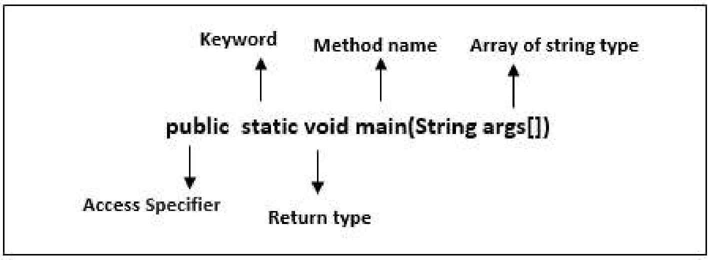
⇧ SCROLL TO TOP

Java main() method

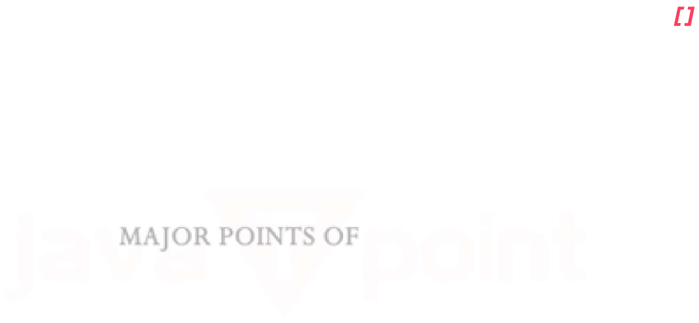
The main() is the starting point for JVM to start execution of a Java program. Without the main() method, JVM will not execute the program. The syntax of the main() method is:



public: It is an access specifier. We should use a public keyword before the main() method so that JVM can identify the execution point of the program. If we use private, protected, and default before the main() method, it will not be visible to JVM.

static: You can make a method static by using the keyword static. We should call the main() method without creating an object. Static methods are the method which invokes without creating the objects, so we do not need any object to call the main() method.

void: In Java, every method has the return type. Void keyword acknowledges the compiler that main() method does not return any value.



main(): It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the main() method.

String args[]: The main() method also accepts some data from the user. It accepts a group of strings, which is called a string array. It is used to hold the command line arguments in the form of string values.

```
main(String args[])
```

Here, agrs[] is the array name, and it is of String type. It means that it can store a group of string. Remember, this array can also store a group of numbers but in the form of string only. Values passed to the main() method is called arguments. These arguments are stored into args[] array, so the name args[] is generally used for it.

What happens if the main() method is written without String args[]?

mpile, but not run, because JVM will not recognize the main() method. Remember JVM always looks

⇧ SCROLL TO TOP

with a string type array as a parameter.

Execution Process

First, JVM executes the static block, then it executes static methods, and then it creates the object needed by the program. Finally, it executes the instance methods. JVM executes a static block on the highest priority basis. It means JVM first goes to static block even before it looks for the main() method in the program.

Example

```
class  Demo
{
static                //static block
{
System.out.println("Static block");
}
public static void main(String args[])  //static method
{
System.out.println("Static method");
}
}
```

Output:

```
Static block
Static method
```

We observe that JVM first executes the static block, if it is present in the program. After that it searches for the main() method. If the main() method is not found, it gives error.

Example

A program that does not have the main() method gives an error at run time.

```
class DemoStaticBlock
{
Static                            //static block
{
System.out.println("Static block");
```

⇧ SCROLL TO TOP

```
}
```

Output:

```
Error: Main method not found in the class Demo, please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

So the main() method should always be written as:

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

We can interchange public and static and write it as follows:

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

We can also use the different name for the String type array and write it as:

```
static public void main(String[] x)
```

Different ways of writing main() method are:

```
static public void main(String []x)
static public void main(String...args)
```

String...args: It allows the method to accept zero or multiple arguments. There should be exactly three dots between String and array; otherwise, it gives an error.

Example

A program that has no main() method, but compile and runs successfully.

```
abstract class DemoNoMain extends javafx.application.Application
{
static      //static block
{
System.out.println("Java");
```

⇧ SCROLL TO TOP

```
}
}
```

Output:

```
Java
```

Overloading of main() method

We can also overload the main() method. We can define any number of main() method in the class, but the method signature must be different.

Example

```
class OverloadMain
{
public static void main(int a)  //overloaded main method
{
System.out.println(a);
}
public static void main(String args[])
{
System.out.println("main method incoked");
main(6);
}
}
```

Output:

```
main method invoked
6
```

← Prev

Next →

⇧ SCROLL TO TOP

For Videos Join Our Youtube Channel: Join Now

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share

Program by IIT Roorkee EICT

Learn from IIT Professors & Industry Practitioners. Adva
Career Now!

Intellipaat

Learn Latest Tutorials

Splunk	SPSS	Swagger	Transact-SQL	Tumblr
React tutorial ReactJS	Regex tutorial Regex	Reinforcement learning tutorial Reinforcement Learning	R Programming tutorial R Programming	RxJS tutorial RxJS
React Native	Python Design Patterns	Python Pillow tutorial Python Pillow	Python Turtle tutorial Python Turtle	Keras tutorial Keras

⇧ SCROLL TO TOP

Python Design Patterns

Preparation

Aptitude
Aptitude

Logical Reasoning
Reasoning

Verbal Ability
Verbal Ability

Interview Questions
Interview Questions

Company Interview Questions
Company Questions

Trending Technologies

Artificial Intelligence Tutorial
Artificial Intelligence

AWS Tutorial
AWS

Selenium tutorial
Selenium

Cloud Computing tutorial
Cloud Computing

Hadoop tutorial
Hadoop

ReactJS Tutorial
ReactJS

Data Science Tutorial
Data Science

Angular 7 Tutorial
Angular 7

Blockchain Tutorial
Blockchain

Git Tutorial
Git

Machine Learning Tutorial
Machine Learning

DevOps Tutorial
DevOps

B.Tech / MCA

DBMS tutorial
DBMS

Data Structures tutorial
Data Structures

DAA tutorial
DAA

Operating System tutorial
Operating System

Computer Network tutorial
Computer Network

⇧ SCROLL TO TOP

Compiler Design tutorial

Compiler Design

Computer Organization and Architecture

Computer Organization

Discrete Mathematics Tutorial

Discrete Mathematics

Ethical Hacking Tutorial

Ethical Hacking

Computer Graphics Tutorial

Computer Graphics

Software Engineering Tutorial

Software Engineering

html tutorial

Web Technology

Cyber Security tutorial

Cyber Security

Automata Tutorial

Automata

C Language tutorial

C Programming

C++ tutorial

C++

Java tutorial

Java

.Net Framework tutorial

.Net

Python tutorial

Python

List of Programs

Programs

Control Systems tutorial

Control System

Data Mining Tutorial

Data Mining

Data Warehouse Tutorial

Data Warehouse

⇧ SCROLL TO TOP