


Engage Users with the JavaScript prompt Function!


Basic Usage of prompt:	2
Example 1: Collecting User's Name	3
Example 2: Calculating with prompt	3

```
const num1 = parseFloat(prompt("Enter a number:"));
const num2 = parseFloat(prompt("Enter another number:"));


if (!isNaN(num1) && !isNaN(num2)) {
  const sum = num1 + num2;
  alert(`The sum of ${num1} and ${num2} is ${sum}`);
} else {
  alert("Please enter valid numbers.");
}
```

In JavaScript, the prompt function is a handy tool for gathering user input through a simple dialog box. Here's how it works:

 **User Interaction:** It displays a prompt message to the user, asking for input. Users can provide information or make choices.

 **Use Cases:** prompt can be used to collect user names, ask for numbers, or prompt users for basic decisions within your program.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

 Security Note: Be cautious and validate user input to prevent security risks like XSS attacks. Remember, the user can cancel the prompt.

While prompt is suitable for simple interactions, in more complex web applications, HTML forms and event handling are often preferred for a more user-friendly and secure input experience.

#JavaScript #UserInteraction #WebDevelopment #CodingTips #LinkedInLearning

In JavaScript, the prompt function is a built-in method that allows you to interact with the user by displaying a dialog box and prompting them to enter some input. This input is typically in the form of text, and it can be used for various purposes such as collecting user information, receiving input for calculations, or making simple choices in your program. Let's explore how the prompt function works with some coding examples.

Basic Usage of prompt:

Here's the basic syntax for using the prompt function:

```
const userInput = prompt("Please enter some information:");
```

- The argument you pass to prompt is the message or prompt that will be displayed to the user in the dialog box.
- When the prompt dialog appears, the user can enter text in an input field and click "OK" to submit their input or "Cancel" to close the dialog without providing any input.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- The text entered by the user is stored in the userInput variable in this example.

Example 1: Collecting User's Name

```
const userName = prompt("What's your name?");
if (userName) {
  alert(`Hello, ${userName}!`);
} else {
  alert("You didn't provide a name.");
}
```

In this example, we use prompt to collect the user's name and store it in the userName variable. We then use an if statement to check if the user provided a name (i.e., the input is not empty), and if so, we display a greeting message with their name; otherwise, we notify them that they didn't provide a name.

Example 2: Calculating with prompt

```
const num1 = parseFloat(prompt("Enter a number:"));
const num2 = parseFloat(prompt("Enter another
number:"));
```

```
if (!isNaN(num1) && !isNaN(num2)) {  
    const sum = num1 + num2;  
    alert(`The sum of ${num1} and ${num2} is ${sum}`);  
} else {  
    alert("Please enter valid numbers.");  
}
```

In this example, we use `prompt` to collect two numbers from the user, and we convert them to floating-point numbers using `parseFloat`. We check if both inputs are valid numbers (not NaN), and if so, we calculate and display their sum.

Caution:

- Remember that the user can cancel the prompt, which will result in null or an empty string as the input value.
- Ensure you validate and sanitize user input when using `prompt` to prevent security vulnerabilities like cross-site scripting (XSS) attacks.
- Modern web applications often use HTML forms and event handlers instead of `prompt` for a more user-friendly and secure user input experience.

The `prompt` function is a simple way to interact with users in a basic JavaScript program, but it's not suitable for complex or production-level applications. In real-world scenarios, consider using HTML forms and event handling for user input.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>