Welcome to the

Java Course

Module 2

Let’s resume!

```
System.out.println("text");
```

Let’s resume!

```
data_type name = value ;
```

Let’s resume!

```
import java.util.Scanner;
Scanner scanner = new Scanner(System.in);
```

Let’s resume!

```
String name = scanner.nextLine();
```

Let’s resume!

```
if (condition) {
    //condition is true;
} else {
    //condition is false;
}
```

Let’s resume!

```
while (condition) {
      // Code to be executed
}
```

Let’s resume!

```
do {
        // Code to be executed
}while (condition) ;
```

Let’s resume!

```
for (initialization;condition;update) {
      // Code to be executed
}
```

Let’s resume!

```
switch(x) {
        case 1: // Code to be executed in case 1
        case 2: // Code to be executed in case 2
        case 3: // Code to be executed in case 3
        [...]
        default : // Code to be executed by default
}
```

Let’s practice!

Functions

A function is a portion of code that can be called and executed unlimited times just using its name.

Function

```
public static void function_name(){
        //portion of code
   }
```

Function

```
public static void printHello(){
        System.out.println(" Hello World! ");
    }
```

Function

```
public static void printHello(){
        System.out.println(" Hello World! ");
    }
```

Function

```
public static void printHello(){
        System.out.println(" Hello World! ");
   }
```

Function

```
public static void printHello(){
     System.out.println(" Hello World! ");
  }
```

Function

```
public static void printHello(){
        System.out.println(" Hello World! ");
   }
```

Function

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

Function

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

Let’s practice!

Return

A function can also return a value after execution.

In this case, you have to declare the type of the returned value.

If a function returns nothing, the type is **void**.

Return

```
public type_returned function_name(){
        return value ;
    }
```

Return

```
public type_returned function_name(){
        return value ;
    }
```

Return

```
public type_returned function_name(){
        return value ;
    }
```

Return

```
public static String getHello(){
     return " Hello World! " ;
  }
```

Return

```
public static String getHello(){
        return " Hello World! " ;
    }
```

Return

```
public static int getNumber(){
        return 10 ;
    }
```

Return

```
public static int getNumber(){
        return 10 ;
    }
```

Let’s practice!

Parameter

Sometimes a function needs external informations.

These informations are called **parameters**.

Parameter

```
public void function_name(type p1, type p2, [...] ){
        //portion of code that uses parameters
    }
```

Parameter

```
public void function_name(type p1, type p2, [...] ){
        //portion of code that uses parameters
  }
```

Parameter

```
public void function_name(type p1, type p2, [...] ){
        //portion of code that uses parameters
    }
```

Return

```
public static int sum(int n1, int n2){
              return n1 + n2 ;
          }
```

Parameter

Be careful because a parameter is just a **temporary** variable that exists **only inside** the function!

Return

```
// outside the main
public void change(int number){
    number = 10 ;
}
            …
// inside the main
int number = 7 ;
change(number);   // the value of number in the main is still 7!
```

Return

```
// outside the main
public void change(int number){
      number = 10 ;
}
              …
// inside the main
int number = 7 ;
change(number);   // the value of number in the main is still 7!
```

Return

```
// outside the main
public void change(int number){
      number = 10 ;
}
            …
// inside the main
int number = 7 ;
change(number);   // the value of number in the main is still 7!
```

Return

```
// outside the main
public void change(int number){
      number = 10 ;
}
            …
// inside the main
int number = 7 ;
change(number);   // the value of number in the main is still 7!
```

Let’s practice!