if-statements

Learn the syntax for if-statements in Java.

WE'LL COVER THE FOLLOWING

- Exercise: thermometer
- Exercise: I see a wall!
- Exercise: Coin flip
- `else` and `else if` statements

The keyword `if` can be used to run a block of code only if some condition has the boolean value `true`. The syntax is like that of C or Javascript: the conditional expression must be wrapped in parentheses, and the statements to be executed should be in curly braces. If you are familiar with those languages, you may wish to skip this section.

Here are a few examples:

```
class IfExamples {
  public static void main(String args[]) {
    if(true) {
      System.out.println("This code gets executed.");
    }
    if(false) {
      System.out.println("This code does not.");
    }

    if(5 > 3) {
      System.out.println("This code gets executed, too.");
    }

    if(3 > 5) {
      System.out.println("This code does not.");
    }
  }
}
```

Exercise: thermometer

Write an if-statement in the following code that checks if the variable `temperature` has a negative value, and if so, prints out, “It’s very, very cold!”

Then change the value of temperature to ensure that nothing is printed out.

Finally, add a second if-statement to check if temperature is non-negative, and if so, prints out “It’s not so cold.”

Thermometer.java | Sample solution

```
class Temperature {

  public static void main(String[] args) {

    int temperature = -5;
  }
}
```

Exercise: I see a wall!

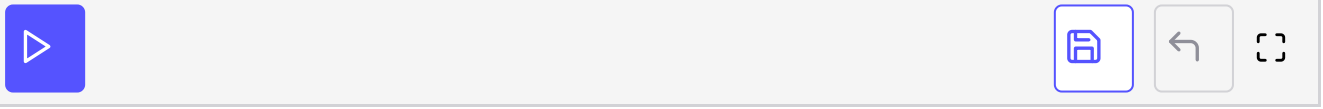
A method call causes some code to be executed, and that code may return a result for use. For example, the method call `Math.sqrt(9.0)` computes and makes available the value `3.0` for printing, storing, or some other use.

The type of value that the method returns depends on the particular method. `Math.sqrt(9.0)` returns a number. In the code below, you can use the method `r.blocked()`, which returns a boolean value of `true` or `false`, depending on whether or not there is a wall in front of the robot `r`.

In the following code, write an if-statement that prints out “I see a wall!” after the robot drives a few steps, if there is a wall in front of the robot. Try changing the movement commands to verify that the robot doesn’t report a wall if there is not one.

SeeWall.java | Sample solution

```
class RobotDemo {

        public static void main(String[] args) {

    SimpleRobot r = new SimpleRobot();

    r.forward();
    r.right();
    r.forward();
    r.left();

    // Add your if-statement here:

  }
}
```

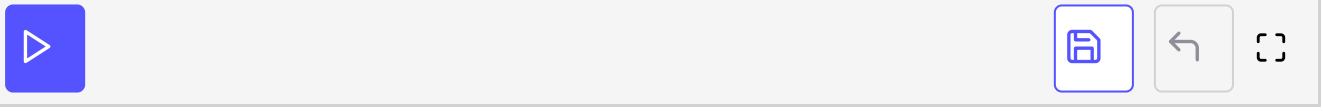


Exercise: Coin flip

The code below picks a random number that is either 0 or 1, and stores the result in the variable `n`. Add some if statements so that the program randomly prints out “Heads” or “Tails”. Run the program several times to make sure that you eventually get both outcomes.

CoinFlip.java | Sample solution

```
import java.util.Random;

class CoinFlip {

        public static void main(String[] args) {
    // create a new random number generator object
    Random rand = new Random();

    // pick a random number between 0 and 1:
    int n = rand.nextInt(2);
    System.out.println("Picked random value " + n);

  }
}
```



`else` and `else if` statements

Java also has `else` and `else if` statements; they work as you expect from other languages. (Python uses `elif`, but like C and Javascript, Java uses the syntax `else if`.)

In the coin flip exercise, replace the second if statement with an `else` statement, and in the thermometer exercise, write an if-ladder that uses `else if` statements.