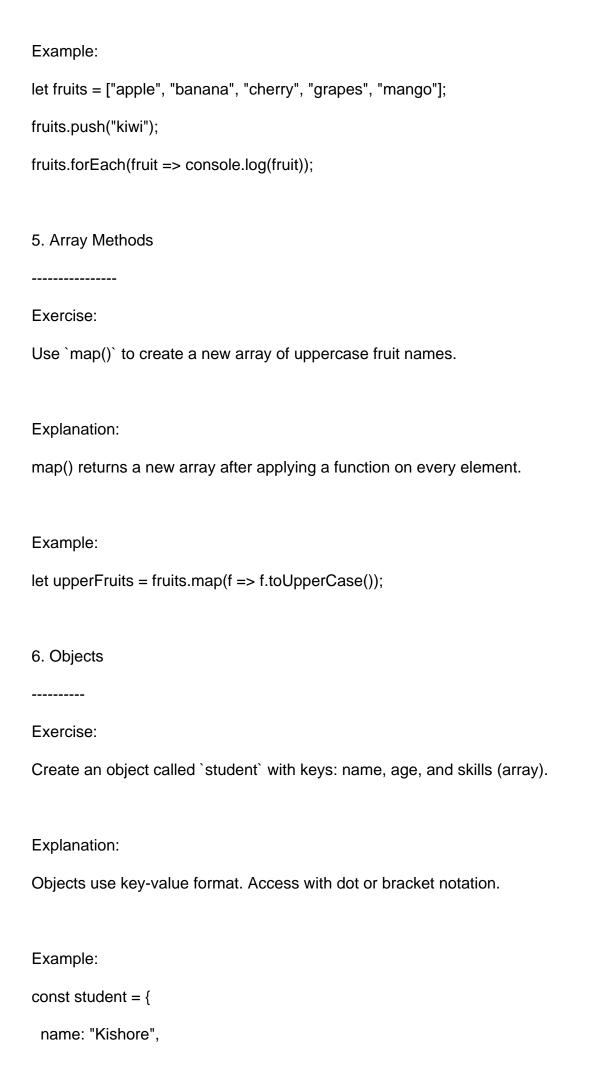
JavaScript Exercises with Explanations
JavaScript Exercises - Day 4
1. Variables
Exercise:
Declare a variable called `username` and assign your name to it.
Try reassigning it with a different name using `let` and then try with `const`.
Explanation:
- `let` allows reassignment.
- `const` throws an error on reassignment.
Example:
let username = "Kishore";
username = "Kumar"; // works
const pi = 3.14;
pi = 3.141; // error
2. Functions
Exercise:
Create a function called `greetUser` that takes a name and returns a greeting message.
Explanation:
Functions allow code reuse and modularity.

```
Example:
function greetUser(name) {
 return "Hello, " + name + "!";
}
console.log(greetUser("Kishore")); // "Hello, Kishore!"
3. Arrow Functions
Exercise:
Convert the `greetUser` function into an arrow function.
Explanation:
Arrow functions provide a shorter syntax.
Example:
const greetUser = (name) => "Hello, " + name;
4. Arrays
Exercise:
Create an array of 5 fruits. Print each fruit using `forEach`. Add one more fruit using `push`.
Explanation:
Arrays hold ordered data. Use methods like push, pop, map, filter etc.
```



```
age: 25,
 skills: ["JS", "HTML", "CSS"]
};
console.log(student.name);
console.log(student.skills[1]);
7. Nested Objects
Exercise:
Create an object `user` with a nested `address` object.
Explanation:
Nested objects allow hierarchical data structures.
Example:
const user = {
 name: "Ravi",
 address: {
  city: "Chennai",
  pin: 600001
 }
};
console.log(user.address.city);
```

8. Looping an Object

Exercise:

Loop through all properties of the `student` object using `for...in`.

Explanation:

Use `for...in` to iterate object properties.

Example:

for (let key in student) {

 console.log(key + ": " + student[key]);
}

--- END ---