Reading / Resources

- Template literals (Template strings) (MDN)
- Object initializer (MDN)
- Arrow function expressions (MDN)
- Destructuring assignment (MDN)
- Default parameters (MDN)
- Spread syntax (...) (MDN)
- Rest parameters (MDN)
- JavaScript modules (MDN)
- Array (MDN)
- Conditional (ternary) operator (MDN)
- Optional chaining (?.) (MDN)
- Logical AND (&&) (MDN)
- Logical OR (||) (MDN)
- Nullish coalescing operator (??) (MDN)

Homework

1. Write a function called `introducePerson` that takes an object representing a person and returns a string introducing them. The object has properties: `firstName`, `lastName`, and `country`. If the country is missing, it should default to "Unknown".

```
// code sample
const person = { firstName: "John", lastName: "Doe", country: "Greece" };
console.log(introducePerson(person)); // "Hello, I am John Doe from
Greece."

const person2 = { firstName: "Jane", lastName: "Smith" };
console.log(introducePerson(person2)); // "Hello, I am Jane Smith from
Unknown."
```

2. Given the below array of (user) objects, complete the below exercises

```
const users = [
  { name: "Alice", age: 25, isAdmin: true, gender: "female" },
  { name: "Bob", age: 30, isAdmin: false, gender: "male" },
  { name: "Charlie", age: 35, isAdmin: true, gender: "male" },
  { name: "Daisy", age: 28, isAdmin: false, gender: "female" },
  { name: "Edward", age: 40, isAdmin: true, gender: "male" },
];
```

Write the following functions:

1. **filterAdmins**: Returns an array of all users who are admins (`isAdmin: true`).
2. **filterByGender**: Accepts a gender string ("male" or "female") and returns an array of user objects
3. **getUserNames**: Returns an array of names of all users in the original array.
4. **findUserByName**: Accepts a name string and returns the user object with that name (or undefined if not found).
5. **getAverageAge**: Returns the average age of all users.