

st125333_assignment1.1

August 27, 2025

0.0.1 Task 1: Personal Introduction

This task focuses on variable definition, basic data types (string, int), and basic syntax (print()).

Create a variable called `my_name` and assign your name to it as a string. Create a variable called `my_age` and assign your age to it as an integer. Create a variable `is_learning_python` and set it to the boolean value `True`. Write a single `print()` statement that uses an f-string to introduce yourself. Example Output:

Hello, my name is Alex and I am 20 years old. It's True that I am learning Python!

```
[1]: # Task1
my_name="Sat Naing Tun"
my_age="32"
is_learning_python = True
print(f"Hello, my name is {my_name} and I am {my_age} years old. It's_
↳{is_learning_python} that I am learning Python!")
```

Hello, my name is Sat Naing Tun and I am 32 years old. It's True that I am learning Python!

0.0.2 Task 2: My Favorite Hobbies

This task focuses on Lists, the `list()` constructor, and Set operations.

Create a list named `hobbies_list` containing at least five of your hobbies. Make sure to include one hobby twice to create a duplicate. Use the `set()` constructor to create a new set called `unique_hobbies` from `hobbies_list`. Create another set called `friend_hobbies` with at least three hobbies, where one is the same as one of your unique hobbies. Find the hobbies you and your friend have in common using the `intersection()` method. Store the result in a variable called `common_hobbies`. Print your set of unique hobbies and the set of common hobbies. Example Output:

My unique hobbies are: {'cooking', 'gaming', 'reading', 'hiking'} Our common hobbies are: {'reading'}

```
[2]: # Task2 Answer
hobbies_list = ["football", "reading", "gaming", "programming", "reading"]

unique_hobbies = set(hobbies_list)
```

```

friend_hobbies = {"reading", "basketball", "travelling", "gaming"}

common_hobbies = unique_hobbies.intersection(friend_hobbies)

print(f"My unique hobbies are: {unique_hobbies}")
print(f"Our common hobbies are: {common_hobbies}")

```

My unique hobbies are: {'gaming', 'football', 'programming', 'reading'}
Our common hobbies are: {'gaming', 'reading'}

0.0.3 Task 3: The Ultimate Movie Log

This task features more complex List and Dictionary operations.

Part A: Complex List Operations Create a list of strings called `watched_movies` with three movies you have already seen. Create a second list called `watchlist` with two movies you want to see. Create a new list called `full_movie_library` by concatenating `watched_movies` and `watchlist` using the `+` operator. Use the `.pop()` method to remove the first movie from your watchlist and store its title in a new variable called `just_watched`. Print a confirmation message, for example: “I just finished watching [just_watched]!” Use the `.remove()` method to remove one movie from `full_movie_library` by its exact title. Sort the `full_movie_library` alphabetically in place using the `.sort()` method and print the final, sorted library. Example Output for Part A:

I just finished watching The Grand Budapest Hotel! My final movie library: ['Dune', 'Inception', 'Parasite', 'The Matrix']

```

[3]: # Task3 Part A Answer
watched_movies = [ "Beyond Rangoon", "The Boys in The Striped Pajamas", "Tomorrow Never Dies"]
watchlist = ["Broker", "Seven O'clock Breakfast Meeting for the Brokenhearted"]
full_movie_library = watched_movies + watchlist

just_watched = watchlist.pop(0)
print(f"I just finished watching {just_watched}!")

full_movie_library.remove("Broker")
full_movie_library.sort()

print(f"My final movie library: {full_movie_library}")

```

I just finished watching Broker!

My final movie library: ['Beyond Rangoon', 'Seven O'clock Breakfast Meeting for the Brokenhearted', 'The Boys in The Striped Pajamas', 'Tomorrow Never Dies']

Part B: Complex Dictionary Operations Create a dictionary called `movie_ratings` for the movies in your original `watched_movies` list. The keys should be the movie titles (strings) and the

values should be your rating from 1 to 10 (integers). Update the value for one of the movies in `movie_ratings` to a new number. Add the movie you just watched (just_watched from Part A) to the `movie_ratings` dictionary with its rating. Use the `.get()` method to look for a movie title that does not exist in `movie_ratings`. Provide a default value of "Not Rated". Store the result in `queried_rating` and print it. Use the `del` keyword to delete one key-value pair from `movie_ratings`. Print the final `movie_ratings` dictionary. Example Output for Part B:

Rating for 'Avatar': Not Rated Final Movie Ratings: {'Inception': 10, 'The Matrix': 9, 'The Grand Budapest Hotel': 9}

```
[4]: # Task3 B Answer
import random
movie_ratings = {movie: random.randint(5, 10) for movie in watched_movies}

movie_ratings["The Boys in The Striped Pajamas"]=10
movie_ratings[just_watched] = 9

queried_rating = movie_ratings.get("Avatar", "Not Rated")
print(f"Rating for 'Avatar': {queried_rating}")

del movie_ratings["Tomorrow Never Dies"]
print(f"Final Movie Ratings: {movie_ratings}")
```

Rating for 'Avatar': Not Rated

Final Movie Ratings: {'Beyond Rangoon': 5, 'The Boys in The Striped Pajamas': 10, 'Broker': 9}

0.0.4 Task 4: Final Summary

This final task is a quick review of printing different data types.

Print the number of movies in your `full_movie_library` using the `len()` function. Get a list of all the titles from your final `movie_ratings` dictionary using the `.keys()` method. (Hint: You may need to wrap it with the `list()` constructor). Print this list. Print the data type of each of the following variables using the `type()` function: `my_age` `full_movie_library` `queried_rating` `movie_ratings` Example Output:

My final movie library has 4 movies. Movie titles I have rated: ['Inception', 'The Matrix', 'The Grand Budapest Hotel'] Type of `my_age`: <class 'int'> Type of `full_movie_library`: <class 'list'> Type of `queried_rating`: <class 'str'> Type of `movie_ratings`: <class 'dict'>

```
[5]: # Task 4: 1.
print(f"My final movie library has {len(full_movie_library)} movies.")
```

My final movie library has 4 movies.

```
[6]: # Task 4: 2.
rated_movies = list(movie_ratings.keys())
print(f"Movie titles I have rated: {rated_movies}")
```

Movie titles I have rated: ['Beyond Rangoon', 'The Boys in The Striped Pajamas', 'Broker']

```
[7]: # Task 4: 3.  
print(f"Type of my_age: {type(my_age)}")  
print(f"Type of full_movie_library: {type(full_movie_library)}")  
print(f"Type of queried_rating: {type(queried_rating)}")  
print(f"Type of movie_ratings: {type(movie_ratings)}")
```

```
Type of my_age: <class 'str'>  
Type of full_movie_library: <class 'list'>  
Type of queried_rating: <class 'str'>  
Type of movie_ratings: <class 'dict'>
```

```
[ ]:
```