JavaScript Keywords

JavaScript keywords are reserved words that have special meanings and purposes within the language. These keywords cannot be used as identifiers (such as variable names or function names) in JavaScript. Here's a list of JavaScript keywords:

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

The keywords marked with an asterisk (*) are not currently used in the ECMAScript specification but are reserved for possible future use. Additionally, some keywords like **‘class’**, **‘enum’**, **‘export’**, **‘extends’**, **‘import’**, and **‘super’** are introduced in later versions of JavaScript (ES6 and later).

It's essential to note that you cannot use these keywords as identifiers in your JavaScript code, such as variable names, function names, or labels. Doing so will result in a syntax error.