



TASK

Beginner Control Structures — elif Statements

Visit our website

Introduction

WELCOME TO THE BEGINNER CONTROL STRUCTURES — ELIF STATEMENTS TASK!

In this task, you will learn about a program's flow of control. A control structure is a block of code that analyses variables and chooses a direction in which to go based on given parameters. In essence, it is a decision-making process in computing that determines how a computer responds to certain conditions.



Get in touch
Connect for support

Remember that with our courses, you're not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to Discord at <https://discord.com/invite/hyperdev> where our specialist team is ready to support you.

Our team is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!





A note from the Hyperion Team

It's more than reasonable to ask whether your job will still be around in the next decade or two. Job automation is happening across a vast range of fields. In a world of automation is programming safe? Read [this Hyperiondev blog](#) in which we look at the likelihood of programming remaining an in-demand skill.

ELIF STATEMENTS

The last piece of the puzzle when it comes to if statements is called an *elif statement*. Elif stands for “else if”. With this statement, we can add more conditions to the if statement. Therefore, we can test multiple parameters in the same statement.

Unlike the *else statement*, you can have multiple elif statements in an if-elif-else statement. If the condition of the if statement is *False*, the condition of the next elif statement is checked. If the first elif statement condition is also *False*, the condition of the next elif statement is checked, etc. If all the elif conditions are *False*, the *else statement* and its indented block of statements is executed.

In Python, *if-elif-else statements* have the following syntax:

```
if condition1:
    indented statements
elif condition2:
    indented statements
elif condition3:
    indented statements
elif condition4:
    indented statements
else:
    indented statements
```

Look at the following example:

```
num = 10

if num < 12:
    print("the variable num is lower than 12")
else:
    print("the variable num is greater than 12")
```

What happens if we want to test multiple conditions? This is where `elif` comes in.

```
num = 10

if num > 12:
    print("the variable num is greater than 12")
elif num > 10:
    print("the variable num is greater than 10")
elif num < 10:
    print("the variable num is less than 10")
else:
    print("the variable num is 10")
```

Remember that you can combine `if`, `else` and `elif` into one big statement. This is what we refer to as a *conditional statement*.

Some important points to note on the syntax of `if-elif-else` statements:

- Make sure that the `if/elif/else` statements end with a colon (the `:` symbol).
- Ensure that your indentation is done correctly (i.e. statements that are part of a certain control structure's 'code block' need the same indentation).
- To have an *elif* you must have an *if* above it.
- To have an *else* you must have an *if* or *elif* above it.
- You can't have an *else* without an *if* — think about it!
- You can have many *elif* statements under an *if*, but only one *else* right at the bottom. It's like the fail-safe statement that executes if the other *if/elif* statements fail!



A note from our coding mentor **Ridhaa**

Have you heard about Margaret Hamilton? This is the woman and engineer who took us to the moon. She wrote the code for Apollo 11's on-board flight software and, as a result of her work, she received NASA's Exceptional Space Act Award. If that's not enough, she is also credited with coining the term "software engineering". In the picture below, you will see a young Margaret standing next to the actual code she wrote to take humanity to the moon.



Margaret Hamilton

Instructions

Before you get started, read and run the **example.py** files to see the output. Feel free to write and run your own example code before doing the compulsory task to become more comfortable with Python.

Compulsory Task 1

Follow these steps:

- Create a Python file called **control.py** in this folder.
- This is going to expand on the first control structure task we created.
- Write code to take in a user's age using `input()` and store their age in an integer variable called `age`.
- Then check if the user's age is 18 or older. If the user is 18 or older, print out the message "You are old enough!".
- Else if they are over 16 but under 18, print "Almost there".
- Otherwise print "You're just too young!".
- You should use one if, elif and else statement to do this.

Compulsory Task 2

Follow these steps:

- Create a new Python file in this folder called **bmi_task.py** in this folder.
- Create a program that calculates a person's BMI
- Ask the user to enter their weight in kg and their height in m
- Use the formula below to calculate the user's BMI:
$$\text{BMI} = (\text{weight in kg}) / ((\text{height in m}) * (\text{height in m}))$$
 - If the user's BMI is 30 or greater the user is obese
 - If the user's BMI is 25 or greater the user is overweight
 - If the user's BMI is 18.5 or greater the user is normal
 - If the user's BMI is less than 18.5 the user is underweight
- Display the user's BMI and whether they are obese, overweight, normal or underweight.

Completed the task(s)?

Ask an expert code reviewer to review your work!

[Review work](#)



Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

[Click here](#) to share your thoughts anonymously.

