? We are in the age of AI, slap AI to your programming journey!

Example of

```
public class Main {
    public static void main(String[] args) {
        //Insert Hello World
        System.out.println("Hello World");
    }
}
```

Welcome to the world of Java !
This is just a simple example.

For now, don't worry about the lines of code just follow the end message.

It is **compulsory-free** to wander around!

Comments :
// Type this to create a single line comment

Multi-Line comments can be created by this format:

/*
Hello
we love ^ /

Writing code we have a type of commenting called **Comments**, it's basically a sentence that is ignored than "runs" data, friends have a website that gives the exact meaning of a word, and you can write what you can, from the data literature. That website you are a **dictionary**, and the **Comments** are **webpage** that holds information or information of methods.

You can start a Java Doc comment like this :

/**

We recommend you GPT or google or search up how to write JavaDoc as writing JavaDoc is not essential for a program to run, and it is too extensive to be simply you.

```
public class Main {
    public static void main(String[] args) {
        //Insert Hello World
        System.out.println("Hello World");
    }
}
```

Really Important Stuff!

It's easier to divide the code into "blocks", so here is 4 extra seconds for you to "see".

Our computers doesn't visually see each "blocks" of code, so we always need to find a way to dissect our code, such that our computer would understand what code is part of the "block", what isn't.

Or in other words :
Imagine a "block" is a "sandwich" (hey ouch), we wouldn't say a bunch of words non-stop and expect them to understand what we said. We would take actual pause to separate our sentences.

And usually in **writing**, we would use **"."** to **separate our sentence**. In **Java**, we would use **";"**.

Method

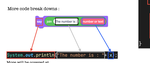
A procedure that can be executed by calling it.

```
public class Main {
    public static void main(String[] args) {
        System.out.println("This is in the main method");
        method_1();//Call
    }

    public static void method_1(int x){//Usage
        System.out.println("The number is : " + x);
    }
}
```

Important note:
If you are calling a method outside of its class, the method must set to public and you call it by its **ClassName.Method_Name()** (i.e Main.method_1())

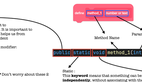
More code break downs :




```
System.out.println("The number is : " + x);
```

More will be covered at **Class & Objects** section.

Declaring a Method(making a method)




```
public static void method_1(int x){
}
```

Parameter vs Argument :
When we create methods, we can optionally **specify the parameter** of methods, but when we call the method, the **argument** we give to the method is called **argument**.

Analogy:
Back in mass drows, we used a key for door(method) is expected a key(parameter), when we insert the key(argument) and try to turn it(execute the method).

if
// ...
// ...
// ...