Code structure

by Afaf KELAI

Statements

Statements are syntax constructs and commands that perform actions.

We've already seen a statement, `alert('Hello, world!')`, which shows the message "Hello, world!".

We can have as many statements in our code as we want. Statements can be separated with a semicolon.

For example, here we split "Hello World" into two alerts:

```
alert('Hello'); alert('World');
```

Usually, statements are written on separate lines to make the code more readable:

```
alert('Hello');
alert('World');
```

Semicolons

A semicolon may be omitted in most cases when a line break exists.

This would also work:

```
alert('Hello')
alert('World')
```

Here, JavaScript interprets the line break as an "implicit" semicolon. This is called an automatic semicolon insertion.

In most cases, a newline implies a semicolon. But "in most cases" does not mean "always"!

There are cases when a newline does not mean a semicolon. For example:

```
alert(3 +
1
+ 2);
```

The code outputs `6` because JavaScript does not insert semicolons here. It is intuitively obvious that if the line ends with a plus `"+"`, then it is an "incomplete expression”, so the semicolon is not required. And in this case that works as intended.

Comments

As time goes on, programs become more and more complex. It becomes necessary to add *comments* which describe what the code does and why.

Comments can be put into any place of a script. They don’t affect its execution because the engine simply ignores them.

One-line comments start with two forward slash characters `//`.

The rest of the line is a comment. It may occupy a full line of its own or follow a statement.

Like here:

```
// This comment occupies a line of its own
alert('Hello');

alert('World'); // This comment follows the statement
```

Multiline comments start with a forward slash and an asterisk `/*` and end with an asterisk and a forward slash `*/`.

Like this:

```
/* An example with two messages.
This is a multiline comment.
*/
alert('Hello');
alert('World');
```

The content of comments is ignored, so if we put code inside `/* … */`, it won’t execute.

Sometimes it can be handy to temporarily disable a part of code:

```
/* Commenting out the code
alert('Hello');
*/
alert('World');
```

Use hotkeys!

In most editors, a line of code can be commented out by pressing the Ctrl+/ hotkey for a single-line comment and something like Ctrl+Shift+/ – for multiline comments (select a piece of code and press the hotkey). For Mac, try Cmd instead of Ctrl and Option instead of Shift.

Nested comments are not supported!

There may not be `/*...*/` inside another `/*...*/`.

Such code will die with an error:

```
/*
  /* nested comment ?!? */
*/
alert( 'World' );
```