

Lesson

Monday

Introduction to Programming

(/introduction-to-programming)

/ JavaScript and Web Browsers

(/introduction-to-programming/javascript-and-web-browsers)

/ Interactivity with Window Methods

Text

Cheat sheet

So far, none of what we've learned lets us interact with users. Let's try out `window` methods that let us do just that!

Interactivity with `window` Methods

We'll continue to use our DevTools console to practice the following `window` methods.

`window.alert()`

Try running the following code in the DevTools console:

```
> window.alert("Hello world!");
```

When we press enter, the page pops open a dialog box that says "Hello world!"

We should be able to tell that `window.alert()` is a **method**. A method is a function that belongs to an object, and a function is something that performs an action. As we know, methods can take arguments. The `window.alert()` method pops up a dialog box with the string that you pass in as an argument.

`window.prompt()`

Here's another `window` method that takes a string as an argument:

```
> window.prompt("What is the air-speed velocity of an unladen swallow?"); // I'm going to type "African or European?"  
"African or European?"
```

Remember, in JavaScript everything after the double forward slash `//` is a **comment**. When JavaScript interprets our code, it will ignore these comments. Comments are a convenient way to leave notes in your code for yourself or other programmers. Note, by the way, that the comment comes *after* the semicolon.

This dialog box lets you type in a response, and then that response is **returned** from the `window.prompt()` method as a string. One cool thing we can do here is set a variable equal to the response, like this:

```
> const favoriteColor = prompt("What is your favorite color?"); // I'm going to type "green"  
> favoriteColor;  
"green"
```

Here we created a variable named `favoriteColor` and assigned it to hold the string inputted by the user.

Just like we can chain methods to each other in JavaScript, we can also chain JavaScript methods to `window` methods:

```
> window.prompt("Type something in lowercase:").toUpperCase()  
(); // I'm going to type "cat"  
"CAT"
```

Remember that browser Web APIs like the `window` object are created to work with JavaScript, so we can perform any JavaScript on these methods. Because `window.prompt()` returns a string, we can chain string methods to `window.prompt()` method calls.

`window.confirm()`

Another `window` method that allows us to interact with the webpage user is the `window.confirm()` method. This too takes a string as an argument.

```
> confirm("Are you sure?"); // I'm going to press the OK bu  
tton  
true
```

Try it out for yourself in the DevTools console. You should note that `window.confirm()` returns one of two values: `true` or `false`. Notice that there are no quotes around these values. That is because these are **booleans**, not strings. They simply represent being true or false. You initially learned about booleans when you worked with comparison and equality operators.

Pay Attention to the Data Type that Methods Return

You might have noticed that `window.alert()` returned `undefined`, also without quotes.

```
> window.alert("Hello world!");  
undefined
```

`undefined` simply represents that nothing has been returned from the method (or function), or as you learned previously, that a variable hasn't been assigned a value.

It is good to be conscious of what arguments a method or function takes, and what data type it returns. Here is a table showing that information for the three `window` methods introduced in this lesson:

WINDOW METHOD	ARGUMENT	RETURN VALUE
<code>window.alert()</code>	string	undefined
<code>window.prompt()</code>	string	string
<code>window.confirm()</code>	string	boolean

The `window.alert()` method does not return anything useful, which is fine because we'd only use it to display some information to the user. It's a way to output information, not collect it.

On the other hand, we'd want to collect the user's response to the `window.prompt()` or `window.confirm()` methods by assigning the returned value to a variable as demonstrated above:

```
> const favoriteColor = window.prompt("What is your favorite  
color?");
```

That way we can do something with the inputted information rather than just ignoring it.

[Previous \(/introduction-to-programming/javascript-and-web-browsers/debugging-in-javascript-reading-console-errors\)](#)

Next (/introduction-to-programming/javascript-and-web-browsers/a-common-data-type-error-and-data-type-coercion)

Lesson 33 of 75

Last updated more than 3 months ago.

disable dark mode



© 2023 Epicodus (http://www.epicodus.com/), Inc.