

ASSIGNMENT 10

Python Repetition

Create Python scripts as per the instructions for each exercise. Save these scripts in separate files named `A10E<n>.py`, where `<n>` is the exercise number. Submit all files to Gradescope Assignment 10.

Use script templates from D2L. Follow best practices, including:

- Module structure
- Descriptive **variable** names
- Consistent **variable** naming convention (snake_case, mixedCase, or CamelCase)
- Module docstring for script functionality and usage
- Comments for clarity in complex code sections

Scripts will be auto-graded for functional correctness on Gradescope. Manual grading will assess adherence to best practices.

Scripts must be submitted by 11:59pm two days prior to the next class. You can resubmit to correct errors before the due date.

EXERCISE 1

Create a Python module with three functions to analyze a tuple (or list) of integers:

- `all_divisible()`
 - Parameters: a tuple (or list) of integers, and a divisor
 - Returns True if all integers in the tuple are divisible by the divisor, False otherwise.
- `any_divisible()`
 - Parameters: a tuple (or list) of integers, and a divisor
 - Returns True if any integer in the tuple is divisible by the divisor, False otherwise.
- `difference()`
 - Parameter: a tuple (or list) of integers
 - Returns the result of subtracting all integers in descending order

Assume the tuple always contains at least one integer. Example function calls and return values are provided.

Example:

```
>>> nums = (9,12,6,8)
>>> all_divisible(nums,3)
>>> True
```

```
>>> nums = (9,12,6,8)
>>> any_divisible(nums,4)
>>> True
```

```
>>> nums = (7,13,6,21)
>>> any_divisible(nums,4)
>>> False
```

```
>>> nums = (9,12,6,8)
>>> difference(nums,4) # 12 - 9 - 6 - 8
>>> -11
```

Hints:

- For `all_divisible()`, iterate the tuple and check divisibility. Return False if any integer is not divisible.
- For `any_divisible()`, iterate and check divisibility. Return True if any integer is divisible.
- For `difference()`, sort the tuple in reverse order and subtract integers from the first item.

GRADESCOPE SUBMISSION

A Python script named `A10E1.py`

EXERCISE 2

Modify `remove_str_from_list()` from a previous activity (M9A4) to remove multiple instances of a specified string from a list.

- Returns an integer count of removed instances.
- Add a new optional parameter `max` (default 1) to specify the maximum instances to remove.

Example function calls and return values are provided.

Hints:

- Use a loop iterating `max` times. Return early if no more instances of the string are found.
- Count the number of removed instances and return this count.

GRADESCOPE SUBMISSION

A Python script named `A10E2.py`

EXERCISE 3

Write a Python script for a guessing game:

- The computer selects a random number between 1 and 100.
- The player guesses the number until correct, with hints if too high or low.
- If an invalid guess is made, prompt again with an error message.
- Offer a replay option at the end of each game.

Example script output is provided.

Script Structure:

- `main()`
 - Plays the game until the player opts out.
- `get_player_guess()`
 - Asks for and validates the player's guess.
- `check_player_guess()`
 - Compares the guess to the magic number and prints feedback.

GRADESCOPE SUBMISSION

A Python script named `A10E3.py`