

4. JavaScript Statements

JavaScript is a sequence of statements to be executed by the browser.

4.1. JavaScript Statements

JavaScript statements are "commands" to the browser.

The purpose of the statements is to tell the browser what to do.

This JavaScript statement tells the browser to write "Hello Dolly" inside an HTML element with id="demo":

```
document.getElementById("demo").innerHTML="Hello Dolly";
```

4.2. Semicolon ;

Semicolon separates JavaScript statements.

Normally you add a semicolon at the end of each executable statement.

Using semicolons also makes it possible to write many statements on one line.



You might see examples without semicolons.

Ending statements with semicolon is optional in JavaScript.

4.3. JavaScript Code

JavaScript code (or just JavaScript) is a sequence of JavaScript statements.

Each statement is executed by the browser in the sequence they are written.

This example will manipulate two HTML elements:

Example

```
document.getElementById("demo").innerHTML="Hello Dolly";  
document.getElementById("myDIV").innerHTML="How are you?";
```

4.4. JavaScript Code Blocks

JavaScript statements can be grouped together in blocks.

Blocks start with a left curly bracket, and end with a right curly bracket.

The purpose of a block is to make the sequence of statements execute together.

A good example of statements grouped together in blocks, are JavaScript **functions**.

This example will run a function that will manipulate two HTML elements:

Example

```
function myFunction()  
{  
  document.getElementById("demo").innerHTML="Hello Dolly";  
  document.getElementById("myDIV").innerHTML="How are you?";  
}
```

You will learn more about functions in later chapters.

4.5. JavaScript is Case Sensitive

JavaScript is case sensitive.

Watch your capitalization closely when you write JavaScript statements:

A function `getElementById` is not the same as `getElementbyID`.

A variable named `myVariable` is not the same as `MyVariable`.

4.6. White Space

JavaScript ignores extra spaces. You can add white space to your script to make it more readable. The following lines are equivalent:

```
var person="Hege";  
var person = "Hege";
```

4.7. Break up a Code Line

You can break up a code line **within a text string** with a backslash. The example below will be displayed properly:

```
document.write("Hello \  
World!");
```

However, you cannot break up a code line like this:

```
document.write \  
("Hello World!");
```