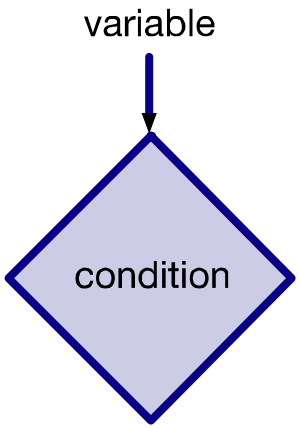
If else statements

`if` statement: Only execute some code if a condition is met (1/7)

variable

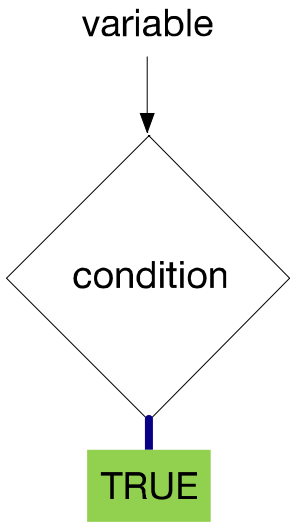
`if` statement: Only execute some code if a condition is met (2/7)




```
x <- -3

if (x < 0)
```

The *condition* tests whether x is below zero

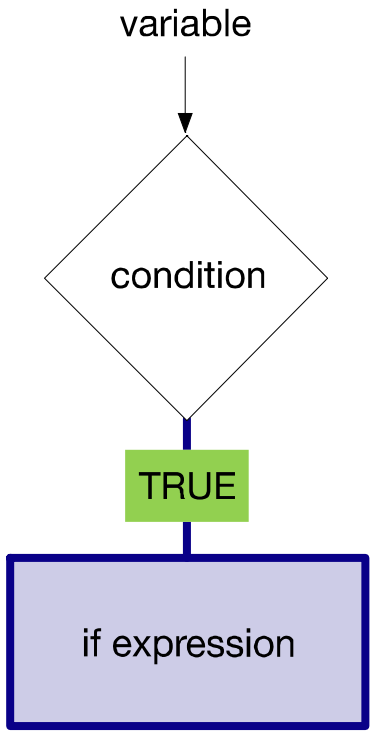
`if` statement: Only execute some code if a condition is met (3/7)




```
x <- -3

if (x < 0)
```

Decide what happens when the *condition* is met, i.e. is `TRUE`

`if` statement: Only execute some code if a condition is met (4/7)

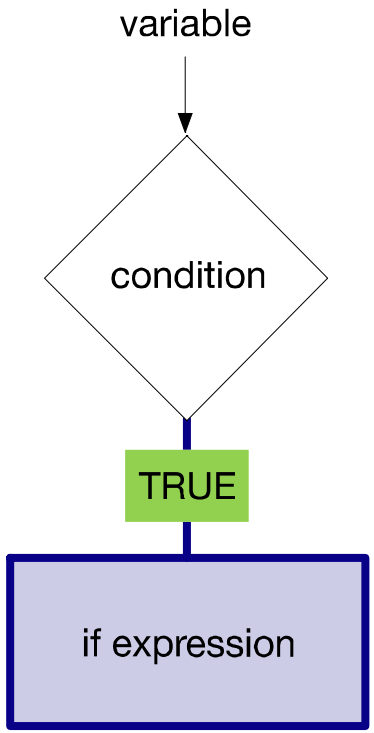



```
x <- -3

if (x < 0){
    print("x is a negative number")
}
```

If the *condition* is `TRUE`, this code is executed

Use `{` and `}`

`if` statement: Only execute some code if a condition is met (4/7)

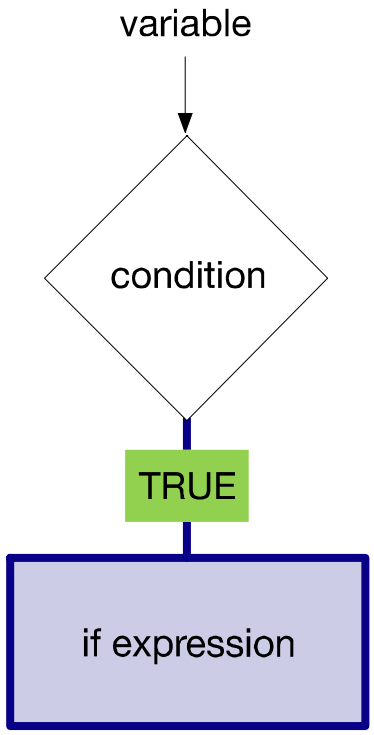



```
x <- -3

if (x < 0){
    print("x is a negative number")
}
```

If the *condition* is `TRUE`, this code is executed

Use `{` and `}`

`if` statement: Only execute some code if a condition is met (5/7)



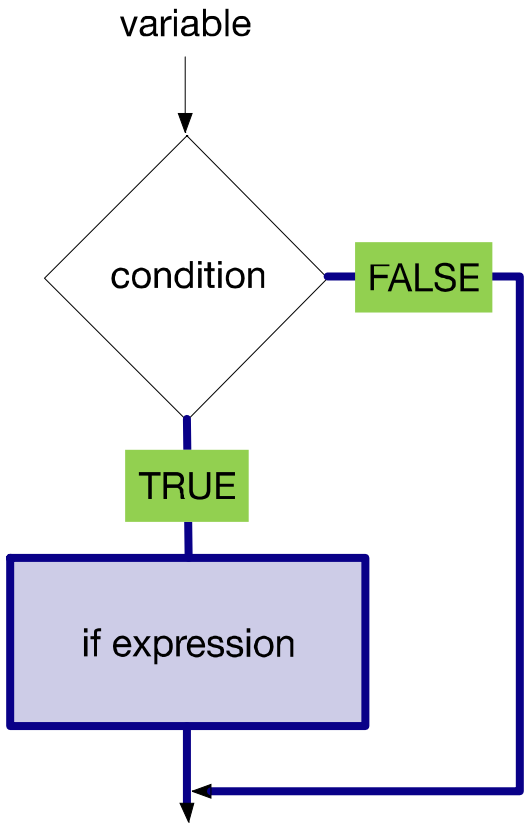

```
x <- -3

if (x < 0){
    print("x is a negative number")
}
```

1 Code is executed

2 This is the output in the Python shell

```
OUTPUT:
"x is a negative number"
```

`if` statement: Only execute some code if a condition is met (6/7)

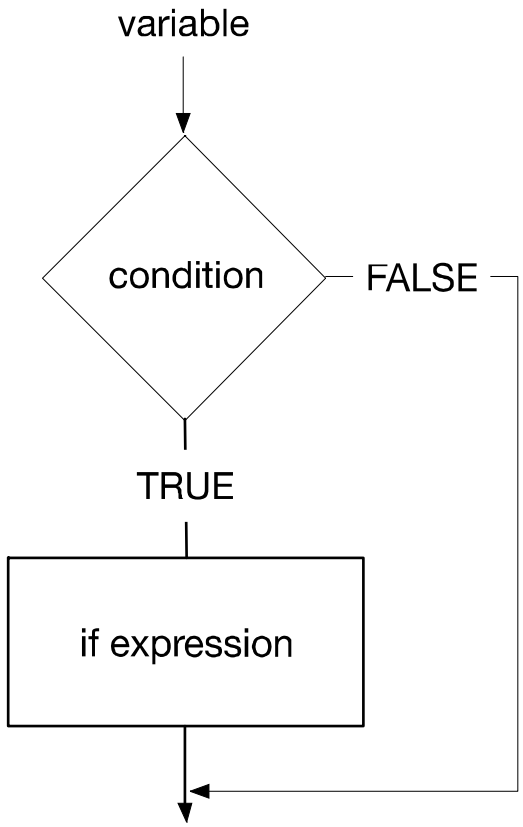



```
x <- 5

if (x < 0){
    print("x is a negative number")
}
```

OUTPUT:

In the case shown here (`x=5`), the *condition* is `FALSE`. Thus, no output is shown.

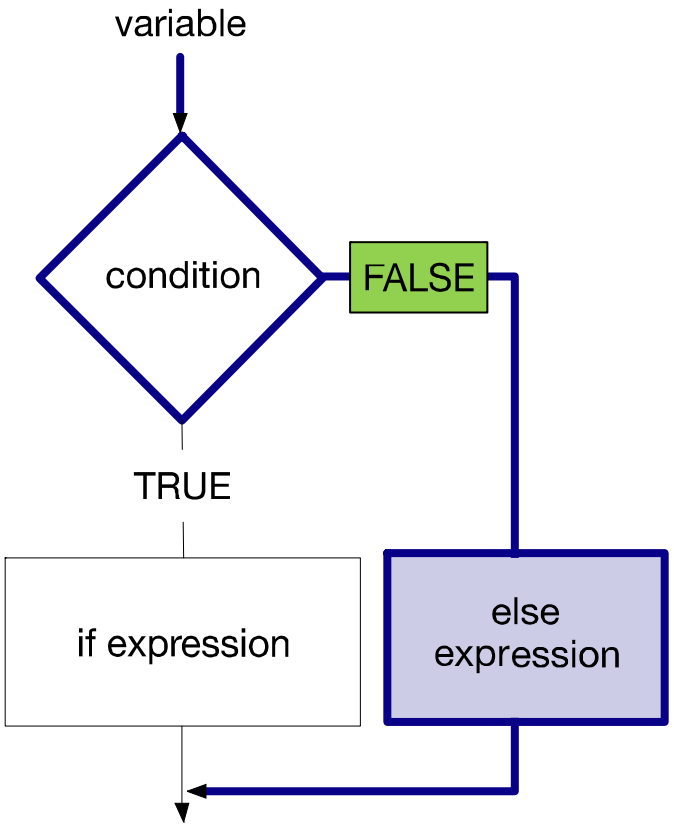
`if` statement: Only execute some code if a condition is met (7/7)



General structure

```
if (condition){
    expr
}
```

`else` statement: Only execute some code if a condition is not met (1/4)

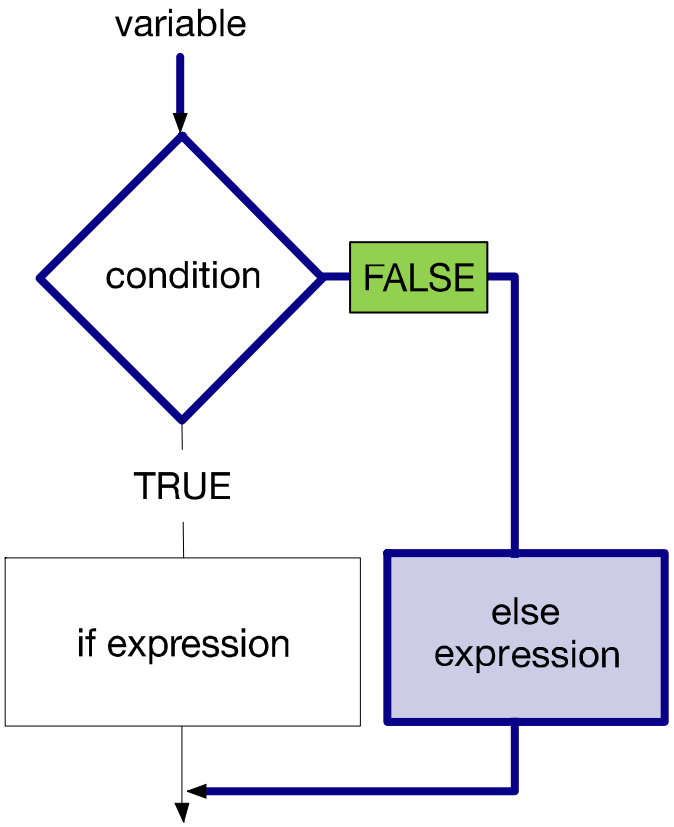



```
x <- -3

if (x < 0){
    print("x is a negative number")
} else {
    print("x is either a positive
        number or zero")
}
```

1 You need to test an `if`-condition first before specifying an `else` condition

2 Only if the `if`-condition is `FALSE`, the code of the `else`-condition is executed

`else` statement: Only execute some code if a condition is not met (1/4)

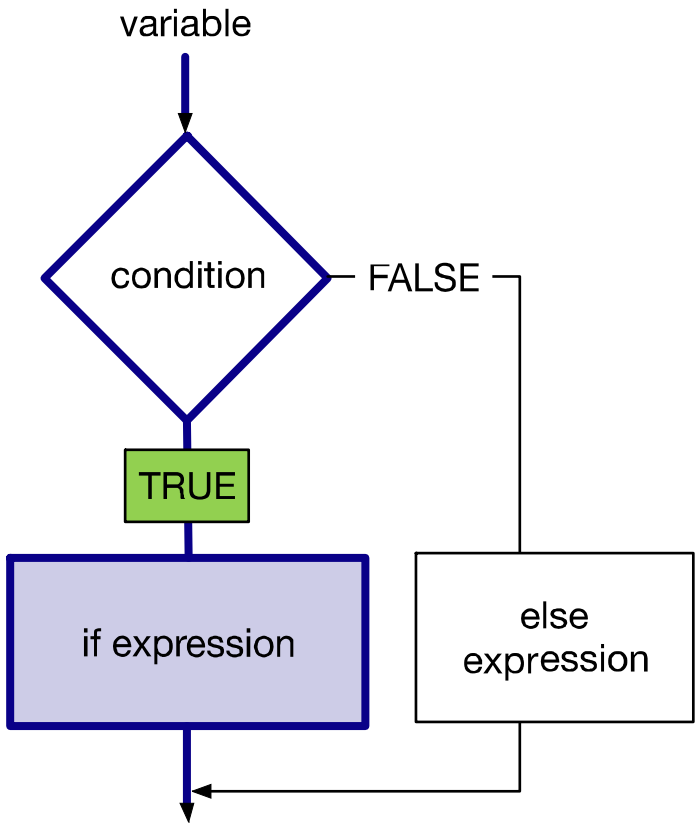


You need to test an `if`-condition first before specifying an `else` condition ❶

```
x <- -3

if (x < 0){
    print("x is a negative number")
} else {
    print("x is either a positive
        number or zero")
}
```

Only if the `if`-condition is `FALSE`, the code of the `else`-condition is executed ❷

`else` statement: Only execute some code if a condition is not met (2/4)




```
x <- -3

if (x < 0){
    print("x is a negative number")
} else {
    print("x is either a positive
        number or zero")
}
```

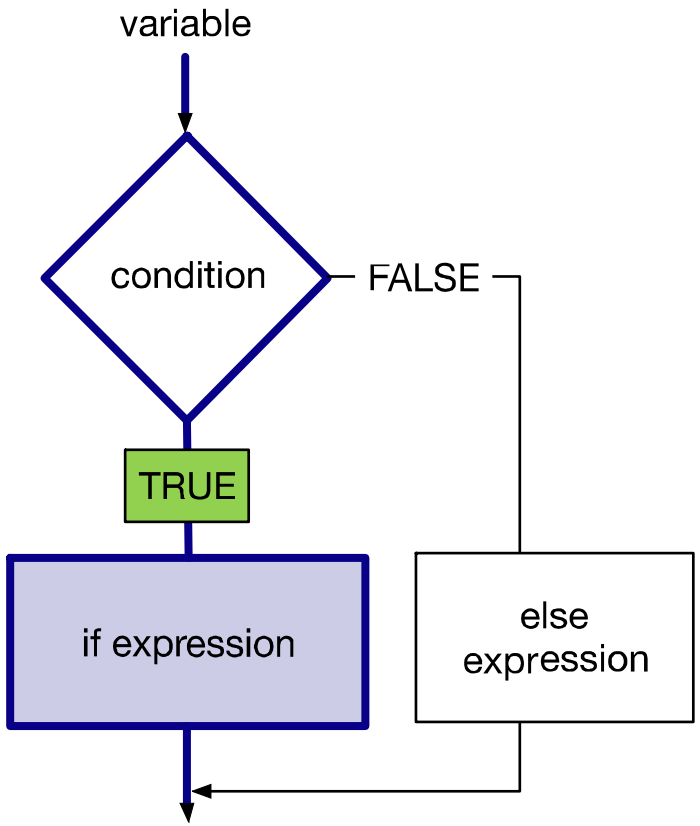
1. Since `x=-3`, *condition* is `TRUE`
2. Expression is executed
3. Expression is not executed because we already executed the `if`-expression

OUTPUT:

```
"x is a negative number"
```

4. Still the same output

`else` statement: Only execute some code if a condition is not met (2/4)




```
x <- -3

if (x < 0){
    print("x is a negative number")
} else {
    print("x is either a positive
        number or zero")
}
```

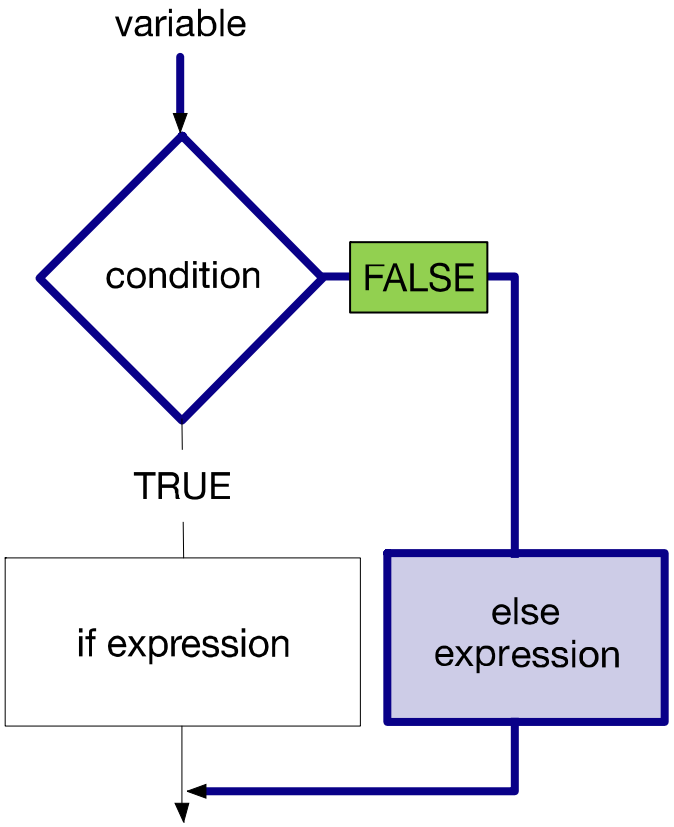
1. Since `x=-3`, *condition* is `TRUE`
2. Expression is executed
3. Expression is not executed because we already executed the `if`-expression

OUTPUT:

```
"x is a negative number"
```

4. Still the same output

`else` statement: Only execute some code if a condition is not met (3/4)




```
x <- 5

if (x < 0){
    print("x is a negative number")
} else {
    print("x is either a positive
        number or zero")
}
```

1 Since `x=5`, *condition* is `FALSE`

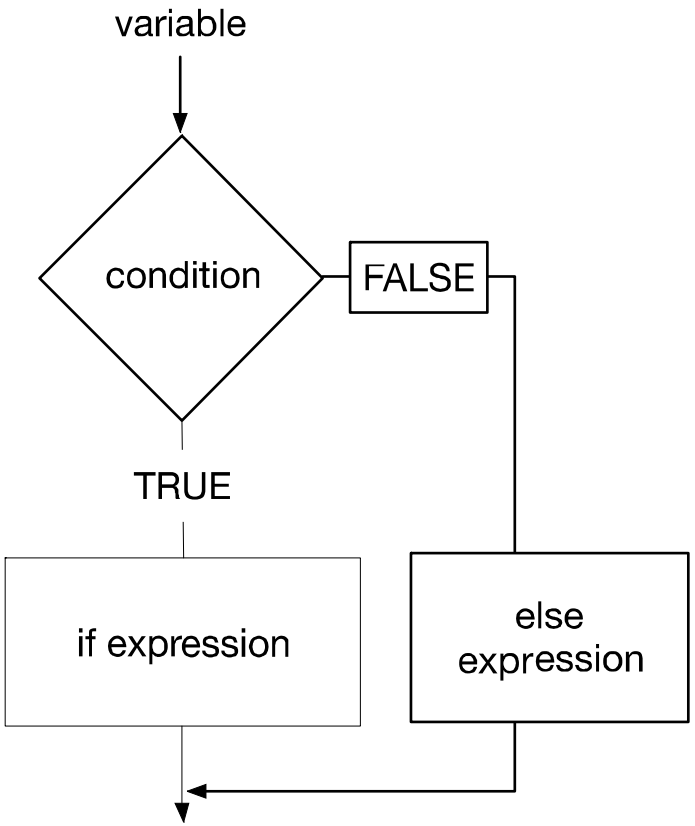
2 Expression is not executed

3 Expression is executed because the `if`-expression didn't hold

OUTPUT:

```
"x is either a positive number or zero"
```

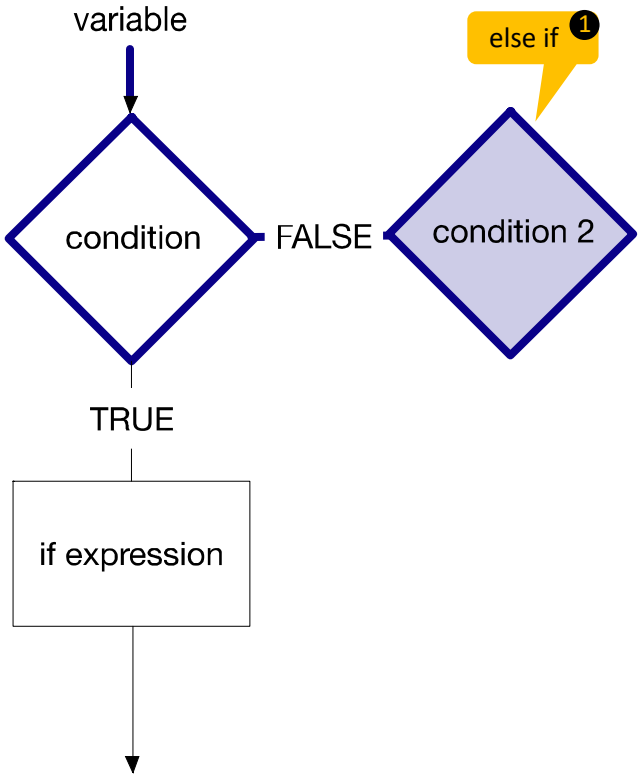
`else` statement: Only execute some code if a condition is not met (4/4)



General structure

```
if (condition){
    expr
} else {
    expr2
}
```

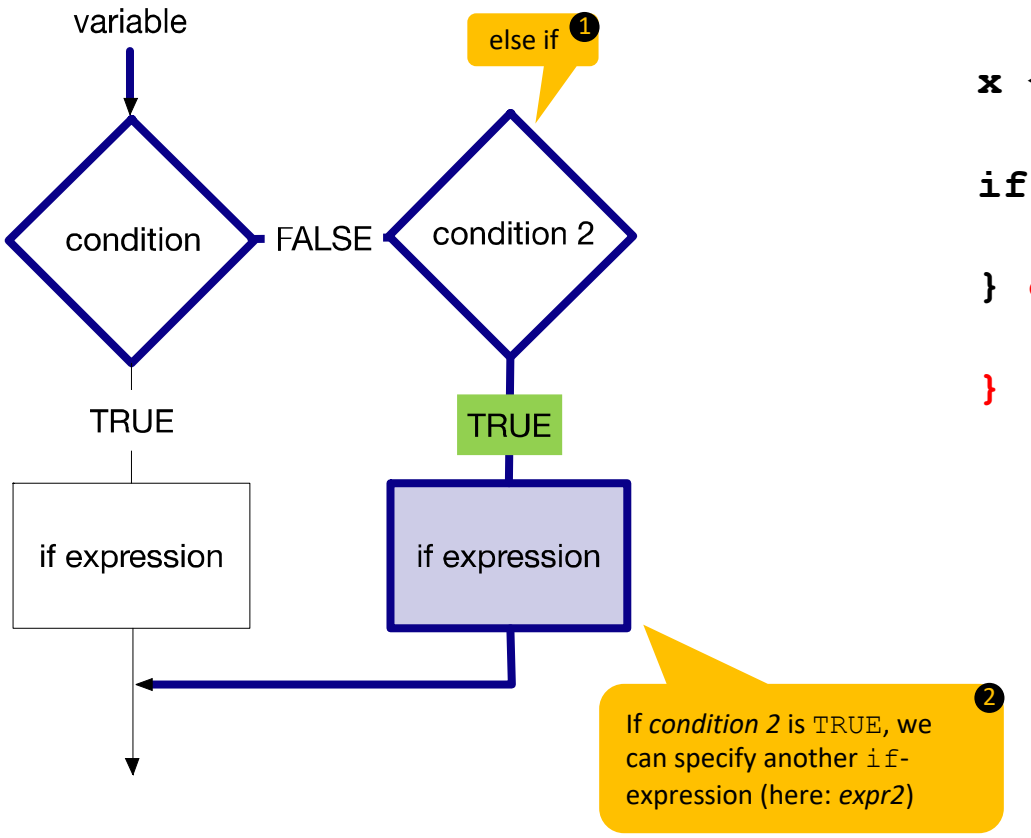
`else if` statement: Add another case (1/7)




```
x <- -3

if(x < 0){
   print("x is a negative number")
} else if (x == 0){

}
```

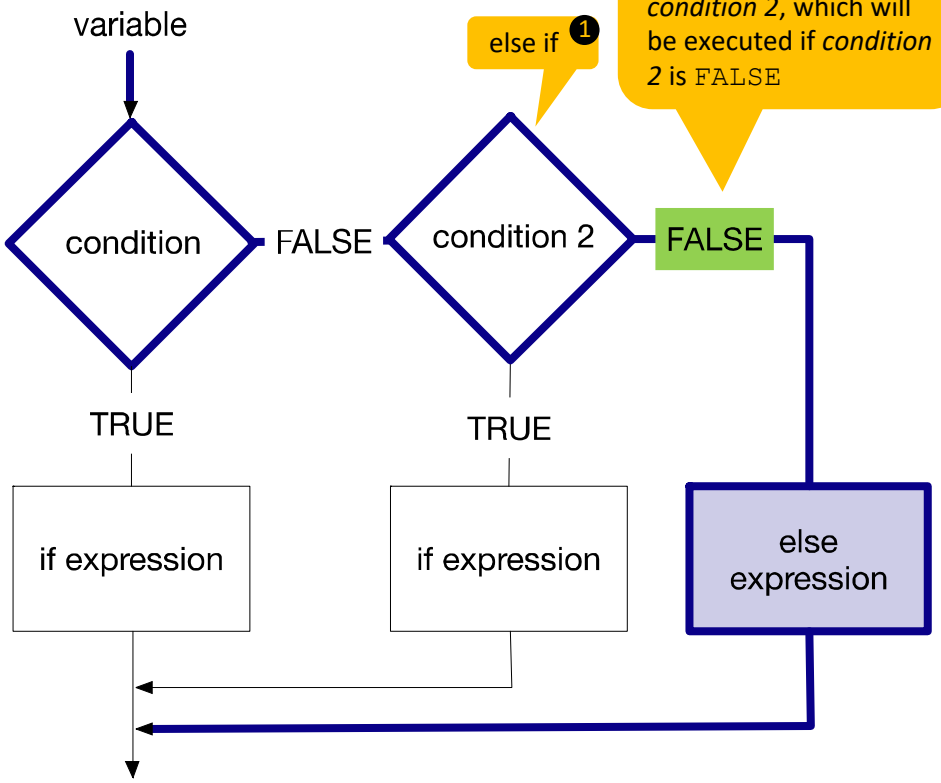
2 Additional condition which is evaluated only if *condition* is `FALSE`.

`else if` statement: Add another case (2/7)




```
x <- -3

if(x < 0){
   print("x is a negative number")
} else if (x == 0){
    print("x is zero")
}
```

`else if` statement: Add another case (3/7)



else if 1

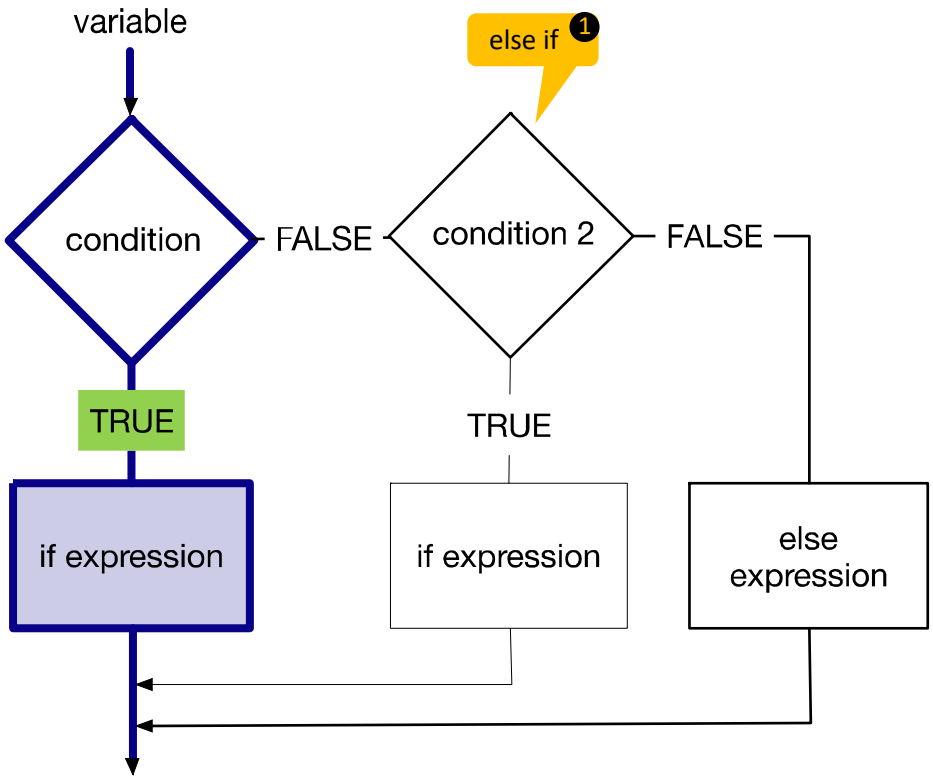
You can also specify an `else`-clause for *condition 2*, which will be executed if *condition 2* is `FALSE` 2

```
x <- -3

if(x < 0){
    print("x is a negative number")
} else if (x == 0){
    print("x is zero")
} else {
    print("x is a positive number")
}
```

Additional condition 3

This code is only executed if *condition* and *condition 2* are **both** `FALSE` 4

`else if` statement: Add another case (4/7)



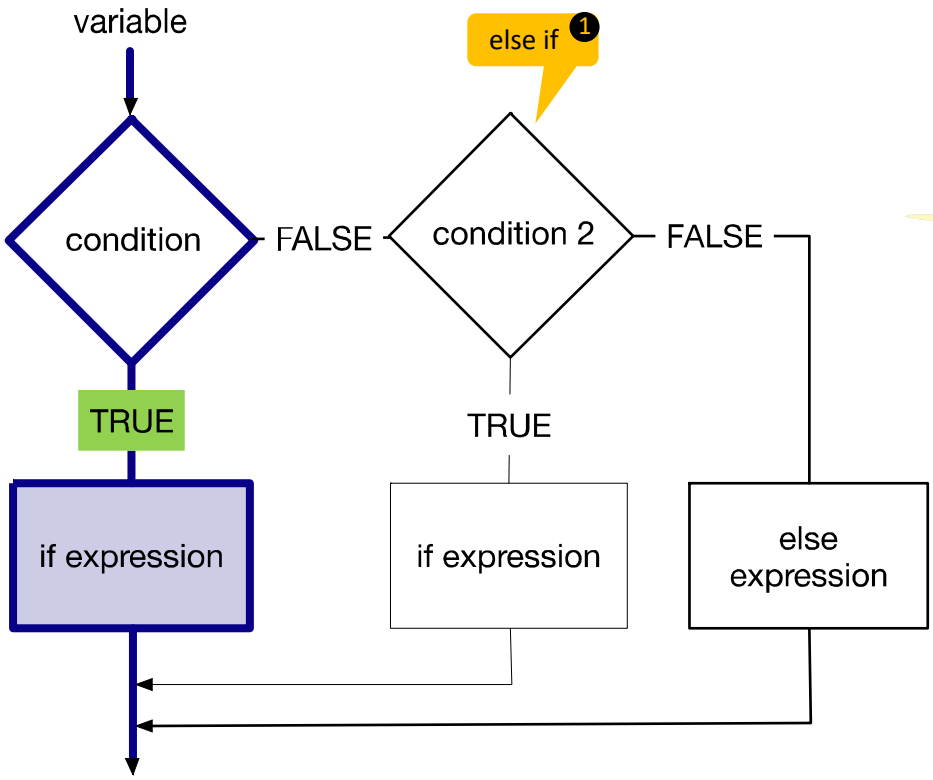

```
x <- -3

if(x < 0){
   print("x is a negative number")
} else if (x == 0){
   print("x is zero")
} else {
   print("x is a positive number")
}
```

1. else if
2. *condition* is TRUE
3. *condition 2* is skipped
4. else expression is skipped
5. *condition* is TRUE, if-expression is executed

OUTPUT:

```
"x is a negative number"
```

`else if` statement: Add another case (4/7)




```
x <- -3
if(x < 0){
    print("x is a negative number")
} else if (x == 0){
    print("x is zero")
} else {
    print("x is a positive number")
}
```

2 *condition* is `TRUE`

3 *condition 2* is skipped

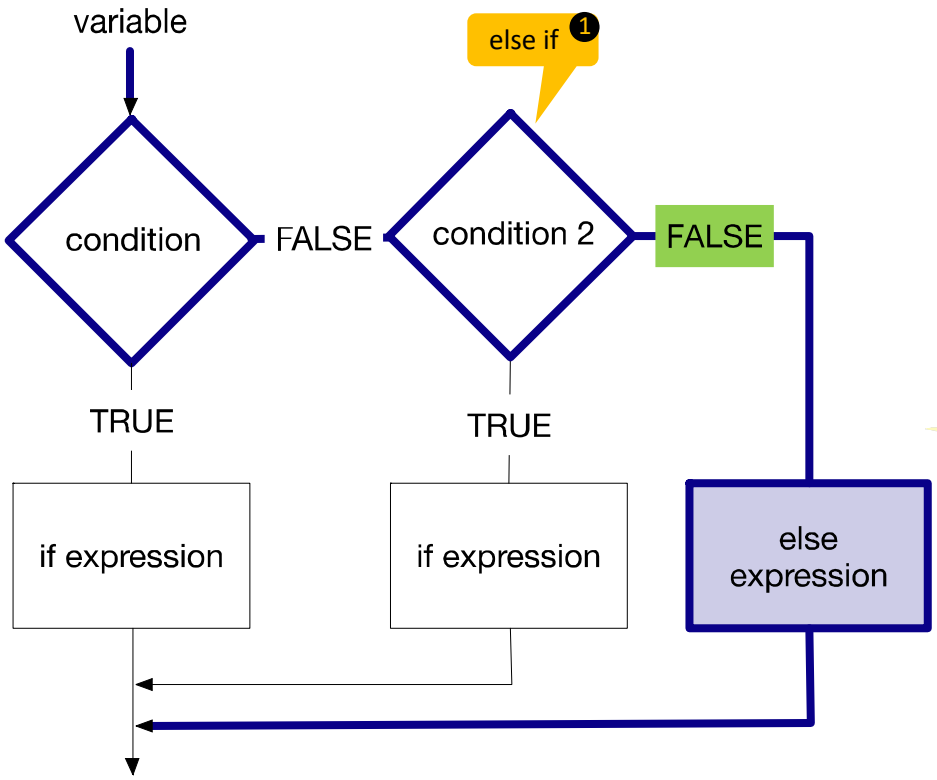
4 `else` expression is skipped

5 *condition* is `TRUE`, `if`-expression is executed

OUTPUT:

```
"x is a negative number"
```

`else if` statement: Add another case (5/7)



else if 1

```
x <- 5

if(x < 0){
   print("x is a negative number")
} else if (x == 0){
   print("x is zero")
} else {
   print("x is a positive number")
}
```

condition is FALSE 2

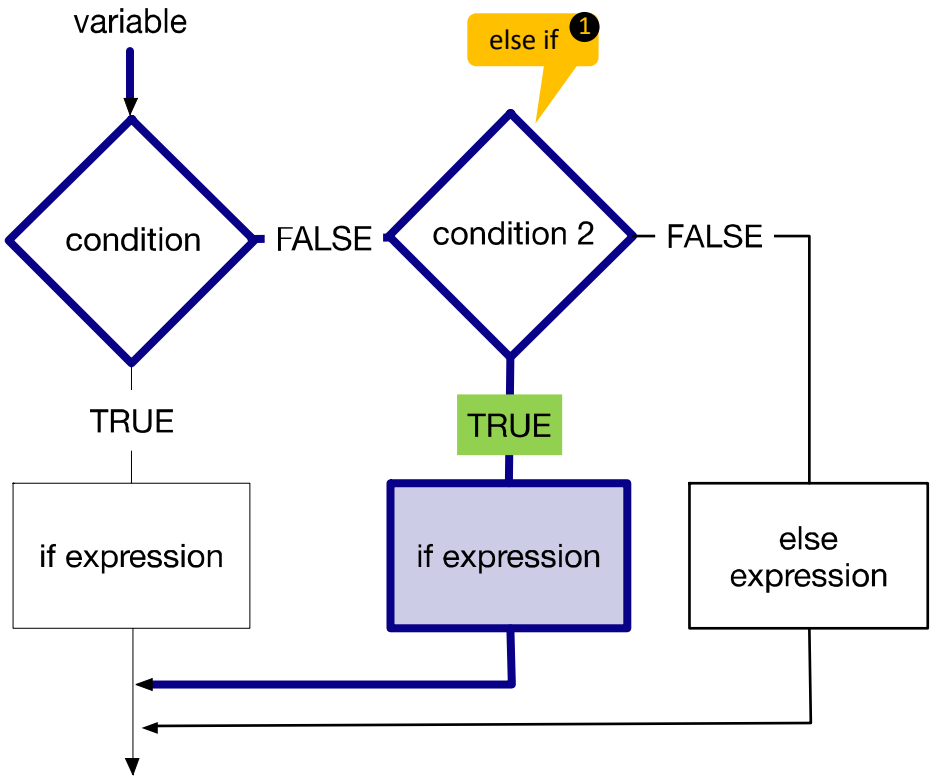
condition 2 is FALSE 3

Both conditions are FALSE. Thus, else-expression is executed 4

OUTPUT:

```
"x is a positive number"
```

`else if` statement: Add another case (6/7)




```
x <- 0

if(x < 0){
    print("x is a negative number")
} else if (x == 0){
    print("x is zero")
} else {
    print("x is a positive number")
}
```

2 *condition* is `FALSE`

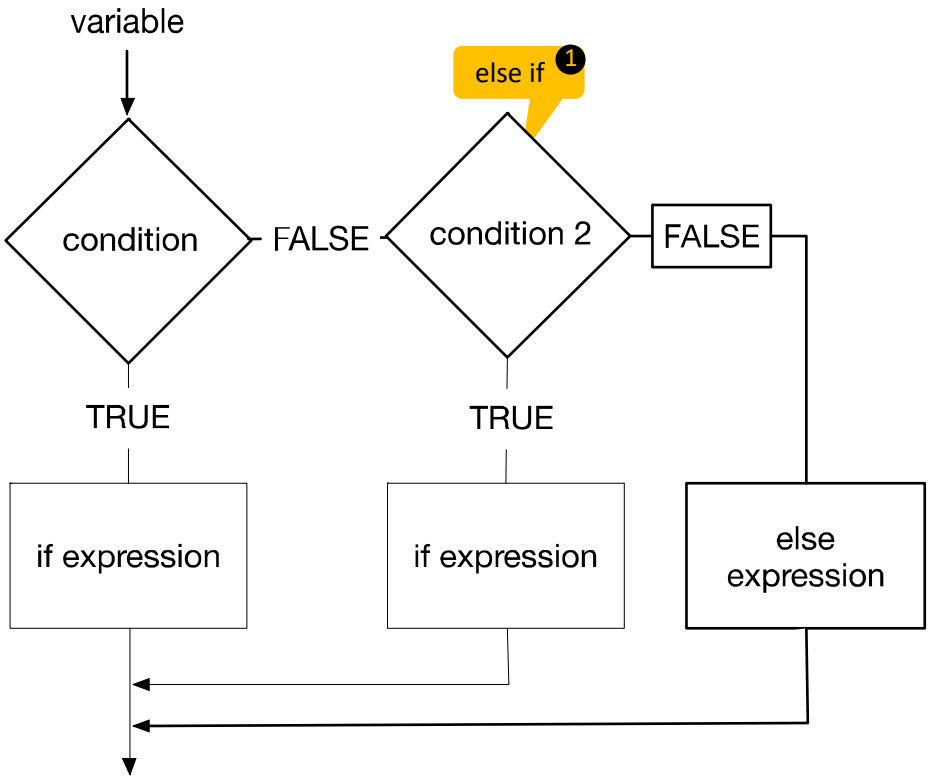
3 *condition 2* is `TRUE`

4 *condition* is `FALSE`, but *condition 2* is `TRUE`, `else if`-expression is executed

OUTPUT:

```
"x is zero"
```

`else if` statement: Add another case (7/7)



General structure 2

Additional condition 3

```
if (condition){
    expr
} else if (condition2){
    expr2
} else {
    expr3
}
```

4 Executed if *condition* and *condition 2* are **both** `FALSE`