

## Writing Your First JavaScript Code in the Browser Console

### Writing Your First JavaScript Code

We will now write our very first line of JavaScript code. For now, we will do this in the browser developer tools to get started as quickly as possible. Later, we will switch to the code editor that was installed previously.

Make sure to open up Google Chrome. There are three ways to open the Chrome developer tools:

- Hit command + option + J on the keyboard (on Mac).
- On Windows, use ctrl + alt + J.
- Right-click and select Inspect, then go to the Elements tab and from there to the Console.
- Alternatively, use the Chrome menu: go to View > Developer > JavaScript Console.

All of these tabs are part of the developer tools, but we are interested in the console. The JavaScript developer console allows us to write and test JavaScript code. It is very useful during development, for example, to fix errors. However, we do not write real applications using this console. For now, to get started, let us use the console because it is an easy way to write some JavaScript.

### Using the Alert Function

Let us write **alert**, which is a JavaScript function. Open parenthesis, and without a space between **alert** and the parenthesis, write a string, which is basically text. Let us write "Hello world." This is the standard way of learning programming in any language. Then, write the closing parenthesis and hit return or enter.

#### javascript Code Sample

```
alert("Hello world.")
```

You will see that JavaScript gives a popup window with "Hello world" just as typed in the string. This is your very first line of JavaScript code. Any code written here and then executed will immediately be evaluated.

### Writing Variables and Conditionals

We can do much more. Let us write some more JavaScript code. You do not need to understand everything yet; this is just to show what can be done. We will go into how everything works as we progress.

#### javascript Code Sample

```
let JS = "amazing";
```

Now, we have defined **JS** as being "amazing". Next, we will use a conditional statement. If **JS** is equal to "amazing", we want an alert window to say "JavaScript is fun."

#### javascript Code Sample

```
if (JS === "amazing") {  
    alert("JavaScript is fun.");  
}
```

If you hit return, you will get a window that says "JavaScript is fun." This demonstrates the logic: we defined **JS** as "amazing", and then checked if it is equal to "amazing". If so, JavaScript shows the alert window with the specified text.

Now, we can also change **JS** to "boring" and run the conditional again.

### **javascript Code Sample**

```
JS = "boring";
```

### **javascript Code Sample**

```
if (JS === "amazing") {  
    alert("JavaScript is fun.");  
}
```

This time, nothing happens because **JS** is no longer "amazing"; it is "boring". Therefore, the condition is not true and the alert window is not shown.

### **Navigating Previous Commands**

There is another way to access previous commands: by hitting the up arrow on the keyboard. This allows you to see all the previous commands that were used.

### **Simple Math Operations in the Console**

Another feature is performing simple math operations. For example, entering **40 + 8 + 23 - 10** and hitting return will display the result. The console can be used as a calculator.

### **javascript Code Sample**

```
40 + 8 + 23 - 10
```

This demonstrates how useful the console can be for quick calculations and experimentation.

### **Conclusion**

This introduction provides a first taste of the JavaScript language and what lies ahead. You are encouraged to experiment further with the console or proceed to the next lecture to learn more about what JavaScript actually is.

### **Key Takeaways**

- Introduced writing JavaScript code in the browser developer tools console.
- Demonstrated how to use the alert function and create variables in JavaScript.
- Showed how to use conditional statements and perform simple math operations in the console.
- Explained how to navigate previous commands and encouraged experimentation.