Taking Decisions: if / else Statements

Taking Decisions: if / else Statements

Let's now make coding more interactive by enabling our code to make decisions, making it appear more realistic.

Suppose we want to write a program that checks whether a person is allowed to start taking a driver's license or not. If the person is eligible, the program will print a confirmation message to the console. Otherwise, it will print how many years remain until the person can start the process.

Let's start by defining an **age** variable and set it to 19. We already know how to check if this age is at least 18, which is the legal required age to start a driving license, at least here in Europe.

We create a variable named **isOldEnough** to describe exactly what we want here. To check if the age is at least 18, we use the condition **age >= 18**. This condition will evaluate to **true** if the age is 18 or above, and **false** otherwise.

Now, we have a Boolean value stored in **isOldEnough**. We can use this to make decisions using an **if** statement. The **if** statement works as follows: write **if** followed by parentheses containing a condition. If this condition evaluates to **true**, the code block enclosed in curly braces **{}** will be executed.

In this case, we use the condition **isOldEnough** inside the **if** statement. Whenever this value is **true**, the code inside the block will execute. If it is **false**, the block will be skipped.

Inside the **if** block, we log to the console: "Sarah can start her driving license" .

When we run this code with **age** set to 19, the output is: "Sarah can start her driving license" because 19 is above 18, making the condition true and executing the block.

If we change **age** to 15, the condition **age >= 18** becomes false, so the code inside the **if** block will not execute, resulting in no output.

Typically, we write the condition directly inside the **if** parentheses, like **if (age >= 18)**, instead of using an intermediate variable like **isOldEnough**.

Now, let's add an **else** block. The **else** block executes whenever the **if** condition is false. We write **else** immediately after the **if** block, followed by another code block enclosed in curly braces.

For example, if the age is below 18, the **if** condition is false, so the **else** block will execute. Inside the **else** block, we calculate how many years are left until the person can start taking the license by computing **yearsLeft = 18 - age**.

We then log a message using a template literal: **Sarah is too young. Wait another ${yearsLeft} years.** This will display the number of years left until eligibility.

For example, if **age** is 15, the output will be: "Wait another 3 years" because $18 - 15 = 3$.

Keep in mind that the **else** block is optional. If omitted, and the **if** condition is false, the program simply skips the **if** block and continues with the next line of code.

The **if**, **else** statement is a control structure, which allows us to control the flow of code execution. Instead of executing code linearly, we can decide which blocks to execute based on conditions.

This control over execution flow is fundamental in programming, enabling dynamic and responsive code behavior.

Let's create another example where we conditionally assign a variable. We define a variable **birthYear** and want to determine the century in which the person was born. For example, 1998 is in the 20th century, while 2015 is in the 21st century.

We can write an **if**, **else** statement: if **birthYear <= 2000**, then **century = 20**; else, **century = 21**. We assume the person was not born in the 19th century or earlier.

To make this work, we must declare the variable **century** outside the **if** and **else** blocks. Variables declared inside blocks are not accessible outside those blocks.

For example, trying to access **century** before declaration results in an error: "century is not defined." To fix this, declare **century** before the conditional blocks and assign it inside them.

When **birthYear** is 1998, the output is **20**, indicating the 20th century. When **birthYear** is 2012, the output is **21**, indicating the 21st century.

Understanding the logic of the **if**, **else** statement is essential, even if the variable declaration details are not yet fully clear.

Recap

- Use **if** followed by a condition in parentheses to execute code only when the condition is true.
- The condition must evaluate to a Boolean value (**true** or **false**).
- If the condition is true, the code block inside the **if** executes.
- If the condition is false and an **else** block exists, the code inside the **else** block executes.
- The **else** block is optional.
- This control structure allows non-linear execution flow, giving more control over how code runs.

Mastering **if**, **else** statements is powerful and fundamental in programming. Once you understand this concept, you can proceed to more advanced topics and challenges.

Key Takeaways

- The **if** statement executes a block of code only when a specified condition evaluates to true.
- The **else** block executes when the **if** condition is false, providing an alternative code path.
- Conditions in **if** statements must evaluate to Boolean values (**true** or **false**).
- Variables declared inside code blocks are not accessible outside; declare variables outside blocks to use them conditionally.