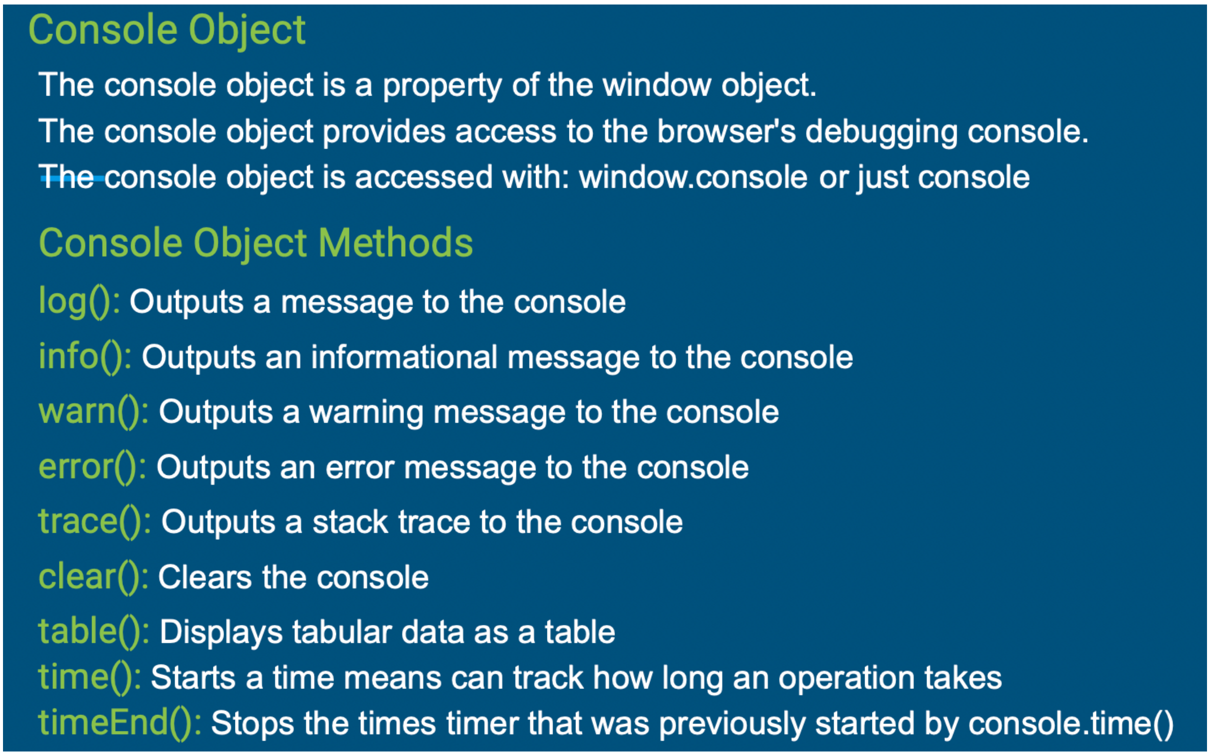
Console object methods used to log output on console.



Console Object

The console object is a property of the window object.
The console object provides access to the browser's debugging console.
The console object is accessed with: window.console or just console

Console Object Methods

log(): Outputs a message to the console
info(): Outputs an informational message to the console
warn(): Outputs a warning message to the console
error(): Outputs an error message to the console
trace(): Outputs a stack trace to the console
clear(): Clears the console
table(): Displays tabular data as a table
time(): Starts a time means can track how long an operation takes
timeEnd(): Stops the times timer that was previously started by console.time()

JS Programs few points:
Whitespace and Line Breaks: JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs

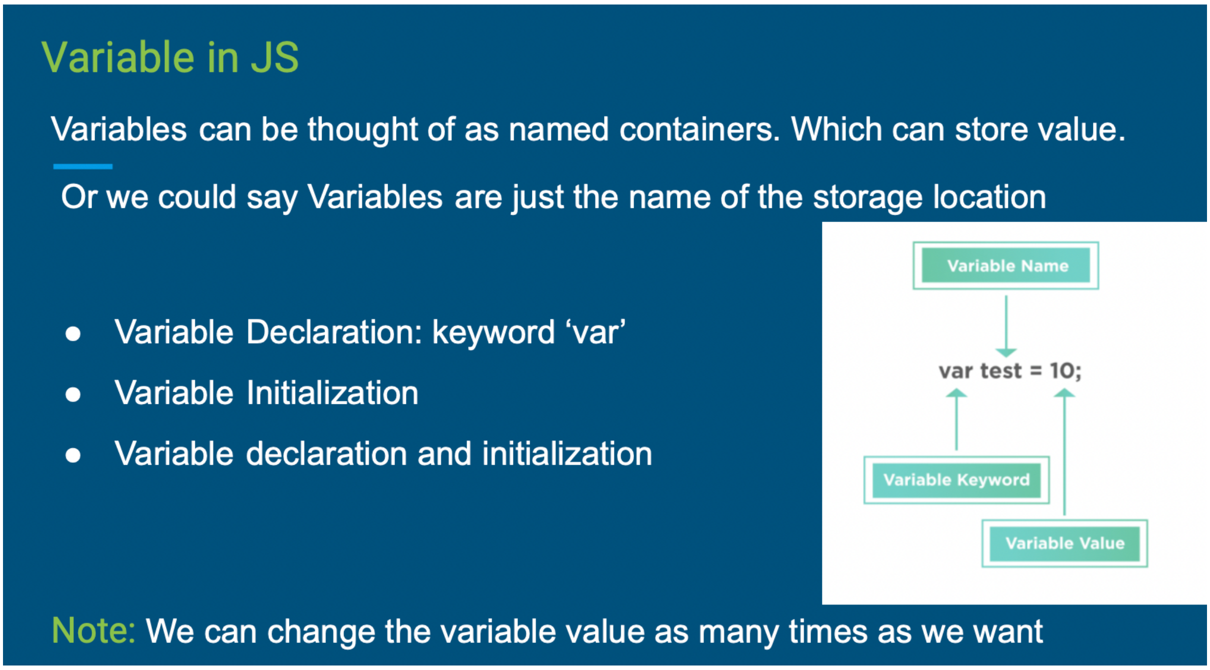
Semicolons are Optional

File Extension: JS file should have **.js** extension

Case Sensitivity: This means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

Comments: comments can be used to explain JavaScript code, and to make it more readable

- Single line comments: starts with **//**
- Multi-line comments: start tart with **/*** and end with ***/**



Variable in JS

Variables can be thought of as named containers. Which can store value.

Or we could say Variables are just the name of the storage location

- Variable Declaration: keyword 'var'
- Variable Initialization
- Variable declaration and initialization

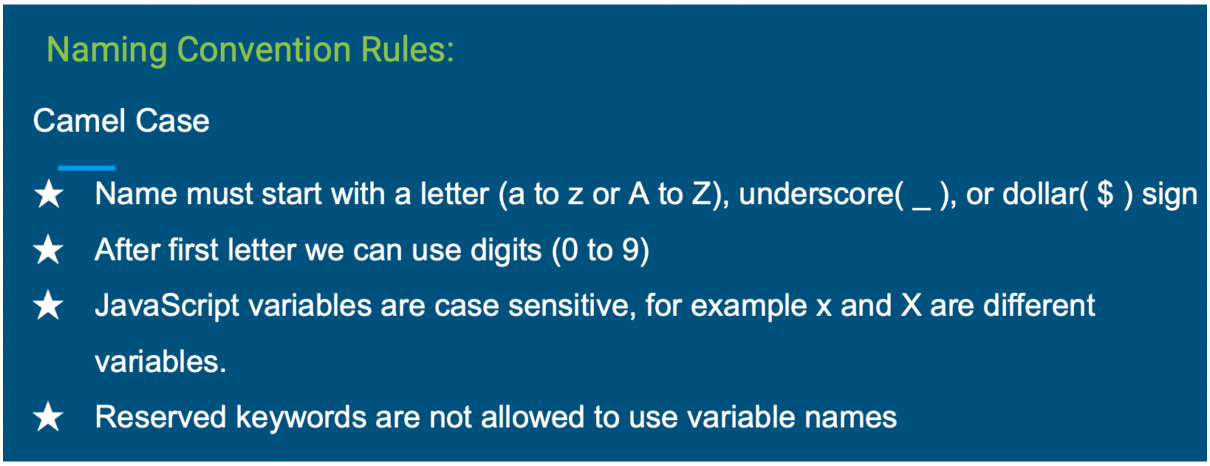
Variable Name

var test = 10;

Variable Keyword

Variable Value

Note: We can change the variable value as many times as we want



Naming Convention Rules:

Camel Case

★ Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign

★ After first letter we can use digits (0 to 9)

★ JavaScript variables are case sensitive, for example x and X are different variables.

★ Reserved keywords are not allowed to use variable names

Variable Naming Convention Rule:

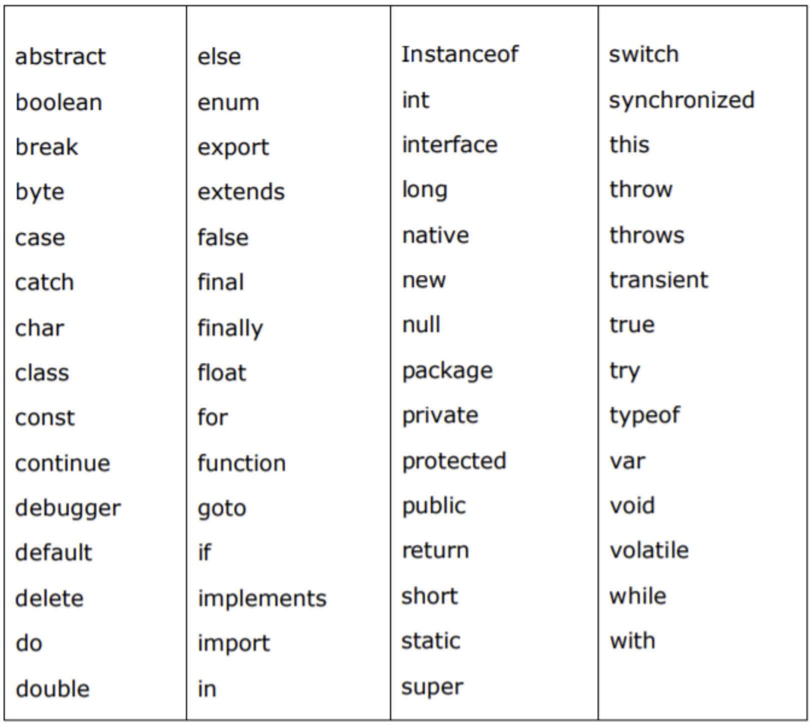
While naming our variables in JavaScript, following rules must be followed.

- We should not use any of the JavaScript reserved keywords as a variable name. These keywords are mentioned in the next section.
  For example, break or boolean variable names are not valid.

- JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character.
  For example, 123test is an invalid variable name but _123test is a valid one.
- JavaScript variable names are case-sensitive. For example, Money and money are two different variables

Reserved Keywords

A list of all the reserved words in JavaScript are given in the following table. They cannot be used as JavaScript variables, functions, methods, loop labels, or any object names.



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

JS Variables

Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container

Variable Declaration:

Before we use a variable in a JavaScript program, we must declare it. Variables are declared with the var keyword as follows.

Example:

```
var money;
var name;
```

We can also declare multiple variables with the same var keyword as follows:

Example: var money, name;

Variable initialization: Storing a value in a variable is called variable initialization.

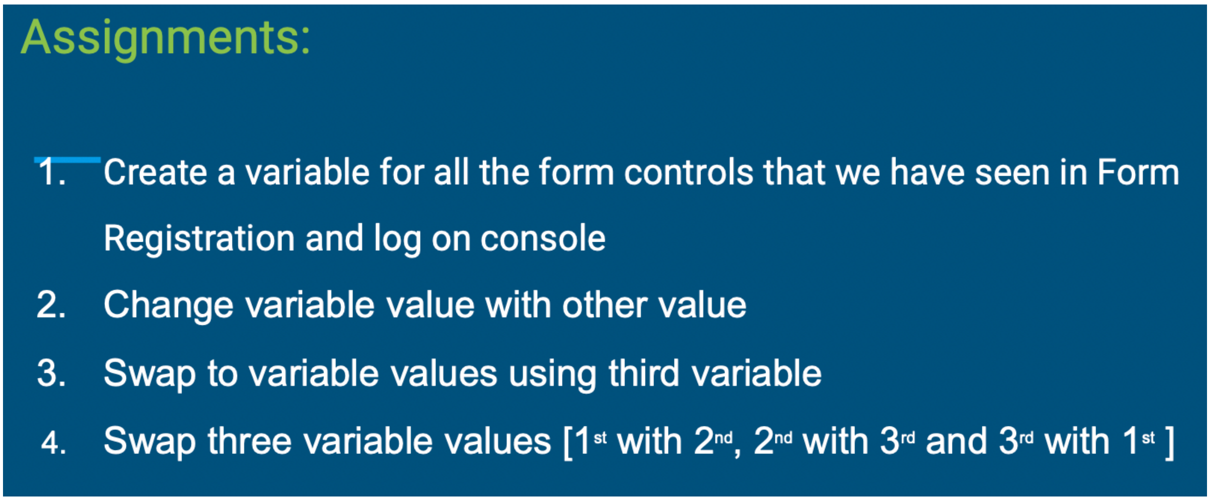
You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

Example:

```
var name = "Ali";
var money;
money = 2000.50;
```

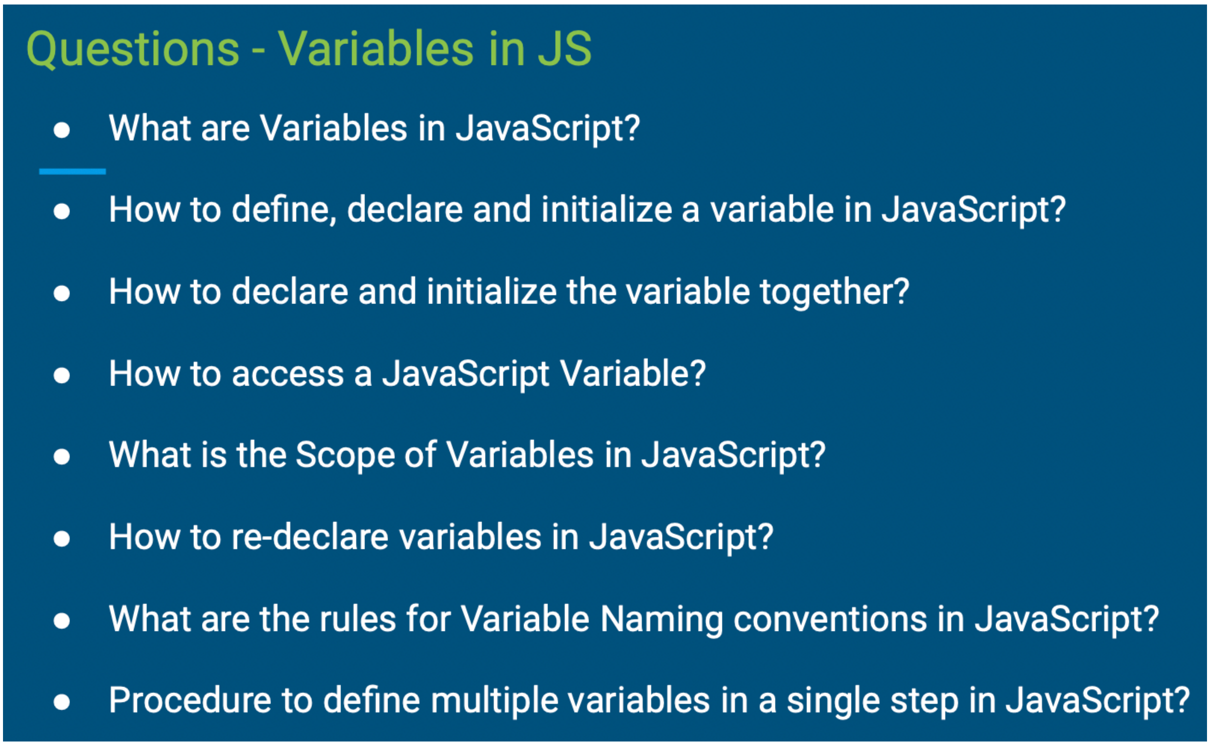
Note: Use the var keyword only for declaration or initialization, once for the life of any variable name in a document. You should not re-declare same variable twice

JavaScript is un-typed language. This means that a JavaScript variable can hold a value of any data type.



Assignments:

1. Create a variable for all the form controls that we have seen in Form Registration and log on console
2. Change variable value with other value
3. Swap to variable values using third variable
4. Swap three variable values [1st with 2nd, 2nd with 3rd and 3rd with 1st]



Questions - Variables in JS

- What are Variables in JavaScript?
- How to define, declare and initialize a variable in JavaScript?
- How to declare and initialize the variable together?
- How to access a JavaScript Variable?
- What is the Scope of Variables in JavaScript?
- How to re-declare variables in JavaScript?
- What are the rules for Variable Naming conventions in JavaScript?
- Procedure to define multiple variables in a single step in JavaScript?