Match Case

Why is this important?

This is a new feature in Python 3.10. It's a new way to write `if` statements. It's a little bit more concise and it's a little bit more readable.

This is a data structure that allows us to match a value to a pattern, but it's handy for cases where we have a lot of `if` and `elif` statements checking for the same value.

What are we going to do?

We're going to take a letter grade and convert it to a Grade Point Average (GPA). We're going to use the `match` statement to do this.

Steps

1. Create a new file called `grade_match_case.py`

2. Get the user input in that file and store it in a variable called `letter_grade`

```
# Get user grade
letter_grade = input("Enter your letter grade: ")
```

3. Create a `match` statement that will convert the letter grade to a GPA and print it out.

- Let's convert the user input to uppercase so we don't have to worry about case sensitivity.

```
# Get user grade
letter_grade = input("Enter your letter grade (A, B, C, D): ")


# Convert letter grade to uppercase
letter_grade = letter_grade.upper()
```

- Let's create our match case so that we can convert the letter grade to a number grade. Now `match` is a good keyword to use here because we're matching a value to a pattern over and over again.

```python
# Get user grade
letter_grade = input("Enter your letter grade (A, B, C, D): ")


# Convert letter grade to uppercase
letter_grade = letter_grade.upper()


# Convert letter grade to number
match letter_grade:
    case "A":
        gpa = 4.0
    case "B":
        gpa = 3.3
    case "C":
        gpa = 2.5
    case "D":
        gpa = 2.0
    case "F":
        gpa = 1.0
    case _: # this is the
        print("could not determine numeric grade")
        gpa = 0.0


# Print numeric grade
print(f"Your GPA is {gpa}")
```

- Observe a few things above here.

  - Each `case` is indented and ends with a colon `:`.
  - If the case is matched then it will execute the code below it that is indented.
    - if `letter_grade` is `"A"` then `gpa` will be `4.0`
    - if `letter_grade` is `"B"` then `gpa` will be `3.3` etc.
  - There's a special case called _ which is the default case. This will execute if none of the other cases match.
    - if `letter_grade` is not `"A"`, `"B"`, `"C"`, `"D"`, or `"F"` then it will print `"could not determine numeric grade"` and set `gpa` to `0.0`

- Run the program in your terminal and test it out. Try entering different values for `letter_grade` and see what happens. As well try to enter lowercase grades and you'll see that it will still work.

- Here's an example of what the output should look like.

```
$ python grade_match_case.py
Enter your letter grade (A, B, C, D): a
Your GPA is 4.0
```

- Here's an example of what the output should look like if you enter a grade that is not in the match case.

```
$ python grade_match_case.py
Enter your letter grade (A, B, C, D): cats
could not determine numeric grade
Your GPA is 0.0
```

Conclusion

Match case is a powerful new feature in Python 3.10. It's a new way to write `if` statements. When you have many `if` statements checking for the same value, it's a good idea to use a `match` statement instead.

In the future we're going to see more example of how to use `match` statements with other data structures like lists and dictionaries (which we haven't covered yet).