◆ Problem: Case Sensitivity in JavaScript

JavaScript is **case-sensitive**, meaning `"Cheyenne"` and `"cheyenne"` are considered **different** strings.

Example:

```
var city1 = "Cheyenne";
var city2 = "cheyenne";

console.log(city1 === city2); // false
```

Even though a human would recognize `"Cheyenne"` and `"cheyenne"` as the same city, JavaScript does **not**.

◆ First (Inefficient) Solution: Duplicating City Names

One way to handle different capitalization styles is to **store multiple versions** of each city name in an array:

```
var cleanestCities = ["Cheyenne", "cheyenne", "Santa Fe", "santa fe", "Tucson", "tucson", "Great Falls", "great falls", 
"Honolulu", "honolulu"];
```

This approach works **but is inefficient** because:

- It **duplicates data**, increasing the size of the array.
- It still **doesn't handle every case**, such as `"santa Fe"` or `"SANTA FE"`.
- It's **not scalable**—as the number of cities increases, this approach becomes impractical.

◆ Best Solution: Convert User Input to Lowercase

Instead of storing multiple versions of each city, we:

1. **Store all city names in lowercase.**
2. **Convert the user's input to lowercase before comparison.**

Code Explanation

```
var cityToCheck = prompt("Enter your city"); // Step 1: Get user input
cityToCheck = cityToCheck.toLowerCase(); // Step 2: Convert input to lowercase

var cleanestCities = ["cheyenne", "santa fe", "tucson", "great falls", "honolulu"]; // Step 3: Store all cities in lowercase

for (var i = 0; i <= 4; i++) { // Step 4: Loop through the array
  if (cityToCheck === cleanestCities[i]) { // Step 5: Compare (case-insensitive)
    alert("It's one of the cleanest cities");
  }
}
```

How It Works

- `.toLowerCase()` ensures that all user input is converted to lowercase.
- We store all city names in **lowercase** in `cleanestCities` to match the input.
- The `for` loop iterates over the array and checks if the input city exists in the list.

◆ Example: Before vs. After Using `.toLowerCase()`

Without `.toLowerCase()`

User Input	Stored Value in Array	Match?
Cheyenne	"cheyenne"	❌ No
sAnta Fe	"santa fe"	❌ No

User Input	Stored Value in Array	Match?
TUCSON	"tucson"	❌ No

With .toLowerCase()

User Input	Converted to	Stored Value	Match?
Cheyenne	cheyenne	"cheyenne"	✅ Yes
sAnta Fe	santa fe	"santa fe"	✅ Yes
TUCSON	tucson	"tucson"	✅ Yes

Now, **any variation of capitalization** works because we compare everything in **lowercase**.

◆ Alternative Approach: Using .toUpperCase()

Instead of converting everything to **lowercase**, we could convert everything to **uppercase**:

```
cityToCheck = cityToCheck.toUpperCase();
var cleanestCities = ["CHEYENNE", "SANTA FE", "TUCSON", "GREAT FALLS", "HONOLULU"];
```

- This works the same way.
- Most coders **prefer lowercase** (toLowerCase()) because lowercase is easier to read and type.

◆ Summary of Key Takeaways

Concept	Explanation
Case Sensitivity	JavaScript treats "Cheyenne" and "cheyenne" as different.
Inefficient Solution	Storing multiple versions ("Cheyenne", "cheyenne", etc.) is **bad practice**.
Best Solution	Convert **user input** and **stored values** to **lowercase** before comparison.
.toLowerCase()	Converts **any string** to all lowercase letters.
Alternative (.toUpperCase())	Converts everything to uppercase instead, but lowercase is preferred.