6.1 Syntax and variables

Background

In 1995, Brendan Eich created JavaScript so the Netscape Navigator browser could dynamically respond to user events. Ex: The web page's content could change when the user clicked a button or hovered over an image.

JavaScript was standardized by Ecma International in 1997 and called ***ECMAScript***. Ecma International continues to improve ECMAScript, releasing a new version each year. JavaScript is an implementation of the ECMAScript specification.

Today, JavaScript is one of the most popular programming languages. JavaScript is supported by every major web browser and makes web applications like Gmail and Google Maps possible. JavaScript is also popular outside the web browser. Ex: Node.js, which runs JavaScript, is a popular technology for creating server-side web applications.

JavaScript is executed by an interpreter. An ***interpreter*** executes programming statements without first compiling the statements into machine language. Modern JavaScript interpreters (also called ***JavaScript engines***) use ***just-in-time (JIT) compilation*** to compile the JavaScript code at execution time into another format that can be executed quickly.

ECMAScript name

The name "ECMAScript" was a compromise between Netscape, Microsoft, and other organizations involved in the standardization of JavaScript. Brendan Eich once commented that "ECMAScript was always an unwanted trade name that sounds like a skin disease." Despite ECMAScript's similarity to eczema (a group of related skin diseases), the name has stuck.

Quote source: Archived email (Oct 3, 2006)

PARTICIPATION ACTIVITY 6.1.1: JavaScript background.

1) ECMAScript and JavaScript are the same thing.
   - ○ True
   - ◉ False

Correct

ECMAScript is the standard upon which JavaScript is based. The standard is called ECMAScript, but the implementation is called JavaScript. A web browser's JavaScript interpreter must be continually updated to support newer ECMAScript versions.

2) JavaScript is only used for programs that run in a web browser.

- ○ True
- ◉ False

Correct

JavaScript can also run on the web server. Ex: Node.js runs JavaScript programs on a web server.

3) All browsers must use the same JavaScript engine.

- ○ True
- ◉ False

Correct

Browsers are free to implement any JavaScript engine. Ex: Chrome uses the V8 JavaScript engine, and Firefox uses SpiderMonkey. Both engines are continually improved to execute JavaScript code more quickly.

4) JavaScript and Java are the same programming language.

- ○ True
- ◉ False

Correct

JavaScript shares some syntactical similarities to Java, but the two languages are not related.

Feedback?

Variables

A ***variable*** is a named container that stores a value in memory. A ***variable declaration*** is a statement that declares a new variable with the keyword ***let*** followed by the variable name. Ex: `let score` declares a variable named score.

A variable may be assigned a value after being declared. An ***assignment*** assigns a variable with a value, like `score = 2`. A variable may also be assigned a value on the same line when the variable is declared, which is called ***initializing*** the variable. Ex: `let maxValue = 5;` initializes `maxValue` to 5.

A variable may be assigned a value without first declaring the variable, but good practice is to always declare a variable before assigning a value to the variable.

PARTICIPATION ACTIVITY 6.1.2: Declaring variables and assigning values.

Start 2x speed

```
// Declaring a variable
let numSongs;

// Variable is assigned a number
numSongs = 5;

// Variable is declared and assigned a number (initialized)
let numAlbums = 20;
```

numSongs
numAlbums
hitCount

```
// Variable may be assigned a value without first being declared
hitCount = 10;
```

Captions

1. The numSongs variable is declared with the "let" keyword.
2. numSongs is assigned with 5.
3. numAlbums is initialized with 20.
4. When hitCount is assigned with 10, hitCount is implicitly declared. Good practice is to explicitly declare all variables with "let".

Feedback?

A name created for an item like a variable is called an ***identifier***. JavaScript imposes the following rules for identifiers:

- An identifier can be any combination of letters, digits, underscores, or $.
- An identifier may not start with a digit.
- An identifier may not be a reserved word like `let`, `function`, or `while`.

A JavaScript coding convention is to name JavaScript variables with camel casing, where the identifier starts with a lowercase letter, and subsequent words begin with a capital letter. Ex: `lastPrice` is preferred over `LastPrice` or `last_price`.

A ***constant*** is an initialized variable whose value cannot change. A JavaScript constant is declared with the ***const*** keyword. Ex: `const slicesPerPizza = 8;` creates a constant `slicesPerPizza` that is always 8.

var keyword

A variable may also be declared with the ***`var`*** *keyword, which is covered elsewhere in this material.*

PARTICIPATION ACTIVITY 6.1.3: Declaring and naming variables.

1) Which statement declares the variable `sum` without assigning a value to `sum`?

- `sum;`
- `let sum;` ✓
- `sum = 0;`

Correct

The `let` keyword declares a variable without assigning a value.

2) Which identifier is illegally named?

- `star_destroyer`
- `ADDRESS`
- `$save`
- `9to5` ✓

Correct

Although a JavaScript identifier may contain numbers, an identifier name may not begin with a number.

3) Which variable is named with the preferred JavaScript naming conventions?

- `total_points`
- `$totalPoints`
- `totalPoints` ✓

Correct

JavaScript coding conventions use camel case for variable names.

4) Which statement declares a constant for Earth's gravity?

- `let earthGravity = 9.8;`
- `const earthGravity = 9.8;` ✓
- `const earthGravity;`

Correct

The `const` keyword declares a constant variable that is initialized to 9.8.

5) Which code segment contains an error?

```
let numLives = 9;
numLives = 8;
```

```
const numLives = 9;
let livesLeft = numLives;
```

```
numLives = 9;
let numLives;
```

Correct

The `numLives` variable cannot be assigned a value before the variable is declared with `let`.

Feedback?

Data types

Variables are not explicitly assigned a data type. JavaScript uses ***dynamic typing***, which determines a variable's type at run-time. A variable can be assigned a value of one type and re-assigned a value of another type. Ex: `x = 5; x = "test";` assigns `x` with a number type, then a string type.

Table 6.1.1: Example JavaScript data types.

Data type	Description	Example
string	Group of characters delimited with 'single' or "double" quotes	`let name = "Naya";` `let quote = 'He asked, "Shall we play a game?"';`
number	Numbers with or without decimal places	`let highScore = 950;` `let pi = 3.14;`
boolean	true or false	`let hungry = true;` `let thirsty = false;`
array	List of items	`let teams = ["Broncos", "Cowboys", "49ers"];`
object	Collection of property and value pairs	`let movie = { title:"Sing", rating:"PG" };`

Data type	Description	Example
undefined	Variable that has not been assigned a value	`let message;`
null	Intentionally absent of any object value	`let book = null;`

Feedback?

PARTICIPATION ACTIVITY 6.1.4: Variable data types.

1) What is the data type of `population`?

```
let population = 650000;
```

- int
- string
- number

Correct

`population` is assigned 650 thousand, which is represented with the number data type. Numbers can range from roughly -9 quadrillion to 9 quadrillion. Larger numbers can be represented with the BigInt datatype by appending "n" to the number. Ex: `let num = 12345678901234567890n;`

2) What is the data type of `z`?

```
let z;
```

- undefined
- string
- number

Correct

A declared variable that is not assigned a value has an undefined data type.

3) What is the data type of `x`?

```
y = false;
x = y;
```

- string
- boolean
- number

Correct

`x` is assigned the value of the boolean variable `y`, so `x` is also boolean.

4) What is syntactically wrong with the following code?

Correct

If single quotes are used as string delimiters, then single quotation marks inside the string must be escaped with a backslash character. Ex: `'Danny O\'Sullivan'`.

```
name = 'Danny O'Sullivan';
```

Alternatively, double quotes can be used as string delimiters. Ex: `"Danny O'Sullivan"`.

- `name` is assigned a value without being declared first.
- Variables may not be assigned strings delimited with single quotes.
- The single quotation mark in O'Sullivan is an error.

Feedback?

Comments and semicolons

A ***comment*** is any text intended for humans that is ignored by the JavaScript interpreter. JavaScript uses the `//` and `/* */` operators to produce comments in code.

Figure 6.1.1: Comments.

```
// Single line
comment

/* Multi-line
   comment
*/
```

Feedback?

JavaScript does not require that statements be terminated with a semicolon. Only when two statements reside on the same line must a semicolon separate the two statements. *Good practice is to avoid placing two statements on the same line.* Some developers prefer to use semicolons at the end of statements, and others do not. *Good practice is to consistently use semicolons or not throughout the code.*

Figure 6.1.2: Using semicolons.

```
let totalPoints = 10;

// No semicolon is required
let totalLives = 3

// Two statements on the same line require a
semicolon
totalPoints = 5; totalLives = 2
```

Feedback?

PARTICIPATION ACTIVITY 6.1.5: Detect the error.

Indicate if the statements contain an error or not.

1)
```
/* a is assigned
2
a = 2;
```

- (•) Error
- ( ) No error

Correct

`/*` begins a comment, but `*/`, which ends the comment, is missing.

2)
```
3.12 = pi;
```

- (•) Error
- ( ) No error

Correct

A number may not be assigned a variable. `pi = 3.12;` is a proper assignment statement.

3)
```
x = 10; let y =
20;
```

- ( ) Error
- (•) No error

Correct

Multiple statements may exist on the same line if separated by semicolons, but good practice is to place two statements on separate lines.

Feedback?

Input and output

A JavaScript program may obtain text input from the user with the `prompt()` function. The ***prompt()*** function prompts the user with a dialog box that allows the user to type a single line of text and press OK or Cancel. The `prompt()` function returns the string the user typed or `null` if the user pressed Cancel.

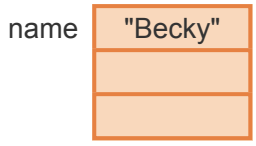
Output may be produced using the function ***console.log()***, which displays text or numbers in the console. The ***console*** is a location where text output is displayed. Web browsers have a console (accessible from the browser's development tools) that displays output from code the browser executes. This chapter's activities display the console output in the web page.

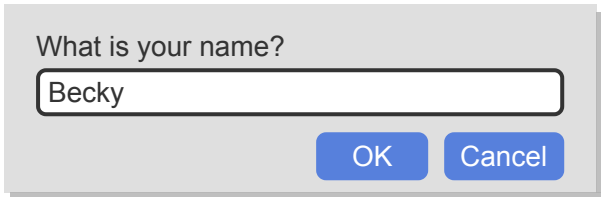
PARTICIPATION ACTIVITY 6.1.6: Prompting for input and displaying output.

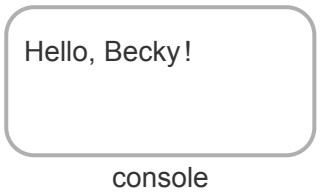
Start 2x speed

```
// Display the prompt dialog box
let name = prompt("What is your name?");

// Output to the console
console.log("Hello, " + name + "!");
```







Captions

1. The prompt() function displays a dialog box with the given prompt text.
2. The user types her name and presses the OK button.
3. The name variable is assigned with the entered text.
4. console.log() outputs "Hello, ", then the value of the name variable, then "!" to the console.

Feedback?

PARTICIPATION ACTIVITY 6.1.7: prompt() and console.log().

1) Write the code to prompt the user with the `question` variable and retrieve the user's age.

```
question = "How old are you?";
age =
prompt(question)
;
```

Check Show answer

Correct

prompt(question)

`prompt()` displays the question in a dialog box.

2) Write the code to display "You are X", where X is the value of the `age` variable.

```
age = 21;
console.log(
"You are " +age
);
```

Check Show answer

Correct

"You are " + age

The + operator appends age's value to the string "You are ", forming a single string.

Feedback?

PARTICIPATION ACTIVITY 6.1.8: JavaScript practice.

The JavaScript code below initializes the variable `tvShow` to a popular TV show. Then, an if-else statement displays a message in the console if `tvShow` is `null`, otherwise the value of `tvShow` is displayed in the console. Change the code to prompt the user for the user's favorite TV show. Then, display "____ is your favorite TV show!" in the console. Press "Run JavaScript" to run your code.

Note: The console will display an error message if the JavaScript interpreter detects a syntax error. A ***syntax error*** is the incorrect typing of a programming statement. Ex: Forgetting to place "quotes" around a string value is a syntax error.

```
1  // Call prompt() to get the user's favorite TV show
2  let tvShow = "The Office";
3
4  if (tvShow === null) {
5     console.log("You did not enter a TV show.");
6  }
7  else {
8     console.log(tvShow);
9  }
10
```

Run JavaScript Reset code

Your console output

```
The Office
```

▼ View solution

Explain

```
--- START FILE: JavaScript ---


let tvShow = prompt();

if (tvShow === null) {
   console.log("You did not enter a TV show.");
}
else {
   console.log(tvShow + " is your favorite TV show!");
}


--- END FILE: JavaScript ---
```

Feedback?

CHALLENGE ACTIVITY 6.1.1: prompt() and console.log().

530096.4000608.qx3zqy7

Start

Write a statement that displays: I love this job

```
1
2 /* Your solution goes here */
3
```

1 2 3

Check Next

View your last submission

Feedback?

Exploring further:

- A Brief History of JavaScript from auth0
- JavaScript Lexical Grammar from MDN

How was this section? Provide section feedback