Zoho Schools for Graduate Studies

Notes

Java introduction

What is the primary requirement for developing a program?

- Platform

What is a platform?

- A **Platform** is a **base or environment** where something can operate, run, or be built upon.
- A Platform comprises of **Operating system** (Hardware Manager) and its **associated hardware.**

Example of Platform:

- **Windows**
- **macOS**
- **Linux**
- **Android**

What is the secondary requirement for developing a program?

- Any editor.

What is an editor?

Editor is a software tool used to **write and edit code.**

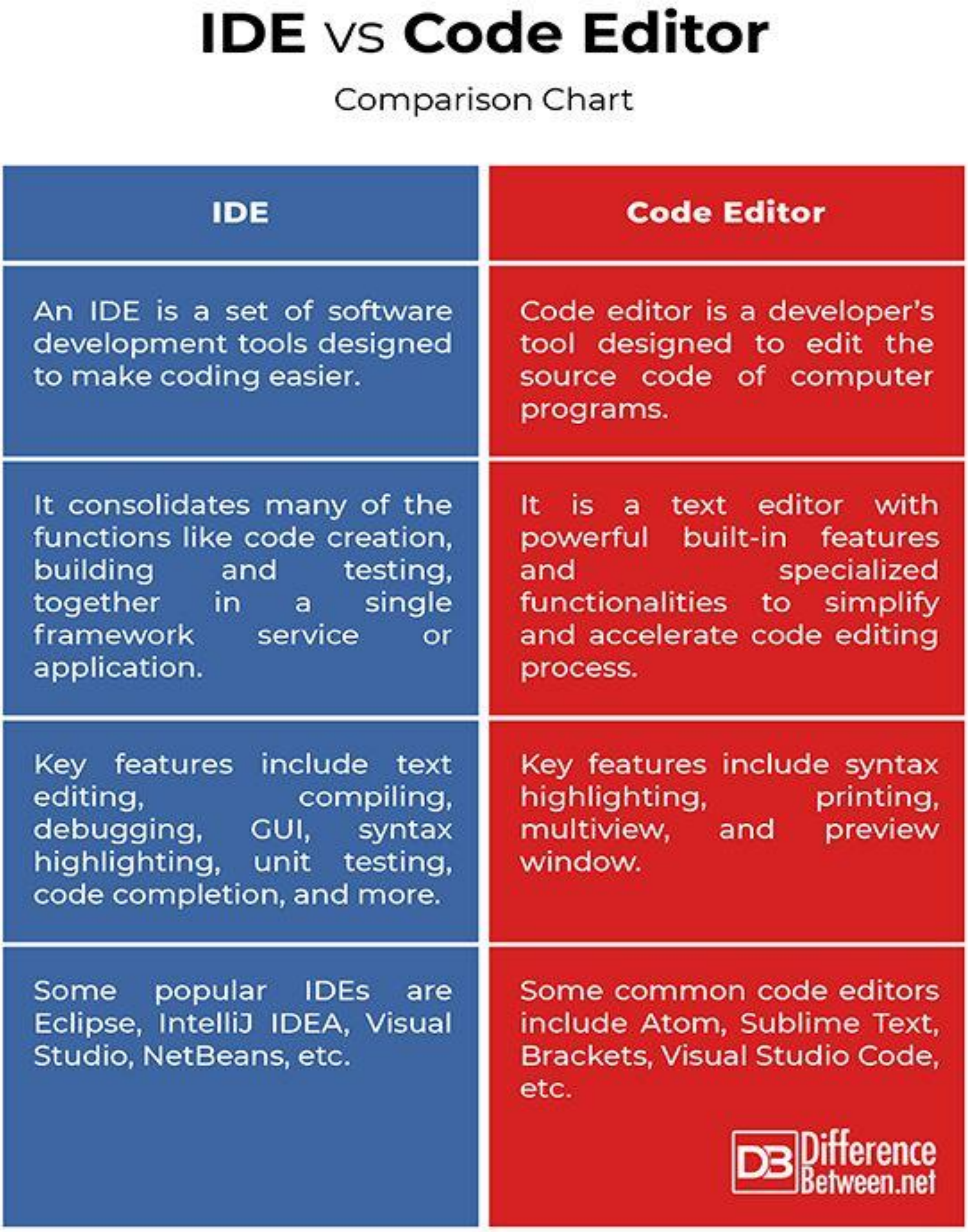
Requirements to develop and run a Java program?

- JDK (java development kit).

Requirement to run a java program?

- JRE (Java runtime environment).

Difference between Editor and IDE



IDE vs Code Editor

Comparison Chart

IDE	Code Editor
An IDE is a set of software development tools designed to make coding easier.	Code editor is a developer's tool designed to edit the source code of computer programs.
It consolidates many of the functions like code creation, building and testing, together in a single framework service or application.	It is a text editor with powerful built-in features and specialized functionalities to simplify and accelerate code editing process.
Key features include text editing, compiling, debugging, GUI, syntax highlighting, unit testing, code completion, and more.	Key features include syntax highlighting, printing, multiview, and preview window.
Some popular IDEs are Eclipse, IntelliJ IDEA, Visual Studio, NetBeans, etc.	Some common code editors include Atom, Sublime Text, Brackets, Visual Studio Code, etc.

DifferenceBetween.net

"Without the main method, we can compile the program, but we cannot run it."

Ex;

```
 public class Hello {
          public void greet() {
                  System.out.println("Hello, world!");
          }
}
```

What happens:

- It compiles successfully — no errors.
- But when you try to run it, you get an error like:

```
Error: Main method not found in class Hello, please define the main method as:
     Public static void main(String[] args)
```

Syntax for a class

class ClassName {}

Why main method required for execution?

The main() method tells the JVM where to start, Without it, the program has no entry point to run.

Where does the execution end?

Execution ends at the end of the main() method.

Syntax for a method

```
[access modifier] [non-access modifier]
returntype <methodName> ([ParameterList ]){
      //code
}
```

Let’s decode this main method syntax

```
public static void main(String[] args){
      //code
}
```

Without access modifier

```
static void main(String[] args) {
      // code
}
```

What happens?

- The program compiles fine (no compile-time error).
- But when you try to run it, the JVM cannot find the entry point.

```
Error: Main method not found in class ClassName, please define the main method as:
   public static void main(String[] args)
```

Without static keyword

```
public void main(String[] args) {
      // code
}
```

What happens?

- **The program compiles successfully** (no compile-time error).
- But when you try to **run** it, the JVM **cannot use this method as the entry point**

```
Error: Main method is not static in class ClassName, please define the main method as:
  public static void main(String[] args)
```

Without void return type

```
public static int main(String[] args) {
      return 0;
}
```

What happens?

- **The program compiles successfully** (no compile-time error in some IDEs, but depends on the compiler).
- When you try to **run** it, the JVM **does not recognize this as the valid entry point.**

```
Error: Main method not found in class ClassName, please define the main method as:
  public static void main(String[] args)
```

Without main keyword

```
public static void start(String[] args) {
       // code
}
```

What happens?

- The program compiles fine.
- But when you try to run it, the JVM cannot find the entry point method named main.

```
Error: Main method not found in class ClassName, please define the main method as:
   public static void main(String[] args)
```

Without parameter name

```
public static void main(String[]){
      // code
}
```

What happens?

- This will NOT compile because the parameter lacks a name.

```
          error: '(' expected
public static void main(String[]){
```

Without String array

```
public static void main() {
      // code
}
```

What happens?

- The program compiles successfully.
- But when you run it, the JVM cannot find the valid main method.

```
Error: Main method not found in class ClassName, please define the main method as:
   public static void main(String[] args)
```

Can change the name of the args parameter

```
public static void main(String[] args) { }

public static void main(String[] myArgs) { }

public static void main(String[] input) { }
```

Why?

- The JVM only cares about the type and position of the parameter (String[]), not the name.
- The name args is just a convention, not a requirement.

Overloaded Methods for main method

```
public class Example {
      public static void main(int[] args) {
            System.out.println("This is a normal method.");
      }

      public static void main(String[] args) {
            Example obj = new Example();
      }
}
```

What happens?

- The JVM runs the main(String[] args) method as the program's starting point.
- The main(int[] args) method is ignored unless called explicitly; no errors occur.

Note: Main method can be overloaded, but it's a bad practice.

Normal method named main

```
public class Example {
      // Normal method named 'main'
      void main() {
           System.out.println("This is a normal method.");
      }

      public static void main(String[] args) {
            Example obj = new Example();
            obj.main();  // Calling the normal method
      }
 }
```

What happens?

- method named main is treated like a regular method.

Note: Any method can be named main, but it’s a bad practice.

Rules for main method:

- Main method must be public.
- Main method must be static.
- Return type of main method must be void.
- Main is the keyword so it must be in lowercase.
- Parameter of main method must be array of string.

Miscellaneous:

- The package statement must be on the first line.
- In Java, any whole number is taken as int by default.
- In Java, any fractional number is taken as double by default.