PRIMATHON
ACADEMY

Part –2

Variables, Identifiers, and Statements

Prerequisite

- If you use Javascript on the Web, you should be familiar with HTML and CSS.
- Familiar with what JavaScript is.

Objective

- In this, we'll discuss Variables, Identifiers, and Statements.

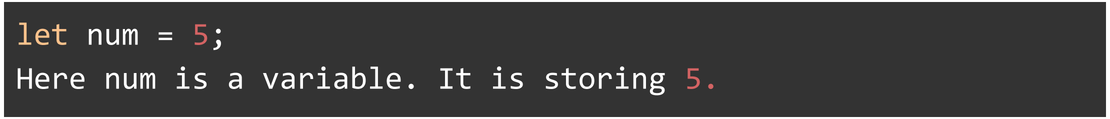
Keywords

- Variables
- Identifiers
- Statements

Introduction – Variables

Variables:

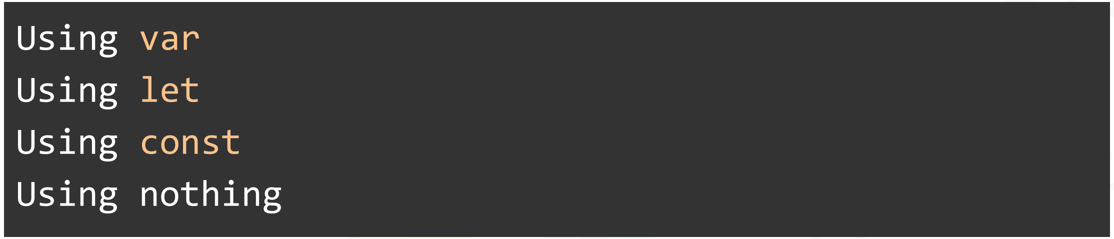
- In programming, a variable is a container (storage area) to hold data.
- For example,



```
let num = 5;
Here num is a variable. It is storing 5.
```

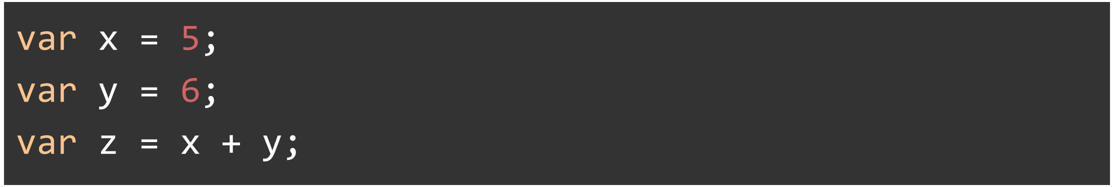
Declare Variables:

- 4 Ways to Declare a JavaScript Variable:



```
Using var
Using let
Using const
Using nothing
```

- In this example, x, y, and z are variables declared with the var keyword:



```
var x = 5;
var y = 6;
var z = x + y;
```

- In this example, x, y, and z are variables declared with the let keyword:

```
let x = 5;
let y = 6;
let z = x + y;
```

- In this example, x, y, and z are undeclared variables:

```
x = 5;
y = 6;
z = x + y;
```

- From all the examples above, you can guess:

```
x stores the value 5
y stores the value 6
z stores the value 11
```

- **When to Use JavaScript var?**

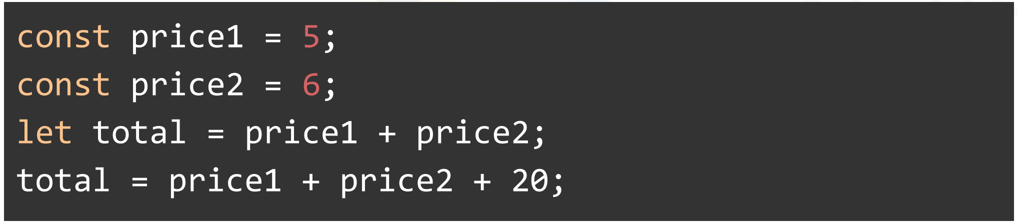
  ○ Always declare JavaScript variables with var, let or const.

  ○ The var keyword is used in all JavaScript code from 1995 to 2015.

  ○ The let and const keywords were added to JavaScript in 2015.

When to Use JavaScript const?

- If you want a general rule, always declare variables with const.
- If you think the value of the variable can change, use let.
- In this example, price1, price2, and total, are variables:

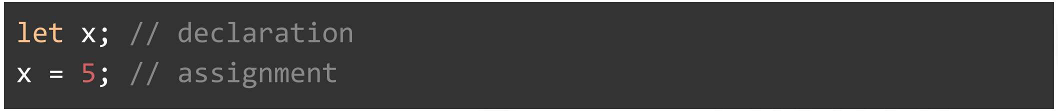


```
const price1 = 5;
const price2 = 6;
let total = price1 + price2;
total = price1 + price2 + 20;
```

- The two variables, price1 and price2, are declared with the const keyword.
- Variables price1 and price2 are constant values and cannot be changed.
- The variable total is declared with the let keyword.
- This is a value that can be changed.

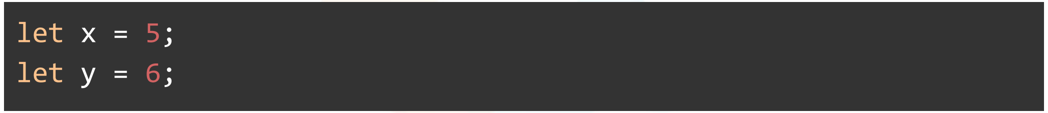
Initialise Variables:

- We use the assignment operator = to assign a value to a variable.



```
let x; // declaration
x = 5; // assignment
```

- We can also initialise variables during its declaration.



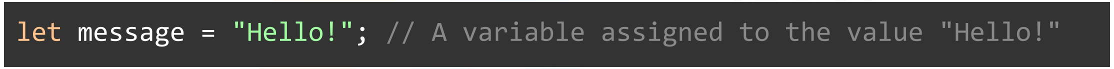
```
let x = 5;
let y = 6;
let x; // x is the name of the variable
console.log(x); // undefined
```

- Here, x is the variable name; since it does not contain any value, it will be undefined.

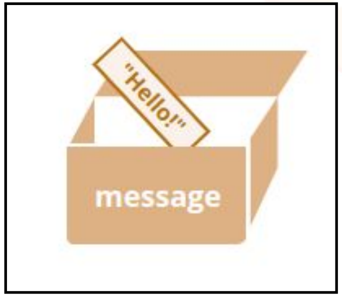
Change the Value of Variables:

- It's possible to change the value stored in the variable. For example,

- Lets declare a variable named message and assign a value “Hello!” to it



```
let message = "Hello!"; // A variable assigned to the value "Hello!"
```

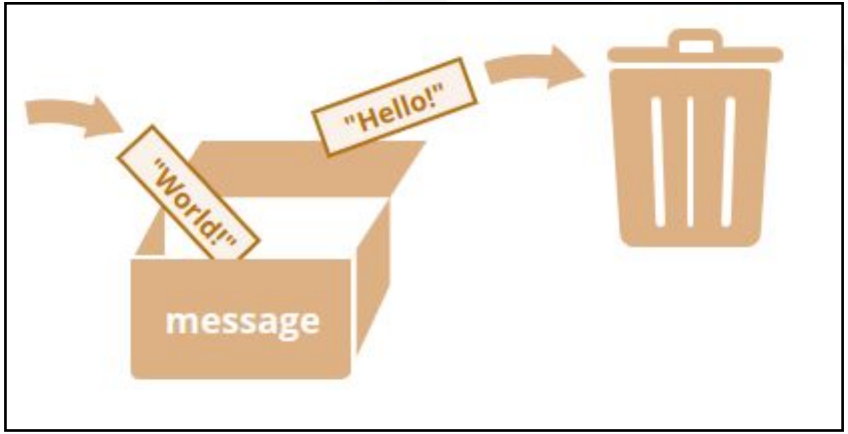


- Print the value of the message variable

```
console.log(message); // Hello! - Will be printed on the console
```

- Change the value of the variable message to “World!”

```
x = "World!";
```



- Print the value of the message variable again.

```
console.log(message);  // World! - Will be printed on console
```

  - The value of a variable may vary depending on the requirement. Hence, the name variable.

Rules for Naming JavaScript Variables:

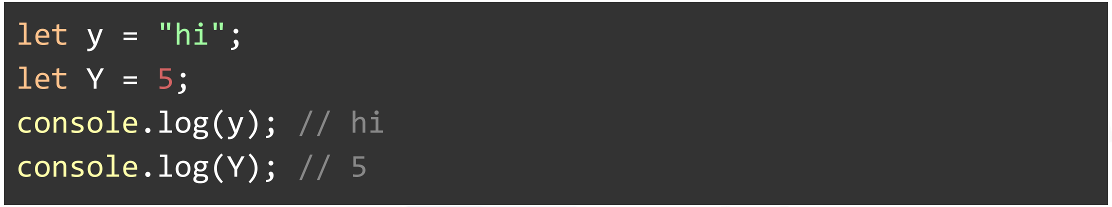
- Variable names must start with either a *letter*, an *underscore* _, or the *dollar* sign $. For example,

```
// valid
let a = 'hello';
let _a = 'hello';
let $a = 'hello';
```

- Variable names cannot start with numbers. For example,

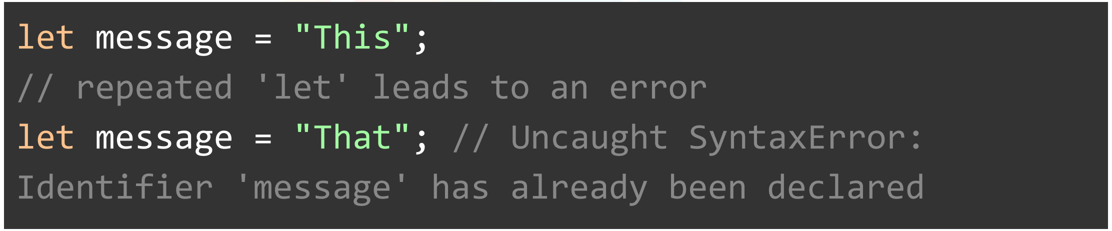
```
// invalid
let 1a = 'hello'; // this gives an error
// Error -  Uncaught SyntaxError: Invalid or unexpected token
```

- JavaScript is case-sensitive. So y and Y are different variables. For example,



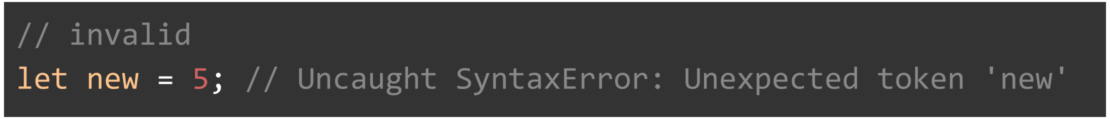
```
let y = "hi";
let Y = 5;
console.log(y); // hi
console.log(Y); // 5
```

- A variable should be declared only once.
- A repeated declaration of the same variable is an error:



```
let message = "This";
// repeated 'let' leads to an error
let message = "That"; // Uncaught SyntaxError:
Identifier 'message' has already been declared
```

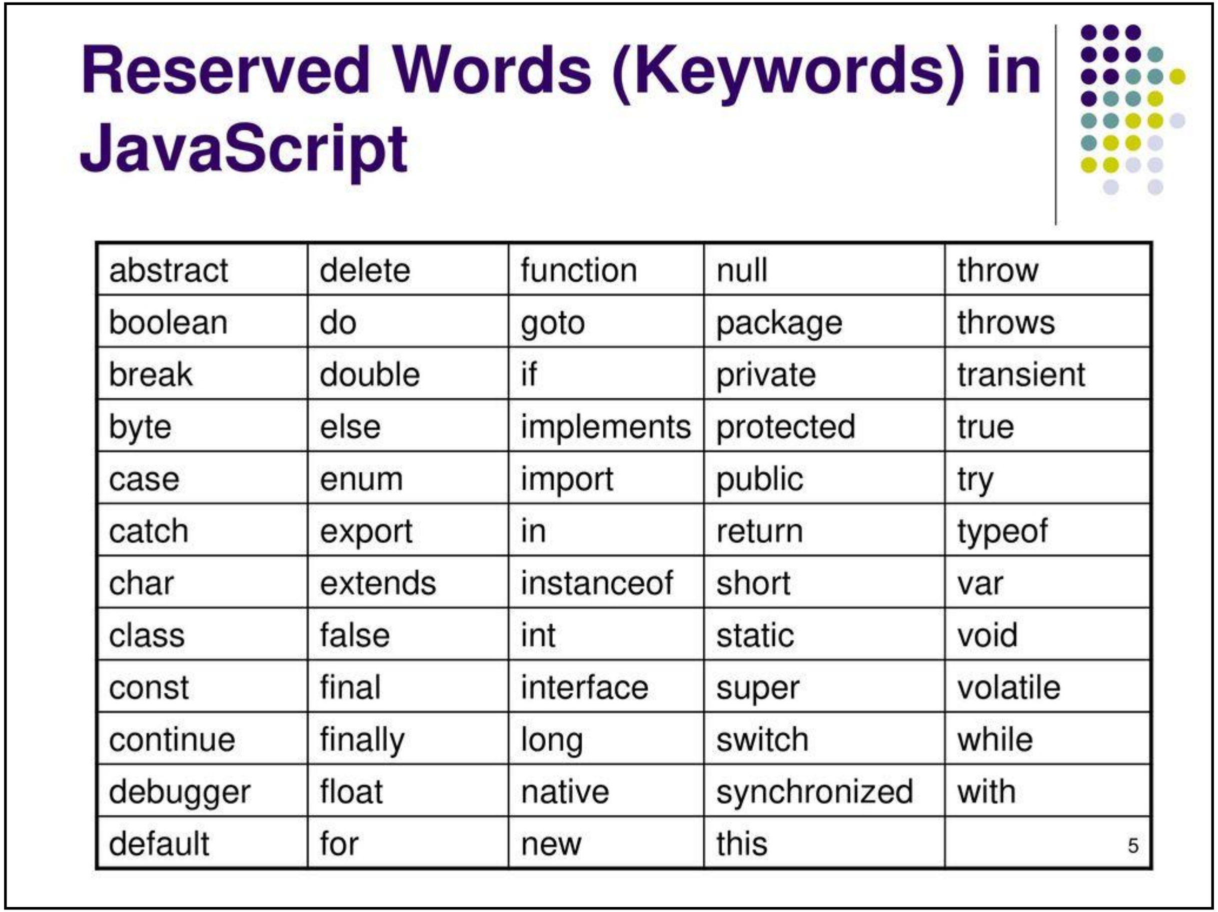
- Keywords cannot be used as variable names. For example,



```
// invalid
let new = 5; // Uncaught SyntaxError: Unexpected token 'new'
```

Reserved keywords in javascript

- its meaning and purpose are already defined within the JavaScript.
- It must be written in lowercase letters.
- it should not be used as identifiers or user-defined names.
- For naming variables, constants, arrays, functions, objects, etc., do not use keywords.

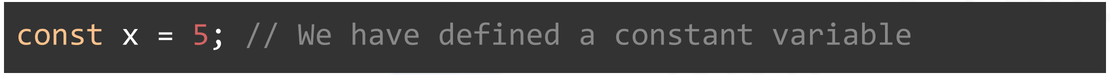


Reserved Words (Keywords) in JavaScript

abstract	delete	function	null	throw
boolean	do	goto	package	throws
break	double	if	private	transient
byte	else	implements	protected	true
case	enum	import	public	try
catch	export	in	return	typeof
char	extends	instanceof	short	var
class	false	int	static	void
const	final	interface	super	volatile
continue	finally	long	switch	while
debugger	float	native	synchronized	with
default	for	new	this	

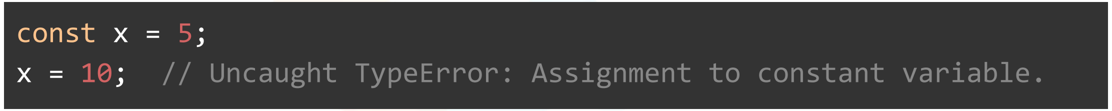
JavaScript Constants:

- The const keyword was also introduced in the ES6(ES2015) version to create constants. For example:-



```
const x = 5; // We have defined a constant variable
```

- Once a constant is initialised, we cannot change its value.



```
const x = 5;
x = 10;  // Uncaught TypeError: Assignment to constant variable.
```

- Simply, a constant is a type of variable whose value cannot be changed.

- Also, you cannot declare a constant without initialising it. For example,

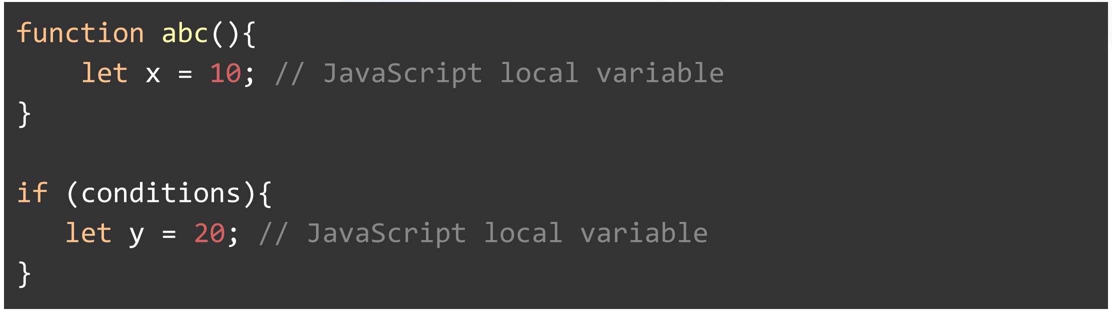
```
const x;  // Uncaught SyntaxError: Missing initializer in const
declaration
x = 5;
```

- Note – If you are sure that the value of a variable won't change throughout the program, it's recommended to use const

- There are two types of variables in JavaScript: local variable and global variable:

JavaScript local variable:

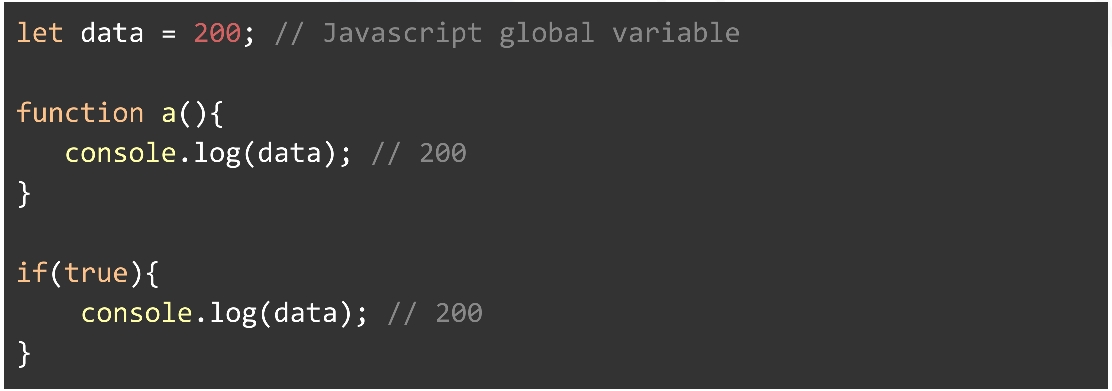
- A JavaScript local variable is declared inside a block or function. It is accessible within the function or block only. For example:



```
function abc(){
    let x = 10; // JavaScript local variable
}

if (conditions){
   let y = 20; // JavaScript local variable
}
```

JavaScript global variable:

- A JavaScript global variable is accessible from any function. A variable, i.e. declared outside the function or declared with a window object, is known as a global variable. For example:



```
let data = 200; // Javascript global variable

function a(){
   console.log(data); // 200
}

if(true){
    console.log(data); // 200
}
```

Introduction – identifiers

- An identifier is a name that is given to entities like variables, functions, classes, etc.

- All JavaScript variables must be identified with unique names.

- These unique names are called identifiers.

```
Example:
name of a variable
name of a constant
name of an array
name of a function
name of an object
etc.
```

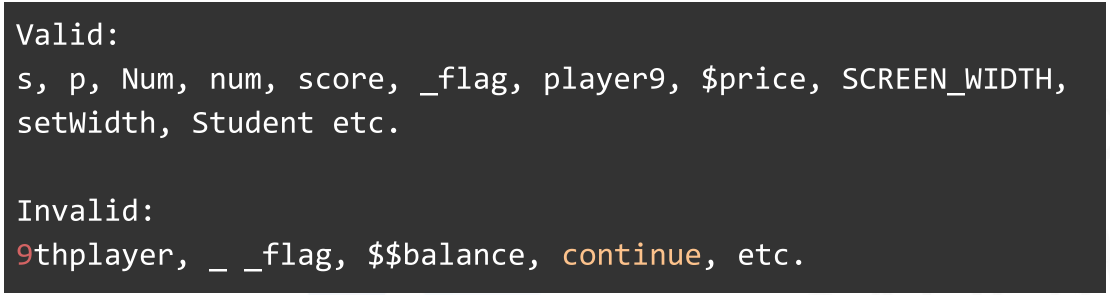
- Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume). But these Identifiers should be meaningful.

```
Example:
name, age, gender, dob, height, weight etc.
etc.
```

- The general rules for constructing names for variables (unique identifiers) are:
  - Identifiers should be meaningful.
  - Reserved keywords should never be used as identifiers.

- The first character can be an alphabet, underscore or dollar character.
- The first character should not be a number.
- All succeeding characters can be alphabets, digits, or underscores.
- No special characters are allowed except an underscore or dollar.
- More than one successive underscore or dollar should not be used.
- Identifiers are case-sensitive.

- Examples for valid and invalid identifiers:



```
Valid:
s, p, Num, num, score, _flag, player9, $price, SCREEN_WIDTH,
setWidth, Student etc.

Invalid:
9thplayer, _ _flag, $$balance, continue, etc.
```

Introduction – statements

- A computer program is a list of "instructions" to be "executed" by a computer.
- In a programming language, these programming instructions are called statements.
- A JavaScript program is a list of programming statements.
- In JavaScript statements, you can use a semicolon to end a statement.

- Semicolon will make your code more readable and help you avoid many problems that you may encounter.

- See the following example for a statement:

```
let x, y, z;    // Statement 1
x = 5;          // Statement 2
y = 6;          // Statement 3
z = x + y;      // Statement 4
var a = 10;     // Statement 5
var b = 20;     // Statement 6

let a, b, c;  // Declare 3 variables
a = 5;        // Assign the value 5 to a
b = 6;        // Assign the value 6 to b
c = a + b;    // Assign the sum of a and b to c

a = 5; b = 6; c = a + b;
```

- On the web, you might see examples without semicolons.

- Ending statements with semicolons is not required but is highly recommended.

- JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

- The following lines are equivalent:

```
let person =     "Mohan";
let person="Mohan";
```

- A good practice is to put spaces around operators (= + - * /):

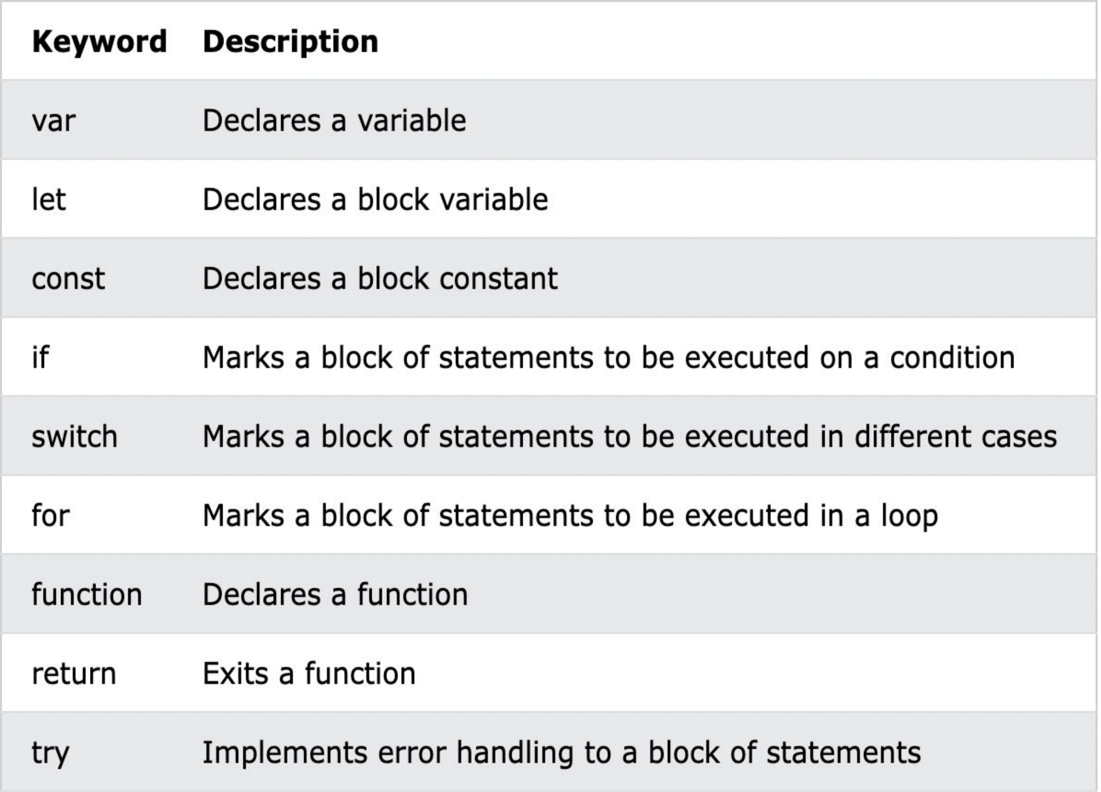
```
let x = y + z;
```

- JavaScript statements can be grouped together in code blocks inside curly brackets {...}.

- The purpose of code blocks is to define statements to be executed together.

```
Example:

if(a > b) {
   console.log('a is greater than b');
   return 1;
};
```

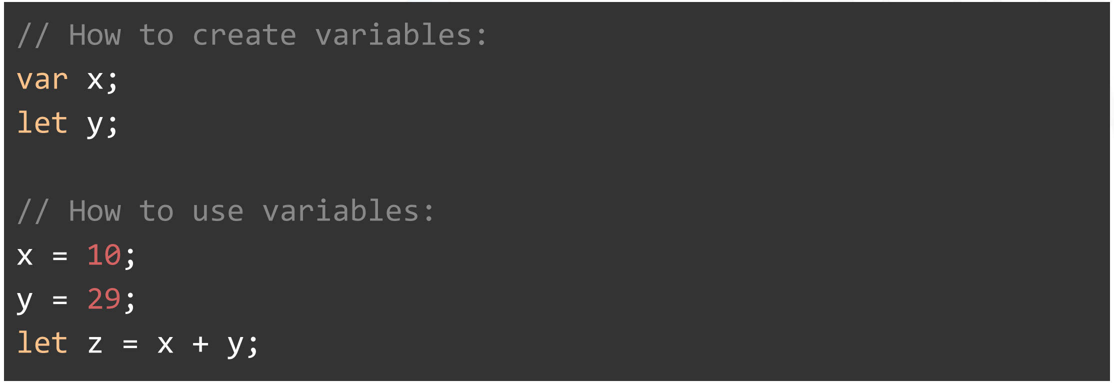
- JavaScript statements often start with a keyword to identify the JavaScript action to be performed.



Keyword	Description
var	Declares a variable
let	Declares a block variable
const	Declares a block constant
if	Marks a block of statements to be executed on a condition
switch	Marks a block of statements to be executed in different cases
for	Marks a block of statements to be executed in a loop
function	Declares a function
return	Exits a function
try	Implements error handling to a block of statements

Syntax

- JavaScript syntax is the set of rules on how JavaScript programs are constructed:



```
// How to create variables:
var x;
let y;

// How to use variables:
x = 10;
y = 29;
let z = x + y;
```

JavaScript Values

- The JavaScript syntax defines two types of values:

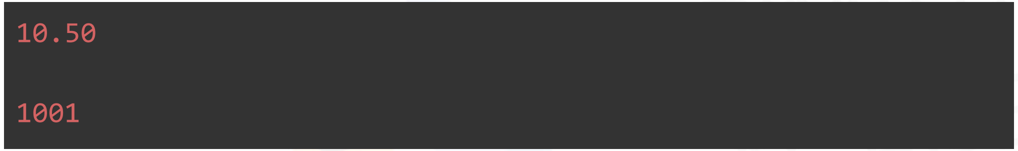
```
1. Fixed values
2. Variable values
```

- Fixed values are called *Literals*.

- Variable values are called *Variables*.

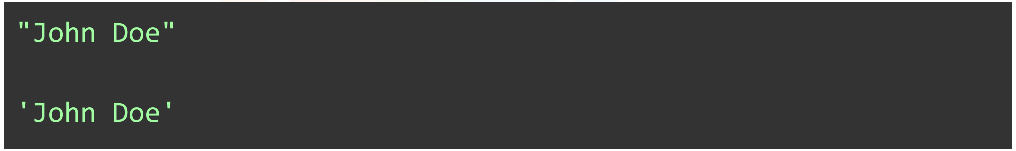
JavaScript Literals

- The two most important syntax rules for fixed values are:
  - Numbers are written with or without decimals:



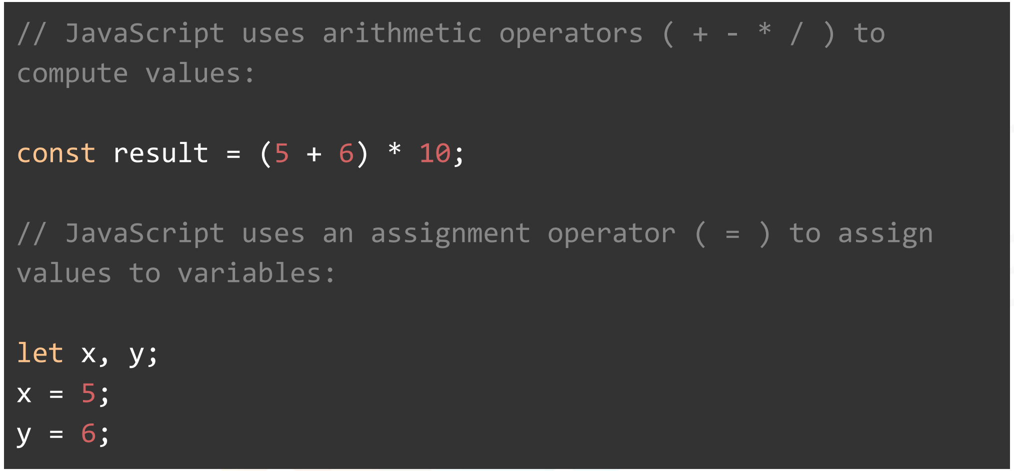
```
10.50

1001
```

  - Strings are text written within double or single quotes:



```
"John Doe"

'John Doe'
```

- Example



```
// JavaScript uses arithmetic operators ( + - * / ) to
compute values:

const result = (5 + 6) * 10;

// JavaScript uses an assignment operator ( = ) to assign
values to variables:

let x, y;
x = 5;
y = 6;
```

JavaScript and Camel Case

- Historically, programmers have used different ways of joining multiple words into one variable name:

- Hyphens (-) :
  - first-name, last-name, master-card, inter-city.
  - Hyphens are not allowed in JavaScript. They are reserved for subtractions.

- Underscore (_) :

- first_name, last_name, master_card, inter_city.

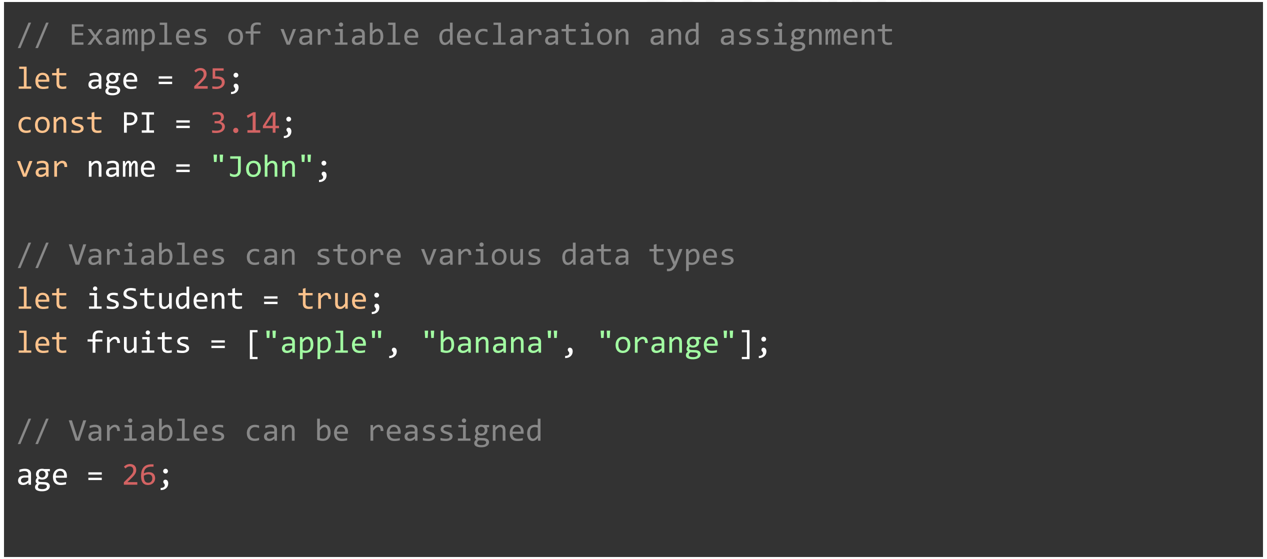
- Upper Camel Case (Pascal Case) :
  - FirstName, LastName, MasterCard, InterCity.

- Lower Camel Case :
  - firstName, lastName, masterCard, interCity.
  - JavaScript programmers tend to use camel case that starts with a lowercase letter:

Examples

1. Variables:

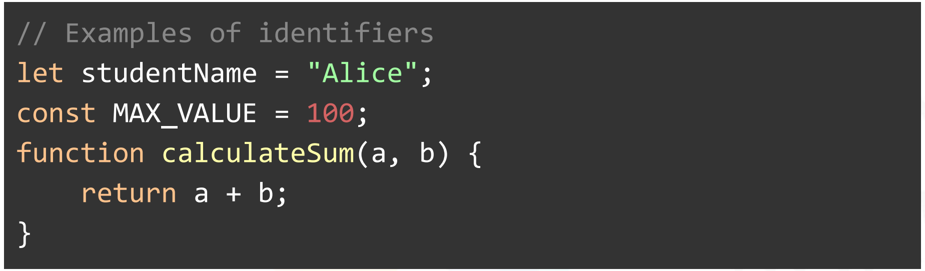
- Variables are used to store and manipulate data in JavaScript.



```
// Examples of variable declaration and assignment
let age = 25;
const PI = 3.14;
var name = "John";

// Variables can store various data types
let isStudent = true;
let fruits = ["apple", "banana", "orange"];

// Variables can be reassigned
age = 26;
```

2. Identifiers:

Identifiers are names given to variables, functions, labels, or any other user-defined items.



```
// Examples of identifiers
let studentName = "Alice";
const MAX_VALUE = 100;
function calculateSum(a, b) {
    return a + b;
}
```

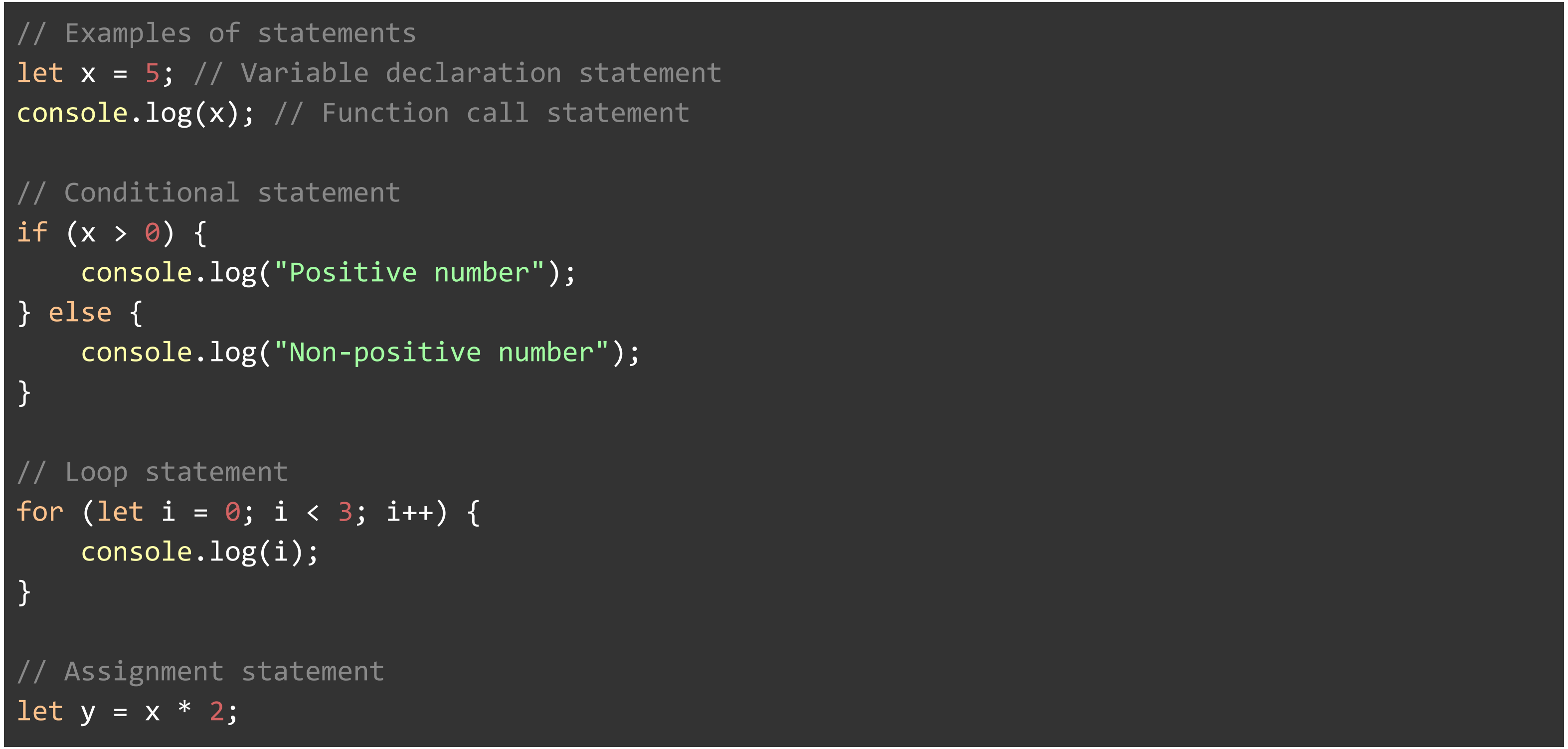
Identifiers must follow certain rules:

- Must start with a letter, underscore (_), or dollar sign ($).
- Can include letters, numbers, underscores, or dollar signs.

- Are case-sensitive.

3. Statements:

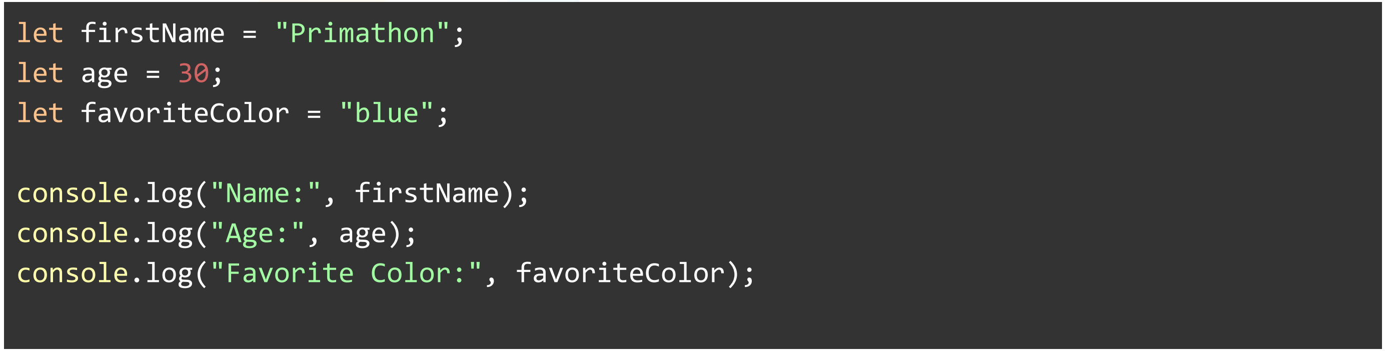
Statements are individual units of code that perform actions or control the flow of the program.



```
// Examples of statements
let x = 5; // Variable declaration statement
console.log(x); // Function call statement

// Conditional statement
if (x > 0) {
    console.log("Positive number");
} else {
    console.log("Non-positive number");
}

// Loop statement
for (let i = 0; i < 3; i++) {
    console.log(i);
}

// Assignment statement
let y = x * 2;
```

Live Assignment

- Declare three variables: firstName, age, and favoriteColor.
- Assign values to these variables with your own information.
- Log the values of these variables to the console.



```
let firstName = "Primathon";
let age = 30;
let favoriteColor = "blue";

console.log("Name:", firstName);
console.log("Age:", age);
console.log("Favorite Color:", favoriteColor);
```

Summary & Key Points

- Variables store data values that can be changed later on.
- Variables can be defined using the var keyword. Variables defined without the var keyword become global variables.
- Variables must be initialised before accessing it.
- JavaScript allows multiple white spaces and line breaks in a variable declaration.

- Multiple variables can be defined in a single line separated by a comma.
- JavaScript is a loosely typed language so that a variable can store any type of value.
- Variable names are case-sensitive.
- Variable names can contain letters, digits, or the symbols $ and _. It cannot start with a digit 0 - 9.
- Variables can have local or global scope. Local variables cannot be accessed out of the function where they are declared, whereas the global variables can be accessed from anywhere.

- JavaScript programmers tend to use camel cases that start with a lowercase letter.

PRIMATHON ACADEMY

Self Learn Assignment

- https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables1
- https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_operators1
- https://www.tutorialsteacher.com/online-test/javascript-test1

Interview Question & Answer

- What are variables in JavaScript?
- What is the difference between let, const, and var when declaring variables?
- Explain the difference between undefined and null in JavaScript.
- How do you check the type of a variable in JavaScript?
- What is an identifier in JavaScript?

- Explain the rules for naming variables and functions in JavaScript.
- What is the significance of case sensitivity in JavaScript identifiers?
- Why is it considered good practice to use meaningful variable and function names?
- What is a statement in JavaScript?
- Explain the difference between declaration and initialisation of a variable.

Answer

Reference/Resources Link

https://www.programiz.com/javascript/keywords-identifiers

https://www.tutorialsteacher.com/javascript/javascript-variable