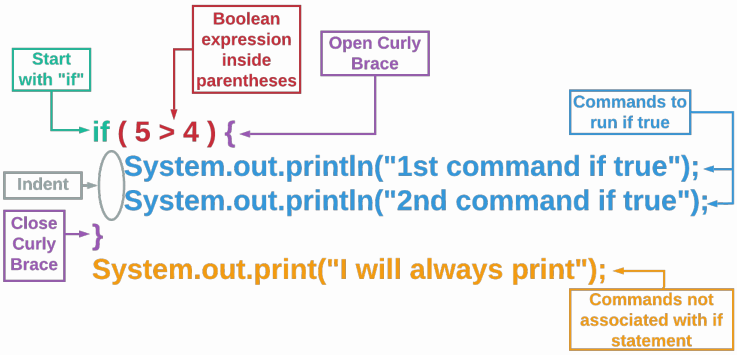
Learning Objectives - If Statement

- **Describe if statement syntax**
- **Use a conditional statement to make decisions**
- **Use two or more boolean expressions in a if statement**
- **Identify when to use compound conditionals and when not to use them**

If Statement Syntax

If Statement Syntax

Conditionals are pieces of code that make a decision about what the program is going to do next. The most common conditional is the if statement.



If Statement Syntax

If statements in Java must contain the following items:
* the keyword `if`
* a boolean expression in parentheses
* curly braces surrounding all lines of code that will run if the boolean expression is true

It is best practice to also indent the lines of code inside the curly braces to visually differentiate them from the commands that will always run.

```
if(5 > 4) {
  System.out.println("1st command if true");
  System.out.println("2nd command if true");
}
System.out.println("I will always print!");
```

Code Visualizer

If Statement

If Statement

If statements test to see if a certain condition is true. If yes, then a specific commands are run. The simple if statement does not do anything if the boolean expression is false.

```
if (7 != 10){
    System.out.println("The above statement is true");
    System.out.println("The above statement is still true");
}
System.out.println("This is not related to the if statement");
```

Code Visualizer

challenge

What happens if you:

- Change `!=` to `==`?
- Change `7 == 10` to `true`?
- Change `true` to `false`?
- Remove the curly braces `{}` with the condition set to `false`?

Code Visualizer

Testing Multiple Cases

You will find yourself needing to test the same variable multiple times. Be sure that you set up your conditionals to test **all** possible values of the variable.

```java
int grade = 90;

if(grade > 70) {
    System.out.println("Congrats, you passed the class");
}

if(grade < 70){
    System.out.println("Condolences, you did not pass the
        class");
}
```

Code Visualizer

challenge

What happens if you:

- Change `grade` to `60`?
- Change `grade` to `70`?
- Change `grade > 70` to `grade >= 70`?

Code Visualizer

Compound Conditional Statements

Compound Conditional Statements

Conditional statements (if statements) are used to match an action with a condition being true. For example, print `Even` if a number is even. If you want to test for a number being even and greater than 10, you will need two conditionals.

```
int num = 16;

if (num % 2 == 0 && num > 10) {
    System.out.println("Even and greater than 10");
}
```

challenge

What happens if you:

- Change `num` to `8`?
- Change `&&` to `||`?
- Change `==` to `!=`?

Why Use Compound Conditionals

Both code snippets below do the same thing: Ask if `my_var` is greater than 15 and if `my_var` is less than 20. If both of these are true, then Java will print the value of `my_var`.

```
int my_var = 19;

if (my_var > 15) {
   if (my_var < 20) {
      System.out.println(my_var);
   }
}
```

```
int my_var = 19;

if (my_var > 15 && my_var < 20) {
   System.out.println(my_var);
}
```

.guides/img/compound-conditional

The code on the left is a **nested** if statement - which means an if statement is *inside* another if statement.

The code with the compound conditional (on the right) has fewer lines of code, and is easier for a human to read. In fact, it almost reads like a sentence.