<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<style>\* Section Style \*/

    .section {

        background-color: #ffffff;

        padding: 40px;

        margin: 40px auto;

        max-width: 1200px;

        border-radius: 10px;

        box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);

        transition: transform 0.3s ease, box-shadow 0.3s ease;

    }

    .section:hover {

        transform: translateY(-10px);

        box-shadow: 0 16px 30px rgba(0, 0, 0, 0.2);

    }

    /\* Heading Styles \*/

    h2, h3, h1 {

        color: #1e2a3a;

        font-weight: 600;

        margin-bottom: 15px;

        letter-spacing: 1px;

        text-align: center;

    }

    h2 {

        font-size: 2.5rem;

        margin-bottom: 20px;

    }

    h3 {

        font-size: 2rem;

        margin-bottom: 10px;

    }

    h1 {

        font-size: 2.8rem;

        margin-top: 30px;

        text-align: left;

    }

    /\* Paragraphs and Strong Text \*/

    p {

        font-size: 1.1rem;

        line-height: 1.7;

        color: #333;

        margin-bottom: 20px;

    }

    strong {

        color: #00bcd4;

        font-weight: bold;

    }

    /\* Code and Preformatted Text \*/

    pre {

        background-color: #2e3b47;

        color: #e0e0e0;

        padding: 20px;

        border-radius: 8px;

        margin-bottom: 20px;

        overflow-x: auto;

        font-size: 1rem;

        line-height: 1.6;

    }

    code {

        font-family: 'Courier New', Courier, monospace;

        color: #e0e0e0;

        background-color: #2e3b47;

        padding: 5px 10px;

        border-radius: 4px;

    }

    pre code {

        display: block;

        white-space: pre-wrap;

        word-wrap: break-word;

    }

    /\* List Styles \*/

    ul {

        list-style-type: disc;

        margin-left: 40px;

        font-size: 1.1rem;

    }

    ul li {

        margin: 10px 0;

        line-height: 1.6;

    }

    /\* Callout Section \*/

    h2 + p {

        font-size: 1.2rem;

        color: #00bcd4;

        font-weight: bold;

    }

    /\* Button or Actions Section \*/

    h3 + p {

        font-size: 1.1rem;

        color: #333;

        margin-top: 15px;

    }

    /\* Code Block Section \*/

    h3 + p + pre {

        margin-top: 20px;

        padding: 20px;

        background-color: #f4f4f4;

        border-left: 4px solid #00bcd4;

    }

    /\* Responsive Design \*/

    @media (max-width: 768px) {

        .section {

            padding: 20px;

        }

        h2, h3, h1 {

            font-size: 1.8rem;

        }

        ul {

            margin-left: 20px;

        }

    }</style>

<body>

    <section id="Java" class="section">

        <h2>Java Introduction</h2>

        <p> <strong>Java  </strong>is a high-level, object-oriented programming language that was developed by James Gosling and Mike Sheridan at Sun Microsystems (which was later acquired by Oracle Corporation). It was first released in 1995 and has since become one of the most widely used programming languages in the world. Java is designed to be platform-independent, meaning that programs written in Java can run on any device or operating system that has a Java Virtual Machine (JVM) installed.</br>

            </p>

        <h3>Basic Syntax of Java </h3>

        <p>A simple Java program consists of the following main components:</br>

            Class Declaration: Every Java application must have at least one class.</br>

            Main Method: The entry point where the program starts executing.</br>

            Statements: Instructions that the program follows to perform actions.

        </p>

        <pre>

    <code>

        public class HelloWorld {

            public static void main(String[] args) {

                System.out.println("Hello, World!");

            }

        }

        </code>

        </pre>

        <p><strong>Why use Java?</strong></p>

        <p> Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)</br></li>

        <li>It is one of the most popular programming languages in the world</br></li>

        <li> It has a large demand in the current job market</br></li>

         <li>It is easy to learn and simple to use</br></li>

         <li>It is open-source and free</br></li>

         <li> It is secure, fast and powerful</br></li>

         <li>It has huge community support (tens of millions of developers)</br></li>

        <li>Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs</br></li>

         <li>As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa</br></li>

         </p>

             </ul>

        <h1><strong> Java Elements</strong></h1>

        <p>Summary of Key Java Elements:</p>

        <ul>

        <li><strong>abstract:-</strong>A non-access modifier. Used for classes and methods: An abstract class cannot be used to create objects (to access it, it must be inherited from another class). An abstract method can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from)</br>

        <li><strong>assert:-</strong>assert For debugging</br>

         <li><strong>boolean:-</strong>boolean  A data type that can only store true or false values</br>

         <li><strong>break:-</strong>Breaks out of a loop or a switch block</br>

         <li><strong>byte:-</strong>A data type that can store whole numbers from -128 and 127</br>

        <li><strong>case:-</strong>case Marks a block of code in switch statements</br>

        <li><strong>catch:-</strong>Catches exceptions generated by try statements</br>

        <li><strong>char:-</strong> A data type that is used to store a single character</br>

        <li><strong>class:-</strong>Defines a class</br>

         <li><strong>continue:-</strong>Continues to the next iteration of a loop</br>

        <li><strong>const:-</strong>Defines a constant. Not in use - use final instead</br>

        <li><strong>default:-</strong>Specifies the default block of code in a switch statement</br>

        <li><strong>do:-</strong>Used together with while to create a do-while loop</br>

        <li><strong>double:-</strong>A data type that can store fractional numbers from 1.7e−308 to 1.7e+308</br>

         <li><strong>else:-</strong>Used in conditional statements</br>

         <li><strong>enum:-</strong>Declares an enumerated (unchangeable) type</br>

        <li><strong>exports:-</strong>Exports a package with a module. New in Java 9</br>

         <li><strong>extends:-</strong>Extends a class (indicates that a class is inherited from another class)</br>

         <li><strong>final:-</strong>A non-access modifier used for classes, attributes and methods, which makes them non-changeable (impossible to inherit or override)</br>

        <li><strong>finally:-</strong>Used with exceptions, a block of code that will be executed no matter if there is an exception or not</br>

        <li><strong>float:-</strong>A data type that can store fractional numbers from 3.4e−038 to 3.4e+038</br>

         <li><strong>for:-</strong>Create a for loop</br>

        <li><strong>goto:-</strong>Not in use, and has no function</br>

        <li><strong>if:-</strong>Makes a conditional statement</br>

        <li><strong>implements:-</strong>Implements an interface</br>

        <li><strong>import:-</strong>Used to import a package, class or interface</br>

        <li><strong>instanceof:-</strong>Checks whether an object is an instance of a specific class or an interface</br>

    <li><strong>int:-</strong>A data type that can store whole numbers from -2147483648 to 2147483647</br>

    <li><strong>interface:-</strong>Used to declare a special type of class that only contains abstract methods</br>

    <li><strong>long:-</strong> A data type that can store whole numbers from -9223372036854775808 to 9223372036854775808</br>

    <li><strong>module:-</strong>Declares a module. New in Java 9</br>

    <li><strong>native:-</strong>Specifies that a method is not implemented in the same Java source file (but in another language)</br>

    <li><strong>new:-</strong>Creates new objects</br>

    <li><strong>package:-</strong>Declares a package</br>

    <li><strong>private:-</strong>An access modifier used for attributes, methods and constructors, making them only accessible within the declared class</br>

    <li><strong>protected:-</strong>An access modifier used for attributes, methods and constructors, making them accessible in the same package and subclasses</br>

    <li><strong>public:-</strong>An access modifier used for classes, attributes, methods and constructors, making them accessible by any other class</br>

    <li><strong>requires:-</strong>Specifies required libraries inside a module. New in Java 9</br>

    <li><strong>return:-</strong>Finished the execution of a method, and can be used to return a value from a method</br>

    <li><strong>short:-</strong>A data type that can store whole numbers from -32768 to 32767</br>

    <li><strong>static:-</strong>A non-access modifier used for methods and attributes. Static methods/attributes can be accessed without creating an object of a class</br>

    <li><strong>strictfp:-</strong>Obsolete. Restrict the precision and rounding of floating point calculations</br>

    <li><strong>super:-</strong>Refers to superclass (parent) objects</br>

    <li><strong>switch:-</strong>Selects one of many code blocks to be executed</br>

    <li><strong>synchronized:-</strong>A non-access modifier, which specifies that methods can only be accessed by one thread at a time</br>

    <li><strong>this:-</strong> Refers to the current object in a method or constructor</br>

    <li><strong>throw:-</strong>Creates a custom error</br>

    <li><strong>throws:-</strong>Indicates what exceptions may be thrown by a method</br>

    <li><strong>transient:-</strong>Used to ignore an attribute when serializing an object</br>

    <li><strong>try:-</strong>Creates a try...catch statement</br>

    <li><strong>var:-</strong>Declares a variable. New in Java 10</br>

    <li><strong>void:-</strong>Specifies that a method should not have a return value</br>

    <li><strong>volatile:-</strong>Indicates that an attribute is not cached thread-locally, and is always read from the "main memory"</br>

    while   Creates a while loop

        </ul>

        <h2>Java Example: A Simple hello world program</h2>

        <pre>

    <code> public class HelloWorld {  // Define the class named HelloWorld

        public static void main(String[] args) {  // The main method - entry point of the program

            System.out.println("Hello, World!");  // Print "Hello, World!" to the console

        }

    }

    </code>

        </pre>

        <h3>Explanation of the Program:</h3>

        <p>1. public class HelloWorld:

         <li>This defines a class named HelloWorld. In Java, every application must have at least one class, and the class name must match the file name (i.e., HelloWorld.java).</li></p>

            <p>2.public static void main(String[] args):

            <li>public: This means that the main method is accessible from anywhere.</li>

            <li>static: This means that the method can be called without creating an instance of the HelloWorld class.</li>

            <li>void: This indicates that the main method doesn't return any value.</li>

            <li>String[] args: This is an array that can hold any command-line arguments passed to the program when it is run. For this simple example, it's not used, but it’s always part of the method signature.</li></p>

            <p> 3.System.out.println("Hello, World!");:

            <li>System.out: Refers to the standard output stream (usually the console).</li>

            <li>println(): This is a method that prints the argument (in this case, the string "Hello, World!") to the console and then moves to the next line.</li>

        </p>

        <h3>How to Compile and Run:</h3>

        <p>1.Save the program to a file named HelloWorld.java.</br>

           2. Compile the program using the Java compiler (javac):</br>

            javac HelloWorld.java</br>

           3. Run the compiled program using the Java Virtual Machine (java):</br>

            java HelloWorld

        <h2>Output:</h2>

        When you run the program, the output will be:</br>

        Hello, World!</br>

        This is a basic example of a Java program that demonstrates the syntax and structure, and how to print output to the console.</br>

    </section>

</body>

</html>