

Java Methods

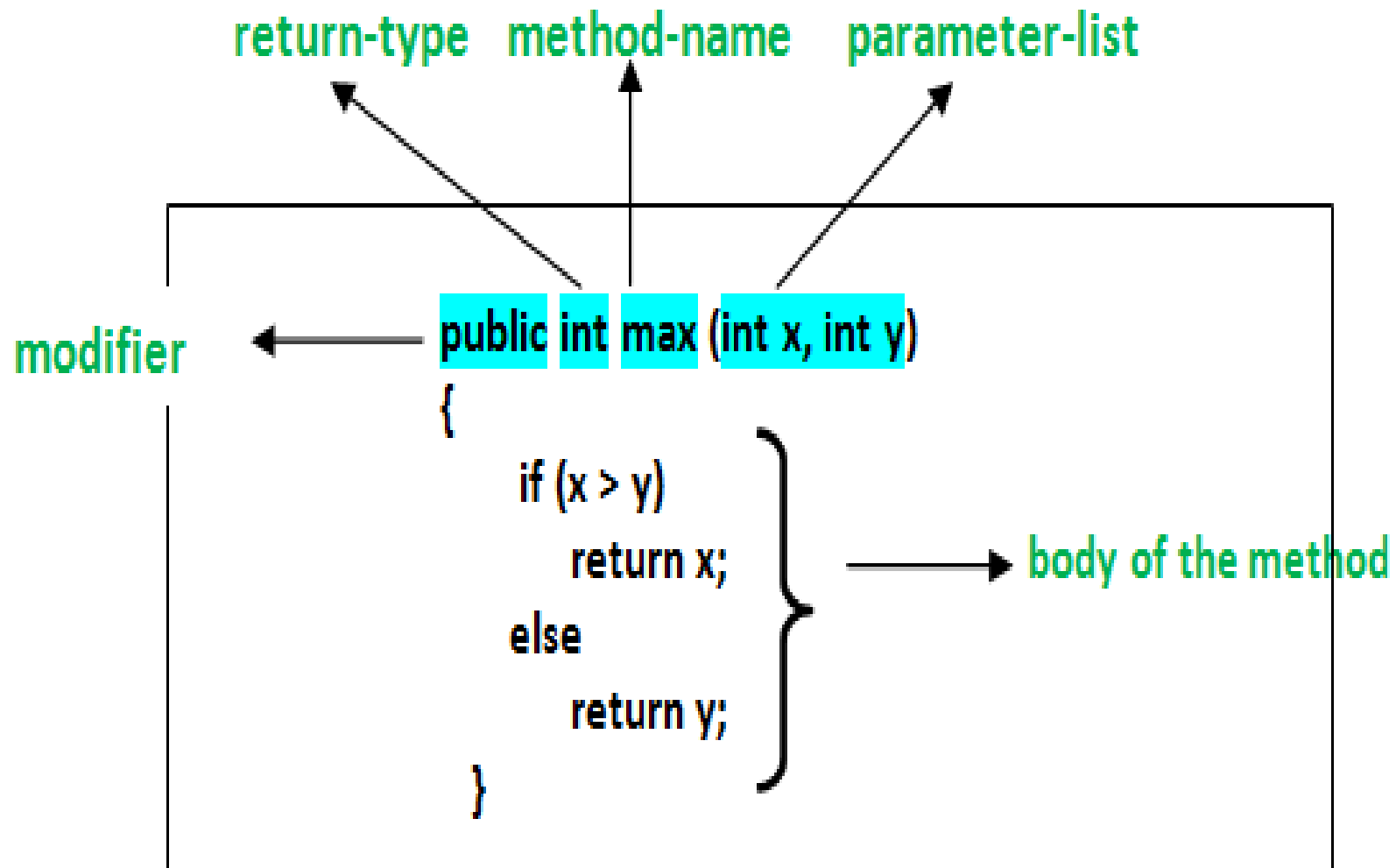
Java Methods

- A **method** is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a method.
- Methods are used to perform certain actions, and they are also known as **functions**.
- Why use methods? To reuse code: define the code once, and use it many times.

Methods Declaration

In general, method declarations has six components :

1. Modifier
2. Return Type
3. Method Name
4. Parameter List
5. Exception List
6. Method Body



Types of Methods in Java

- **Predefined Method**
- Math functions – `sqrt()`, `log()`, `min()`, `max()`, `avg()`, `sin()`, `cos()`, `tan()`, `round()`, `floor()`, `abs()` etc.
- String function – `length()`, `substring()`, `replace()`, `charAt()`, `indexOf()`, `trim()` etc.
- **User-defined Method**

Method Signature

- It consists of the method name and a parameter list (number of parameters, type of the parameters, and order of the parameters). The return type and exceptions are not considered as part of it.
- Method Signature of the above function:
- `max(int x, int y)` Number of parameters is 2, Type of parameter is int.

How to Name a Method?

Rules to Name a Method

- While defining a method, remember that the method name must be a **verb** and start with a **lowercase** letter.
- If the method name has more than two words, the first name must be a verb followed by an adjective or noun.
- In the multi-word method name, the first letter of each word must be in **uppercase** except the first word. For example, findSum, computeMax, setX and getX.

Call a method in java

```
public class Main {  
    static void myMethod() {  
        System.out.println("I just got executed!");  
    }  
  
    public static void main(String[] args) {  
        myMethod();  
    }  
}
```

I just got executed!

A method can also be called multiple times:

```
public class Main {  
    static void myMethod() {  
        System.out.println("I just got executed!");  
    }  
  
    public static void main(String[] args) {  
        myMethod();  
        myMethod();  
        myMethod();  
    }  
}
```

```
I just got executed!  
I just got executed!  
I just got executed!
```

Parameters and Arguments

```
public class Main {  
    static void myMethod(String fname) {  
        System.out.println(fname + " Refsnes");  
    }  
  
    public static void main(String[] args) {  
        myMethod("Liam");  
        myMethod("Jenny");  
        myMethod("Anja");  
    }  
}
```

```
Liam Refsnes  
Jenny Refsnes  
Anja Refsnes
```

Multiple Parameters

```
public class Main {  
    static void myMethod(String fname, int age) {  
        System.out.println(fname + " is " + age);  
    }  
  
    public static void main(String[] args) {  
        myMethod("Liam", 5);  
        myMethod("Jenny", 8);  
        myMethod("Anja", 31);  
    }  
}
```

Liam is 5

Jenny is 8

Anja is 31

Return Values

```
public class Main {  
    static int myMethod(int x) {  
        return 5 + x;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(myMethod(3));  
    }  
}
```

8

Return with 2 parameters

```
public class Main {  
    static int myMethod(int x, int y) {  
        return x + y;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(myMethod(5, 3));  
    }  
}
```

8

```
public class Main {  
    static int myMethod(int x, int y) {  
        return x + y;  
    }  
  
    public static void main(String[] args) {  
        int z = myMethod(5, 3);  
        System.out.println(z);  
    }  
}
```

8

A Method with If...Else

```
public class Main {  
  
    // Create a checkAge() method with an integer parameter called age  
    static void checkAge(int age) {  
  
        // If age is less than 18, print "access denied"  
        if (age < 18) {  
            System.out.println("Access denied - You are not old enough!");  
  
            // If age is greater than, or equal to, 18, print "access granted"  
        } else {  
            System.out.println("Access granted - You are old enough!");  
        }  
  
    }  
  
    public static void main(String[] args) {  
        checkAge(20); // Call the checkAge method and pass along an age of 20  
    }  
}
```

Access granted - You are old enough!

Method Overloading

```
public class Main {  
    static int plusMethod(int x, int y) {  
        return x + y;  
    }  
  
    static double plusMethod(double x, double y) {  
        return x + y;  
    }  
  
    public static void main(String[] args) {  
        int myNum1 = plusMethod(8, 5);  
        double myNum2 = plusMethod(4.3, 6.26);  
        System.out.println("int: " + myNum1);  
        System.out.println("double: " + myNum2);  
    }  
}
```

int: 13

double: 10.559999999999999