SYNTAX TECHNOLOGIES

JAVA

Class 14

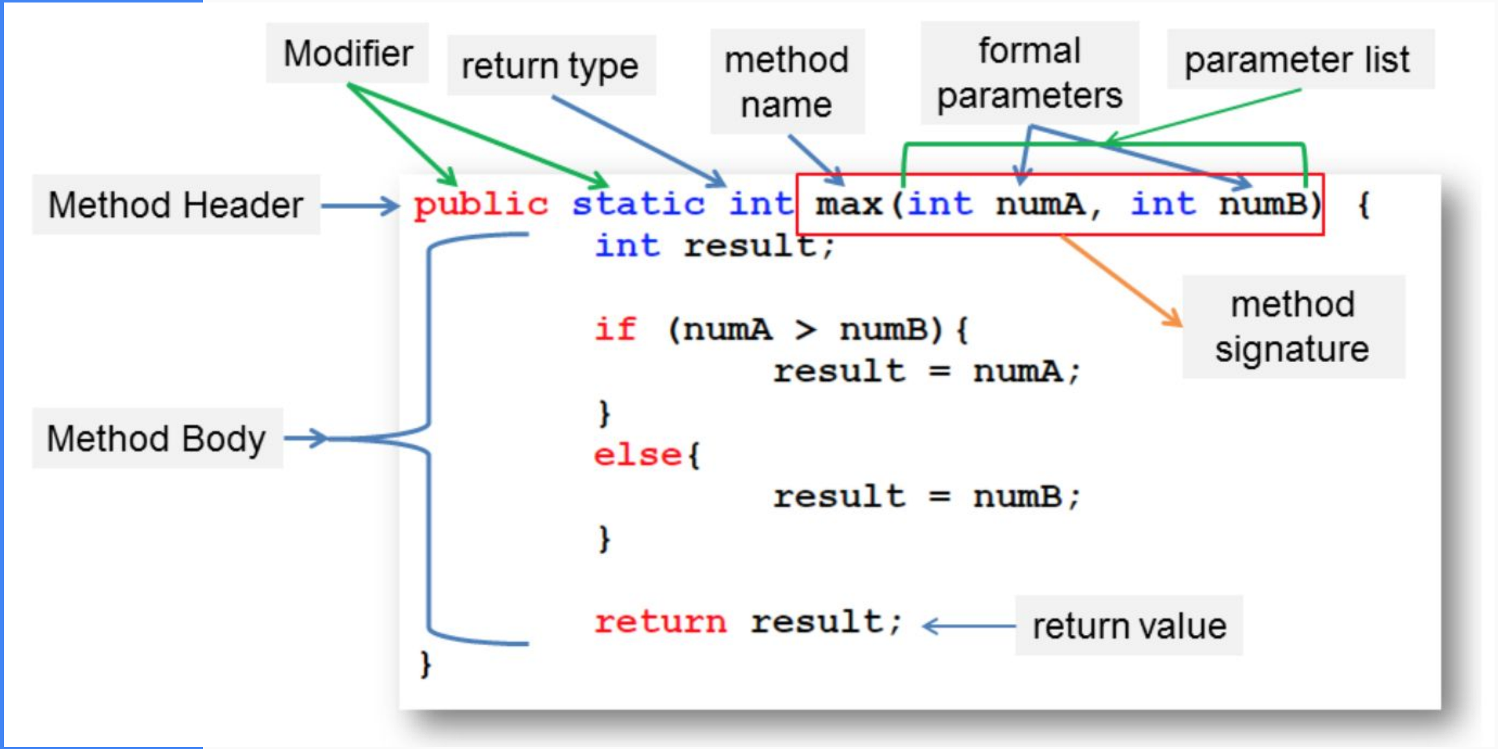
Agenda

Methods with parameters and without
Methods with return values and without

Methods in Java

Method definitions have 2 basic parts:

- The method header
- The body of the method



```
public static int max(int numA, int numB) {
        int result;

        if (numA > numB){
                result = numA;
        }
        else{
                result = numB;
        }

        return result;
}
```

Methods in Java

Modifiers: The modifier, which is optional, tells the compiler how to call the method.

Return Type: A method may return a value.
The return Value Type is the datatype of the value the method returns.

Method Name: This is the actual name of the method.

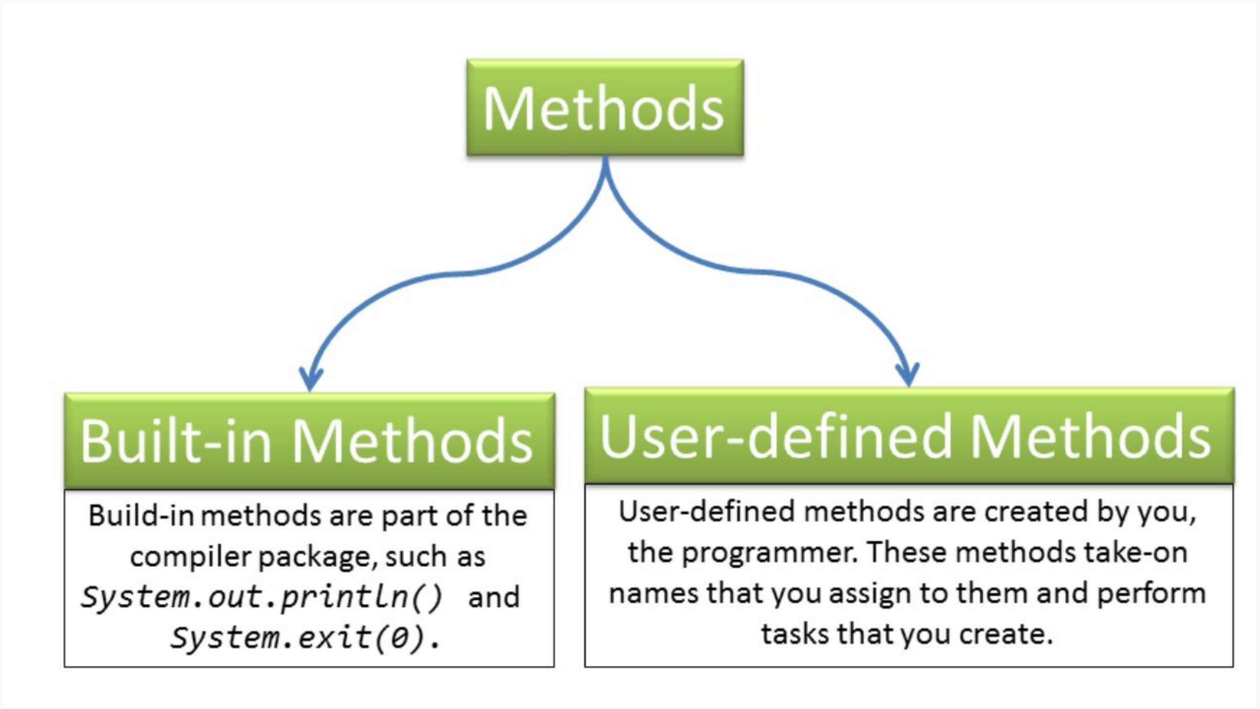
Parameters: This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a method. Parameters are optional; that is, a method may contain no parameters.

Method Body: The method body contains a collection of statements that define what the method does.

Note: In certain other languages, methods are referred to as procedures and functions. A method with a non void return value type is called a function; a method with a void return value type is called a procedure

Type of Methods in Java

There are 2 types of Methods in JAVA



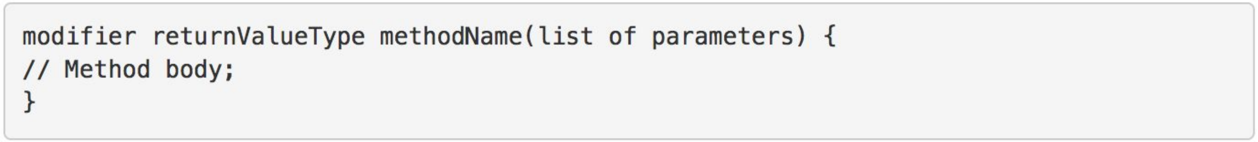
Build-in methods are part of the compiler package, such as *System.out.println()* and *System.exit(0)*.

User-defined methods are created by you, the programmer. These methods take-on names that you assign to them and perform tasks that you create.

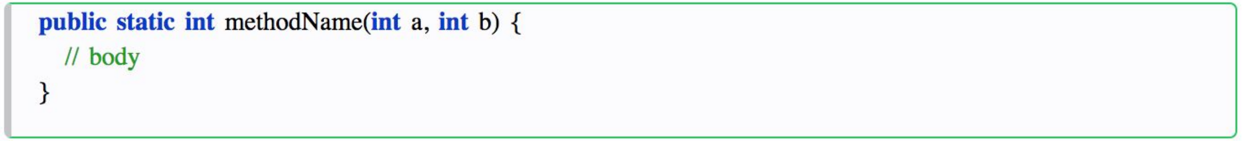
Methods without parameters

```
public class Greetings {

    public static void main(String[] args) {

        Greetings obj = new Greetings();
        obj.hello();

    }

    void hello() {
        System.out.println("Hello");
    }
}
```

Methods with parameters



```
modifier returnValueType methodName(list of parameters) {
// Method body;
}
```



```
public static int methodName(int a, int b) {
   // body
}
```

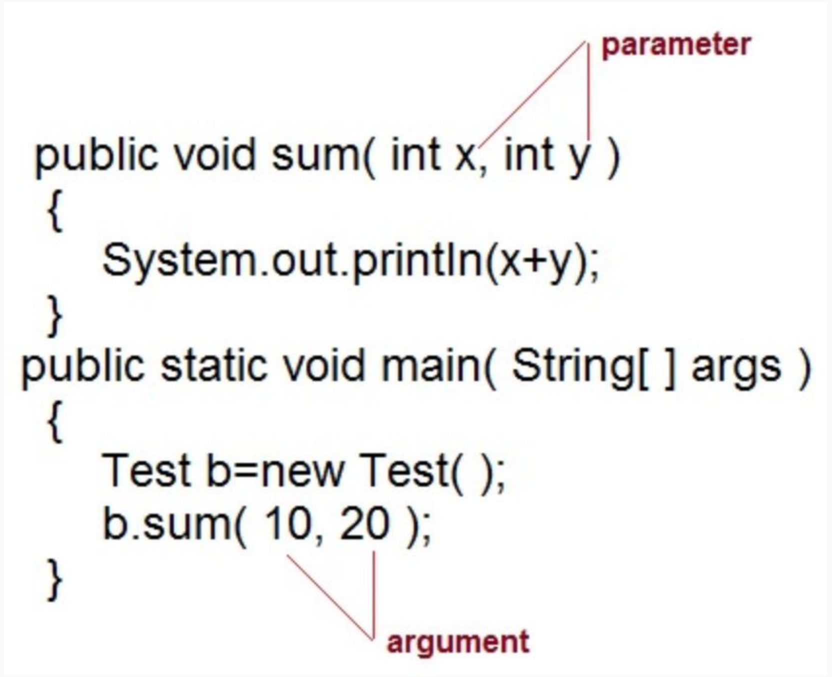
```
void sum(int a, int b) {
        System.out.println(a+b);
}

void sub(int a, int b) {

        System.out.println(a-b);
}
```

Task

1. Create a method that will take 2 parameters as a numbers and prints which number is larger.
2. Create a method that will take a number and prints whether the number is even or odd.
3. Create a method that will print whether given String is palindrome or not.
4. Create a method that will say Hello in different language based on the country that will passed when method is executed.

Parameter vs. Argument

- Parameter is variable defined by a method that receives value when the method is called.
- Parameter are always local to the method they don't have scope outside the method.
- Argument is a value that is passed to a method when it is called.



```
public void sum( int x, int y )
{
    System.out.println(x+y);
}
public static void main( String[ ] args )
{
    Test b=new Test( );
    b.sum( 10, 20 );
}
```

Methods in Java

1. Method without return any value

   Uses **void** keyword

2. Method with return values.

   Uses **return** keyword

Void keyword allows to create methods which do not return a value.

1.Method without return type and without arguments.

```
class sample{

public void add(){

int a=40;
int b=50;
int c=a+b;
SOP(c);
}

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

2.Method without return type and with arguments.

```
class sample{

public void add(int a, int b){

int c=a+b;
SOP(c);
}

public static void main(String args[]) {
 sample obj= new sample();
obj.add(13,24);
 }
}
```

3.Method with return type and without arguments.

```
class sample{

public int add(){
int a=40;
int b=50;
int c=a+b;
return c;
}

public static void main(String args[]) {

sample obj= new sample();
int x=obj.add();
SOP(x);
 }
}
```

4.Method with return type and with arguments.

```
class sample{

public int add(int a, int b){

int c=a+b;
return c;
}

public static void main(String args[]) {

sample obj= new sample();
int x=obj.add(1,2);
SOP(x);
 }
}
```

Task

1. Create a method createEmail(). Based on values of users name, lastName and email type, your method should return complete email Address. Example: createEmail(John, Snow, gmail) → johnsnow@gmail.com or

2. Write a method to return whether given number is prime or not?

3. Create class Student that will have a method getGrade. Your method should accept the score of a student and return a grade:

   score > 90 - A
   score >80 - B
   score >70 - C
   score > 50 - D
   anything else - F

Rules

- The return type is the **must** in a method, you can't declare the method without it's return type.
- If the return type is **void** it means method will **not return** any value.
- The **return** statement should be the **last** statement in the method.
- You **can't** declare more than one **return** statement in one method.
- The data type of the returned value must match the method's declared return type.
- You can **call** the method with its name only.
- The method **must** have the **body**.
- The method can accept the n number of **parameters**.
- If the method has n number of **parameter** then it's a **must** to pass all parameter in method body while **calling** the method in program code.
- The variables declared **inside** the method body are called the **local** variables.
- Methods access modifiers define the access level of method.