LESSON : 2

Data & Functions._

ANNOUNCEMENTS

Before we start, let's see if we have any codewizards announcements.

Click here to go to the notice board.

WHAT WE DID IN LAST CLASS

Discussed the importance of programming in today's world.

Learned about different ways we can use the print statement in python.

Used print statements to create a receipt generator in python.

HOMEWORK SHOWCASE

Any doubts about homework or from the previous class?

Ferrari GTB

Ferrai 488 GTB comes with a large air intake scallop divided by a splitter into two separate sections.
The front spoiler is wider and there's a new blown spoiler which helps improve downforce.
The tail lamps are tweaked but still remind you of the 458.
Inside, it looks quite familiar and comes with a string of new equipment including new satellite control clusters, angled air vents and a slightly revised instrument panel. Additionally, it also gets body-hugging seats along with a new interface and graphics for the infotainment system.

```
                 __.. ' _.\""- .`-._
 _____.-'          (_) |     \          ` \`-._
/_          --------------'-------\---....______\__`.   -..___
| T        _.----._                Xxx|x...          |        _..._ `-. _
| |      .'  ..--..  `.            XXX|XXXXXXXXXxx==  |      .'---..`.     -._ `-.
\_j    /  /   __    \  \           XXX|XXXXXXXXXX==  |     / /    __   \ \        `-.
 _|    |  |  /  \   |  |           XXX|"""'           |    / |   /  \   | |          |
|__\_j  |  \__/   |  L_________|_______________|____j |   \__/   | L_________J
        `'\ \       / ./_____________________________________\ \       / /___________\
          `.`----'.'   dp                                      `.`----'.'
            `'                                                   `'
```

EVER NOTICED THAT EVERY OBJECT IN REAL WORLD HAS TWO THINGS:

It has **values**:

- color: red
- Year Launched: 2018
- Customer Rating: 8.3
- Sunroof: True

For example:
A Car

It can perform **actions** like:

- Play horn
- Accelerate
- Apply Brake
- Shift Gears

CAN YOU TELL THE VALUES AND ACTIONS OF A SMARTPHONE?

TYPES OF DATA:

Data can be classified into following four types:

STRING

This includes all the non-numeric data, like the color of car: `red`

INTEGER

This includes all the numeric data, like the launch year of car: `2018`

FLOAT

This includes all the decimal data, like the customer rating of car: `8.3`

BOOLEAN

This type of data can only have two values, true or false. e.g. Car Sunroof: `True`

PROGRAMMING SOLVES REAL WORLD PROBLEMS

So the code we write can also be categorized into values (data) and actions (functions).

```
print("anything you want")
```

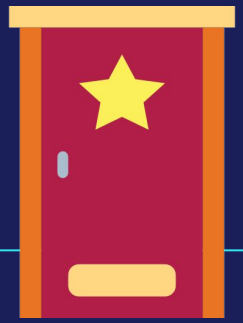
Data & functions in the above line of code would be:

```
print()
```

FUNCTION

```
"anything you want"
```

DATA

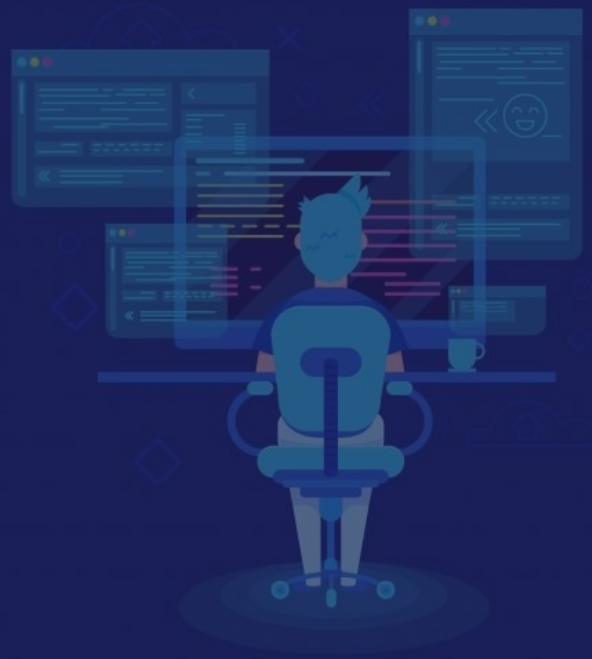


Project of the day

CELEBRITY FACTS

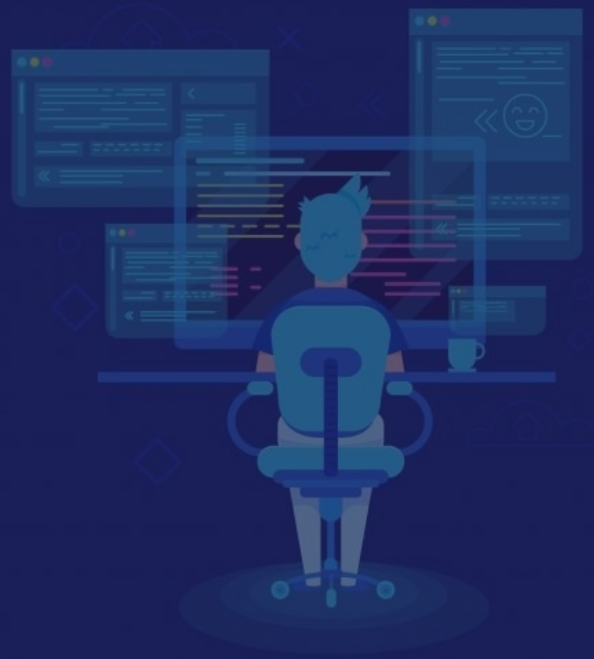
WHAT'S ALREADY DONE:

- ☑ Added python shebang and HTML content type.
- ☑ Imported magicwand for clear output.
- ☑ Added a multiline print statement.



Exercise: 1

Fill in the blanks in the print statement with celebrity facts.



CHEAT SHEET

In file facts.py

- Inside the **print** statement, fill the following details of your favorite celebrity:
  - Name
  - Birth year
  - Total no of movies
  - First movie
  - First movie rating

You can fill it with anything you want.

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

print("""
Celebrity facts about [Some name here]
- [some name here] was born in [birth year]
- [some name here] has done a total of [some number] movies.
- [some name here]'s first movie was [movie name here].
- [movie name here] has a rating of [movie rating here]
""")
```

SOLUTION

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand


print("""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
""")
```

CAN YOU POINT OUT DATA AND FUNCTIONS IN OUR CURRENT CODE?

facts.py

```
...
print("""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
""")
```

facts.py

```
...
print("""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
""")
```

FUNCTION

```
print()
```

DATA

```
"""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
"""
```

DATA

```
"""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
"""
```

Problem in above data:

- Did you notice, how we are writing the same celebrity name again and again?
- What if we want to change the name or we did a mistake while writing it? We will have to make changes in all the places again.

DRY (DON'T REPEAT YOURSELF)

According to the DRY principle, if we are writing the same code multiple times, we are doing something wrong.

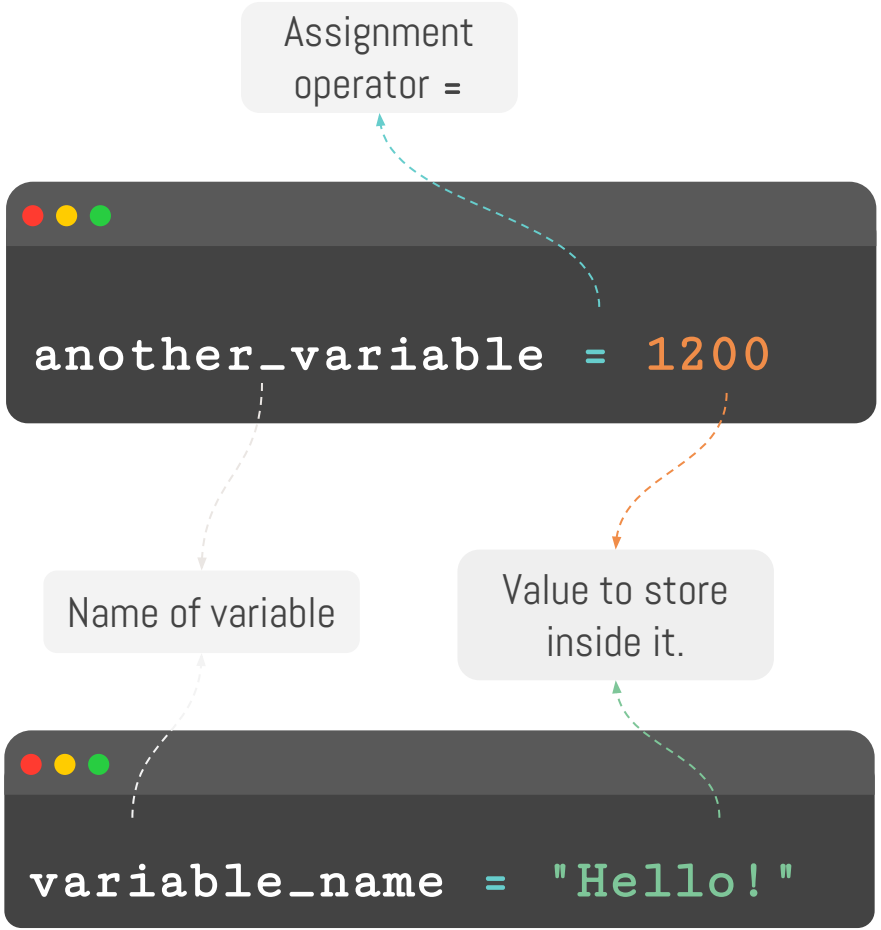
```
"""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr.'s first movie was Pound.
- Pound has a rating of 6.3
"""
```

To fix this repetition in our code, we need to use variables.

VARIABLES

Variables are like buckets of data. We can use a variable instead of writing the entire value multiple times in the code.

To create a variable, write the variable name on the left with its value on the right side. Just like we did in algebra.



RULES FOR NAMING VARIABLES

- It must start with an uppercase/lowercase letter or _

```
demo = 1200
Demo = 1200
_demo = 1200
```

```
1emo = 1200
#emo = 1200
0demo = 1200
```

- No spaces or special characters in between variable name.

```
demo_one = 1200
demoOne = 1200
DemoOne = 1200
```

```
demo one = 1200
demo&one = 1200
demo)one = 1200
```

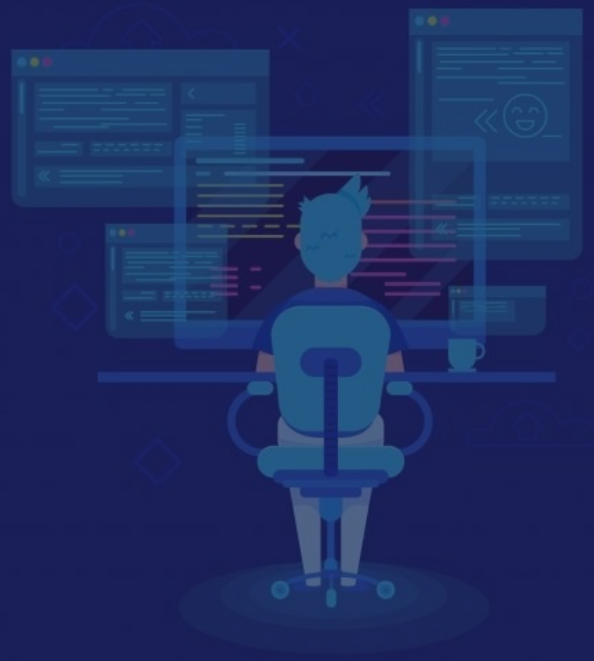
- Can not use python keywords like for, class, and, true ...

```
integer = 1200
fort = 1200
classic = 1200
```

```
int = 1200
for = 1200
class = 1200
```

Exercise: 2

Use a variable name to store the celebrity name.



VARIABLE: STRINGS

In programming, we call this non-numeric data of type **String**.
We can store the multiline data the same way using triple quotes.

We can use both single quotes and double quotes to store data. But for the sake of consistency, we will use double quotes in this course.

```
variable_name = 'Hello!'
```

```
variable_name = "Hello!"
```

```
variable_name = """Hello!
I am a multi line data"""
```

CHEAT SHEET

Step 1

In file facts.py

- Before the **print** statement, create a variable with the name **celebrity_name.**
- Store the celebrity name inside it.

You can use any other name for the variable if you want.

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

variable_name = "enter value here"

print("""
Celebrity facts about Robert Downey, Jr.
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
""")
```

PRINT STATEMENT: VARIABLES

To show the value of the variable using print statement. Just write the variable name without quotes.

Use a comma to print it along with some text.

```
print(variable_name)
```

```
print("some text here", variable_name)

print("""This is text as well
Text in second line""", variable_name)
```

CHEAT SHEET

Step 2

In file facts.py

- ❑ Remove the celebrity name from the first line of the **print** statement.
- ❑ Instead, use variable **celebrity_name** created in the previous step.
- ❑ Add triple quotes and comma before and after writing the variable name.

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

variable_name = "enter value here"

print("""
Celebrity facts about Robert Downey, Jr.""", variable_name ,"""
- Robert Downey, Jr. was born in 1965
- Robert Downey, Jr. has done a total of 79 movies.
- Robert Downey, Jr's first movie was Pound.
- Pound has a rating of 6.3
""")
```

IS IT WORKING?

We are now using a variable to show the celebrity name in the first line instead of writing it directly.

If you can see the celebrity name in your output, we can move forward.

CHEAT SHEET

Step 2

In file facts.py

- ❑ Remove the celebrity name everywhere else in the **print** statement.
- ❑ Instead, use variable **celebrity_name** created in the previous step.
- ❑ Add triple quotes and comma before and after writing the variable name.

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

variable_name = "enter value here"

print("""
Celebrity facts about""", variable_name ,"""
- """, variable_name ,""" was born in 1965
- """, variable_name ,""" has done a total of 79 movies.
- """, variable_name ,"""'s first movie was Pound.
- Pound has a rating of 6.3
""")
```

SOLUTION

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

celebrity_name = "Robert Downey, Jr."

print("""
Celebrity facts about""", celebrity_name ,"""
- """, celebrity_name ,""" was born in 1965
- """, celebrity_name ,""" has done a total of 79 movies.
- """, celebrity_name ,"""'s first movie was Pound.
- Pound has a rating of 6.3
""")
```

CONGRATULATIONS!!

You just created your first variable and made the code a little **DRY**.

Let's see if we can make the code even cleaner.

```
...
print("""
Celebrity facts about""", celebrity_name ,"""
- """, celebrity_name ,""" was born in 1965
- """, celebrity_name ,""" has done a total of 79 movies.
- """, celebrity_name ,"""'s first movie was Pound.
- Pound has a rating of 6.3
""")
```

Well, there are so many triple quotes!!

It's hard to keep track of quotes. If we add another variable, it might mess up the code.

Only if we had a better way to do this.

THE F STRING

Recently this quotes issue was fixed by introduction of **F** or **f** string in python.

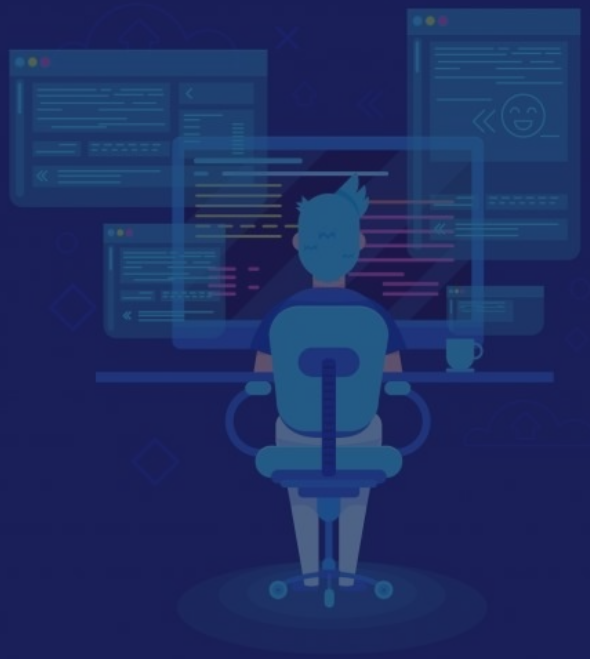
All we have to do is put an **F** or **f** before the quotes, and write variable names within curly brackets **{ }** inside quotes.

```
print(F"some text here {variable_name}")

print(F"""This is text as well
Text in second line {variable_name}""")
```

Exercise: 3

Use **F** string to show celebrity name in print statement.



CHEAT SHEET

- Add an **F** before the **print** statement.
- Inside the **print** statement, remove all the extra quotes and comma.
- Add the variable name inside curly brackets.

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

celebrity_name = "Robert Downey, Jr."
print(F"""
Celebrity facts about """, {celebrity_name}, """
- """, {celebrity_name}, """ was born in 1965
- """, {celebrity_name}, """ has done a total of 79 movies.
- """, {celebrity_name}, """'s first movie was Pound.
- Pound has a rating of 6.3
""")
```

SOLUTION

facts.py

```
#!/usr/bin/python3
print("Content-type: text/html \n")
import magicwand

celebrity_name = "Robert Downey, Jr."
print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in 1965
- {celebrity_name} has done a total of 79 movies.
- {celebrity_name}'s first movie was Pound.
- Pound has a rating of 6.3
""")
```

```
...
print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in 1965
- {celebrity_name} has done a total of 79 movies.
- {celebrity_name}'s first movie was Pound.
- Pound has a rating of 6.3
""")
```

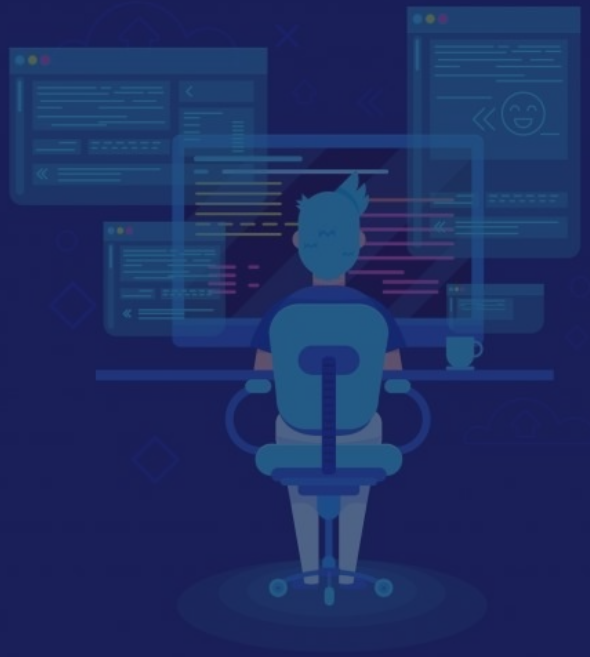
HOORAY!!

We can now change celebrity name at one place, and it will reflect the change everywhere.

Can you find out any other data from this print statement we can store in variables?

Exercise: 4

Use variables for other values like the first movie or rating.



VARIABLE: INTEGER & FLOAT

To store numbers in a variable, write them without using quotes. Just like how we showed them in the print statement.

In python, there's no difference in the syntax for storing integer and float (decimal numbers) data.

```
variable_name = 12000
```

Integer data

```
variable_name = 120.521
```

Float data
(decimal numbers)

CHEAT SHEET

Step 1

- Create variables for
  birth_year
  total_movies
  first_movie
  movie_rating
- Store the values we wrote in print statement inside these variables.

facts.py

```
...
celebrity_name = "Robert Downey, Jr."
variable2 = 1965
variable3 = 79
variable4 = "Pound"
variable5 = 6.3

print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in 1965
- {celebrity_name} has done a total of 79 movies.
...
```

CHEAT SHEET

Step 2

- Use these variables in the print statement instead of values.

facts.py

```
...
celebrity_name = "Robert Downey, Jr."
variable2 = 1965
variable3 = 79
variable4 = "Pound"
variable5 = 6.3

print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in ~~1965~~{variable2}
- {celebrity_name} has done a total of ~~79~~{variable3} movies.
- {celebrity_name}'s first movie was ~~Pound~~{variable4}.
- ~~Pound~~{variable4} has a rating of ~~6.3~~{variable5}
""")
```

SOLUTION

facts.py

```
...
celebrity_name = "Robert Downey, Jr."
birth_year = 1965
total_movies = 79
first_movie = "Pound"
movie_rating = 6.3

print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in {birth_year}
- {celebrity_name} has done a total of{total_movies} movies.
- {celebrity_name}'s first movie was {first_movie}.
- {first_movie} has a rating of {movie_rating}
""")
```

```
print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in {birth_year}
- {celebrity_name} has done a total of{total_movies} movies.
- {celebrity_name}'s first movie was {first_movie}.
- {first_movie} has a rating of {movie_rating}
""")
```

Now the data we are sending to the print statement is made up of smaller data elements:

celebrity_name	Robert Downey, Jr.
birth_year	1965
total_movies	79

first_movie	Pound
movie_rating	6.3

CAN YOU TELL THE TYPE OF EACH DATA WE STORED IN VARIABLES TODAY?

celebrity_name
Robert Downey, Jr.

first_movie
Pound

movie_rating	6.3
birth_year	1965
total_movies	79

CAN YOU TELL THE TYPE OF EACH DATA WE STORED IN VARIABLES TODAY?

STRING

celebrity_name
Robert Downey, Jr.

first_movie
Pound

INTEGER

birth_year	1965

total_movies	79

FLOAT

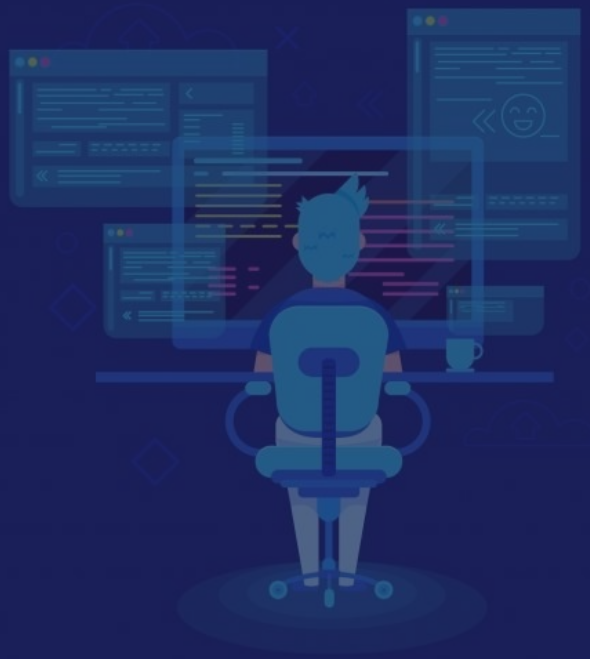
movie_rating	6.3

THAT'S ALL FOR TODAY

- ➔ Discussed the problem of repeating code.
- ➔ Learned about variables and how they help implement the DRY principle.
- ➔ Printed celebrity facts using variables.

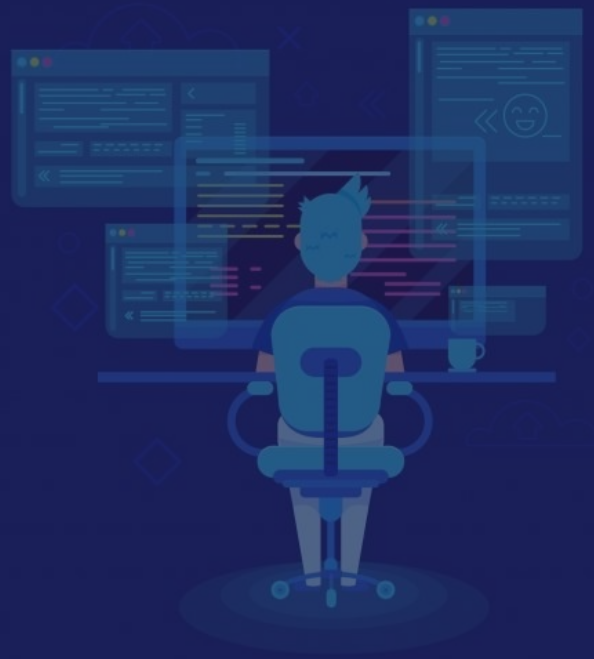
HOMEWORK: 1

Use variables in the previous class project receipt generator.



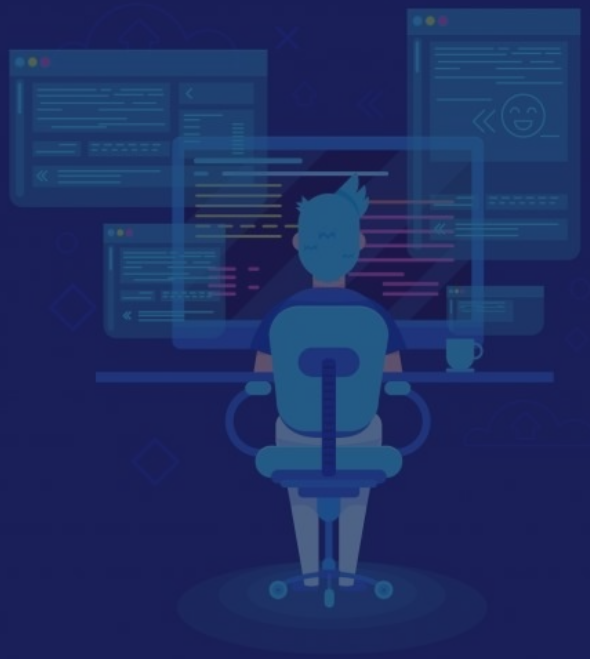
In file **`awesome.py`**

- ❑ Create seven variables to store the name of each item we have.
- ❑ Use these variables in **`print`** statement instead of showing directly.
- ❑ Similarly, use another seven variables for the price of each item.
- ❑ Create another variable total and store sum of all seven prices.
- ❑ Use this total variable to print the total amount in receipt.



HOMEWORK: 2

Print an arithmetic table using variables. _



MULTIPLYING NUMBERS

To multiply a number in python or any other programming language we use the * operator

```
print(12 * 13)
```

Output:
156

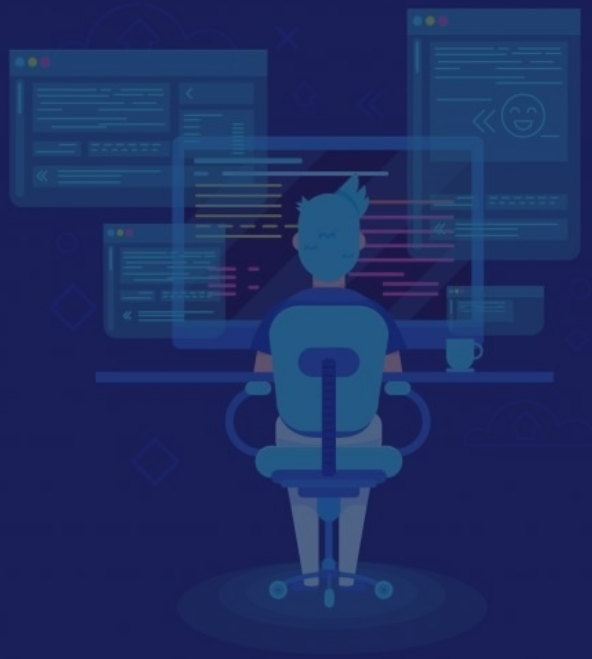
In file **`table.py`**

- ❑ Add the python shebang and the HTML content type.
- ❑ Import magicwand to get clear output.
- ❑ Create a variable and store any number in it.
- ❑ Use a print statement to print the arithmetic table of that number multiplied till 10.
- ❑ Use * to multiply the value in a variable with a number.

```
13 x 1 = 13
13 x 2 = 26
13 x 3 = 39
13 x 4 = 52
13 x 5 = 65
13 x 6 = 78
13 x 7 = 91
13 x 8 = 104
13 x 9 = 117
13 x 10 = 130
```

BONUS Exercise: 1

To perform the arithmetic operation in F Strings.



Output:
The total is 15

THE F STRING: ALGEBRA

We can use arithmetic operators like `+`, `-`, `*`, `/` inside the curly brackets `{ }` with numeric data. It will print the resulting number.

```
variable_name = 10
print(F"The total is {variable_name + 5}")
```

```
print(F"The total is {50 - 5}")
```

Output:
The total is 45

CHEAT SHEET

- Calculate the age of our celebrity by subtracting value in variable **birth_year** from **2019**.

facts.py

```
...
print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in {birth_year}
- {celebrity_name}'s age is {2019 - some_var}
- {celebrity_name} has done a total of{total_movies} movies.
- {celebrity_name}'s first movie was {first_movie}.
- {first_movie} has a rating of {movie_rating}
""")
```

SOLUTION

facts.py

```
...
print(F"""
Celebrity facts about {celebrity_name}
- {celebrity_name} was born in {birth_year}
- {celebrity_name}'s age is {2019 - birth_year}
- {celebrity_name} has done a total of{total_movies} movies.
- {celebrity_name}'s first movie was {first_movie}.
- {first_movie} has a rating of {movie_rating}
""")
```