



**FREE eBook**

# LEARNING if-statement

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#if-

statement

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with if-statement.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Introduction to the if statement.....	2
Usage of relational operators.....	2
The else and else if statements.....	3
<b>Credits.....</b>	<b>5</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [if-statement](#)

It is an unofficial and free if-statement ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official if-statement.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with if-statement

## Remarks

This section provides an overview of what if-statement is, and why a developer might want to use it.

It should also mention any large subjects within if-statement, and link out to the related topics. Since the Documentation for if-statement is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

The if statement does not need any particular installation or setup.

### Introduction to the if statement

The if statement is a conditional statement that allows a program to enter or not a specific section of code depending if the condition(s) of the statement are met or not. It can be found in mostly all the existing programming languages.

The if statement will usually take the following shape:

```
if(statement)
{
    // Code to execute
}
```

The code in the brackets will only be executed if the statement is true. If that's not the case, the code section included in the if section will be ignored, and the program will continue without executing the corresponding code.

### Usage of relational operators

A statement is usually a test on a variable or the return value of a function. To test those values, we can use some relational operators:

Operator	Meaning	Example
==	Equal to	1 == 1 is TRUE, 1 == 2 is FALSE
!=	Not equal to	1 != 2 is TRUE, 1 != 1 is FALSE
<	Less than	1 < 2 is TRUE, 2 < 1 is FALSE

Operator	Meaning	Example
>	Greater than	2 > 1 is TRUE, 1 > 2 is FALSE
<=	Less than or equal to	2 <= 2 is TRUE, 2 <= 3 is TRUE, 3 <= 2 is FALSE
>=	Greater than or equal to	2 >= 2 is TRUE, 3 >= 2 is TRUE, 1 >= 2 is FALSE

If we are taking the following example:

```
a = 5;

if(a < 6)
{
    // Some code
}
```

Here the value of the variable **a** is inferior to 6. So the statement is true: the code will be executed.

## The else and else if statements

It is possible to ask a program to execute a specific section of code only if an if statement is considered false. For this, we use the **else** key word.

```
if(statement)
{
    // Code to execute if the statement is true.
}
else
{
    // Code to execute if the statement is false.
}
```

Both code sections will never be executed together. The first section (the if one) is only executed if the statement is true, while the section section (the else one) is only executed if the statement is false.

It is also possible to ask, if a statement has not been verified, to verify another one. For this, we use the **else if** key words. This statement work the exact same way as a regular if statement, except that the test is only executed if the previous statement is considered false.

```
if(statement)
{
    // Code to execute if the statement is true.
}
else if(another_statement)
{
    // Code to execute if the second statement is true.
}
```

The same way as before, both code will never be executed together. If the first statement is true, the second test will simply be skipped, and the first section of code will be executed. If the first

statement is false, the second statement is verified, and the second section is executed only if this statement is true.

It is possible to add as many **else if** sections one after another as needed to test different statements. It is also possible to add an **else** section at the end of all the else if sections that will be executed only if all the statements are false.

```
if(statement)
{
    // Code to execute if the statement is true.
}
else if(second_statement)
{
    // Code to execute if the second statement is true.
}
else if(third_statement)
{
    // Code to execute if the third statement is true.
}
else
{
    // Code to execute if none of the three above statements are true.
}
```

Only one code section will be executed. At the moment one statement is verified, all the following sections are skipped and won't be executed.

Read [Getting started with if-statement online](https://riptutorial.com/if-statement/topic/7947/getting-started-with-if-statement): <https://riptutorial.com/if-statement/topic/7947/getting-started-with-if-statement>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with if-statement	<a href="#">Community</a> , <a href="#">Isuka</a> , <a href="#">sandyJoshi</a>