

If Statement fundamentals

In this example, we'll be introducing the `if` statement and how to use it in Python. We'll also be using logical and comparison operators, and booleans to make decisions in our code.

Why is this important?

If statements are the building blocks of decision-making in programming. They allow us to make decisions based on the state of our program.

This technique is used in many different ways in programming. For example, if you're building a game, you might want to check if the player has enough health to continue playing. If they don't, you might want to end the game. You might want to check if a user is logged in before allowing them to access a certain page on a website. You might want to check if a user has entered a valid email address before allowing them to sign up for a newsletter.

What are we going to do?

We're going to make a program that checks if the user is at the right concert and if they have a ticket. If they are at the right concert and they have a ticket, we'll let them in. If they are at the right concert but they don't have a ticket, we'll let them know that they need a ticket to get in. If they are at the wrong concert, we'll let them know that they are at the wrong concert.

Steps

1. Create a new file named `concert_example.py`
2. Create a variable named `concert_name` and assign it to the input from the user.

- Add the input prompt `What is the name of the concert? rememeber that what is going to be returned is a string.`

```
concert_name = input("What is the name of the concert tonight? ")
```

- The output should look like this:

```
$ python concert_example.py
What is the name of the concert tonight? taylor swift
```

3. Using if statements check if the user is in the right place.

- let's add some logic that will inform the user that if they are in the right place

```
concert_name = input("What is the name of the concert tonight? ")

if concert_name == "taylor swift":
    print("you're in the right place")
```

- You can see here we are using `==` to compare the value of `concert_name` to the string `taylor swift`. If they are the same then the code inside the if statement will run. If they are not the same then the code inside the if statement will not run. The result of an comparison operator is a boolean. In this case, it will be `True` if the values are the same and `False` if they are not.
- Note you can see here that `print("you're in the right place")` is indented. This is because it is inside the if statement. This is how Python knows that this code is part of the if statement. Most other languages you use curly braces so you might be used to that. Python uses indentation instead.
- Now let's run our program a couple of times. First, let's run it with the correct concert name.

```
$ python concert_example.py
What is the name of the concert tonight? taylor swift
you're in the right place
```

- You can see that the program printed out `you're in the right place` because we entered the correct concert name. The program compared `concert_name` to `taylor swift` and since they are the same it printed out the message.
- Second, let's run it with the wrong concert name.

```
$ python concert_example.py
What is the name of the concert tonight? boyz 2 men
```

- Now you can see that nothing was printed out. This is because the program compared `concert_name` to `taylor swift` and `boyz 2 men` is not equal to `taylor` and since they are not the same it didn't print out the message.

Awesome! we just did our first if statement! We see that the output depends on what the user enters.

4. Let's add an `else` statement to our `if` statement.

We aren't notifying the user if they are in the wrong place. Let's add that functionality now.

- Let's add an `else` statement to our if statement to notify the user if they are in the wrong place.

```
concert_name = input("What is the name of the concert tonight? ")

if concert_name == "taylor swift":
    print("you're in the right place")
else:
    print("this is not the concert you are looking for")
```

- Now let's input the wrong concert name.

```
$ python concert_example.py
What is the name of the concert tonight? boyz 2 men
this is not the concert you are looking for
```

- You can see here that the program printed out `this is not the concert you are looking for` because we entered the wrong concert name. The program compared `concert_name` to `taylor swift` and since they are not the same it printed out the message in the `else` statement.

5. Let's See what happens when you have the wrong indentation.

You're going to see some errors regarding indentation in your code sometimes. This is going to throw an `IndentationError` and it will tell you what line the error is on. Let's see what happens when we have the wrong indentation.

- in the code add a line that has 5 spaces in front of it. This will throw an error.

```
concert_name = input("What is the name of the concert tonight? ")

if concert_name == "taylor swift":
    print("you're in the right place")
    print("have fun!")
else:
    print("this is not the concert you are looking for")
```

- you can see here that `print("have fun!")` has 5 spaces, let's take a look at the output.

```
$ python concert_example.py
File "/path/to/your/folder/concert_example.py", line 5
    print("have fun!")
IndentationError: unexpected indent
```

- This shows us that we have an unexpected indent on line 5. This is because we have 5 spaces in front of `print("have fun!")`. Let's fix that to four spaces and run the program again.

```
$ python concert_example.py
What is the name of the concert tonight? taylor swift
you're in the right place
have fun!
```

6. Let's add some more logic to our `if` statement.

People have let you know that there's a "billie eilish" concert happening next door, tonight as well. Let's add some logic to our `if` statement to check if the user is at the right place, or let them know to go next door. We're going to do this by adding an `elif` statement to our `if` statement.

- Let's add an `elif` statement to our `if` statement to notify the user that the "billie eilish" concert is next door.

```
concert_name = input("What is the name of the concert tonight? ")

if concert_name == "taylor swift":
    print("you're in the right place")
    print("have fun!")
elif concert_name == "billie eilish":
    print("this concert is next door")
else:
    print("this is not the concert you are looking for")
```

- Let's talk about the above a bit
 - the `==` operator will always return a boolean.
 - the blocks (what is indented) will only run if the condition (what you're comparing with the `==`) is true.
 - if you have many `elif` statements, only the first block with a true condition will run. We'll look at this in the next step.
- Let's take a look at the output.

```
$ python concert_example.py
What is the name of the concert tonight? billie eilish
this concert is next door
```

7. Let's add another `elif` block and see observe the functionality.

We're going to add another `elif` block to our `if` statement and see what happens when it has the same condition as the condition above it.

- let's add that block to the if statement.

```
concert_name = input("What is the name of the concert tonight? ")

if concert_name == "taylor swift":
    print("you're in the right place")
    print("have fun!")
elif concert_name == "billie eilish":
    print("this concert is next door")
elif concert_name == "billie eilish":
    print("this code will never run")
else:
    print("this is not the concert you are looking for")
```

- You can see here that there are two conditions with the same condition. Let's take a look at the output.
- Let's take a look at the output.

```
$ python concert_example.py
What is the name of the concert tonight? billie eilish
this concert is next door
```

- You can see that even though we have two conditions that are the same, only the first one ran and there's no "this code will never run" in the output. This is because the first one was true and the second one was not. The program will only run the first block that has a true condition.
 - remove this code and then move on to the next step.

8. Let's check if the user has a ticket.

We've checked if the user is at the right place but we haven't checked if the user has a ticket. Let's add that functionality now.

- Let's get the user input if they have the ticket.

```
concert_name = input("What is the name of the concert tonight? ")
has_ticket = input("Do you have a ticket? (y/n) ")

# other code removed for brevity
```

- So now we have a variable called `has_ticket` that will either be `y` or `n`. Let's add some logic to our if statement to check if the user has a ticket. We'll need to use the `==` operator again and the `and` operator together to check if the user has the ticket.

```
concert_name = input("What is the name of the concert tonight? ")
has_ticket = input("Do you have a ticket? (y/n) ")

if concert_name == "taylor swift" and has_ticket == "y":
    print("you're in the right place")
    print("have fun!")
elif concert_name == "billie eilish":
    print("this concert is next door")
else:
    print("this is not the concert you are looking for")
```

- You can see in the above that we can use the `and` operator to check if both conditions are true. If they are both true then the code inside the `if` statement will run. If one of them is false then the code inside the `if` statement will not run.
 - Note here that the `or` operator wouldn't work here because we want both conditions to be true. `or` only needs one of them to be true.
- Let's take a look at the output if we have a ticket.

```
$ python concert_example.py
What is the name of the concert tonight? taylor swift
Do you have a ticket? (y/n) y
you're in the right place
have fun!
```

- let's take a look at the output if we don't have a ticket.

```
$ python concert_example.py
What is the name of the concert tonight? taylor swift
Do you have a ticket? (y/n) n
this is not the concert you are looking for
```

- change this to an `or` and give the wrong input and observe what happens in the code.

9. Let's make `elif` statement to handle if users don't have a ticket

- let's add this to our code below

```
concert_name = input("What is the name of the concert tonight? ")
has_ticket = input("Do you have a ticket? (y/n) ")

if concert_name == "taylor swift" and has_ticket == "y":
    print("you're in the right place")
    print("have fun!")
elif concert_name == "taylor swift" and has_ticket == "n":
    print("sorry you need a ticket to get in")
elif concert_name == "billie eilish":
    print("this concert is next door")
else:
    print("this is not the concert you are looking for")
```

- now let's take a look at the output if we don't have a ticket.

```
$ python concert_example.py
What is the name of the concert tonight? taylor swift
Do you have a ticket? (y/n) n
sorry you need a ticket to get in
```

Conclusion

In this example, we've covered the fundamentals of if/elif/else statements.

We've also covered comparison operators, logical operators, and booleans. These are the building blocks of if statements.

We'll be using all of these concepts in tandem a lot in our code throughout this course.