

JavaScript Syntax

Summary: in this lesson, you will learn about JavaScript syntax, including whitespace, statements, identifiers, comments, expressions, and keywords.

Whitespace

Whitespace refers to characters that provide the space between other characters. JavaScript has the following whitespace:

- Carriage return
- Space
- New Line
- tab

JavaScript engine ignores whitespace. However, you can use whitespace to format the code to make it easy to read and maintain.

The following JavaScript code doesn't use whitespace:

```
let formatted = true; if (formatted)
{console.log('The code is easy to read');}
```

It's is equivalent to the following code that uses whitespace. Hence, this code is much easy to read:

```
let formatted = true;

if (formatted) {
  console.log('The code is easy to read');
}
```

Note that JavaScript bundlers remove all whitespace from JavaScript files and put them into a single file for deployment. By doing this, JavaScript bundlers make the JavaScript code lighter and faster to load in the web browsers.

Statements

A statement is a code that declares a variable or instructs the JavaScript engine to do a task. A simple statement is terminated by a semicolon (;).

Although the semicolon (;) is optional; you should always use it to terminate a statement. For example, the following [declares a variable](#) and shows it to the console:

```
let message = "Welcome to JavaScript";  
console.log(message);
```

Blocks

A block is a sequence of zero or more simple statements. A block is delimited by a pair of curly brackets {}. For example:

```
if (window.localStorage) {  
    console.log('The local storage is supported');  
}
```

Identifiers

An identifier is a name you choose for variables, parameters, [functions](#), classes, etc. An identifier name starts with a letter (a-z, or A-Z), an underscore(_), or a dollar sign (\$) and is followed by a sequence of characters including (a-z, A-Z), numbers (0-9), underscores (_), and dollar signs (\$).

Note that the letter is not limited to the ASCII character and may include extended ASCII or Unicode though not recommended.

Identifiers are case-sensitive. For example, the message is different from the Message.

Comments

Comments allow you to add notes or hints to JavaScript code. When executing the code, the JavaScript engine ignores the comments.

JavaScript supports single-line and block comments.

Single-line comments

A single-line comment starts with two forward-slashes characters (`//`). A single-line comment makes all the text following the `//` on the same line into a comment. For example:

```
// this is a single-line comment
```

Block comments

A delimited comment begins with a forward slash and asterisk `/*` and ends with the opposite `*/` as in the following example:

```
/* This is a block comment  
that can span multiple lines */
```

Code language: JSON / JSON with Comments (json)

Expressions

An expression is a piece of code that evaluates to a value. For example:

```
2 + 1
```

The above expression returns three.

Keywords & Reserved words

JavaScript defines a list of reserved keywords that have specific uses. Therefore, you cannot use the reserved keywords as identifiers or property names by rules.

The following table shows the JavaScript reserved words defined in ECMA-262:

<code>break</code>	<code>case</code>	<code>catch</code>
<code>continue</code>	<code>debugger</code>	<code>default</code>

<code>else</code>	<code>export</code>	<code>extends</code>
<code>function</code>	<code>if</code>	<code>import</code>
<code>new</code>	<code>return</code>	<code>super</code>
<code>throw</code>	<code>try</code>	<code>null</code>
<code>void</code>	<code>while</code>	<code>with</code>
<code>class</code>	<code>delete</code>	<code>finally</code>
<code>in</code>	<code>switch</code>	<code>typeof</code>
<code>yield</code>	<code>const</code>	<code>do</code>
<code>for</code>	<code>instanceof</code>	<code>this</code>
<code>var</code>		

In addition to the reserved keywords, ECMA-252 also define a list of future reserved words that cannot be used as identifiers or property names:

<code>enum</code>	<code>implements</code>	<code>let</code>
<code>protected</code>	<code>private</code>	<code>public</code>
<code>await</code>	<code>interface</code>	<code>package</code>
<code>implements</code>	<code>public</code>	

Summary

- Use whitespace including carriage return, space, newline, and tab to format the code. The JavaScript engine ignores the whiespace.

- Use a semicolon (;) to terminate a simple statement.
- Use the curly braces ({ }) to form a block that groups one or more simple statements.
- A single-line comment starts with // followed by a text. A block comment begins with /* and ends with */. JavaScript engine also ignores the comments.
- Identifiers are names that you choose for variables, functions, classes, etc.
- Do not use the reserved keywords and reserved words for identifiers.