Javascript Keywords

by Atomic Badger

1-27-2025

On 1-27-2025, I asked ChatGPT the following to learn about Javascript keywords.

“What are the keywords of javascript as a programming language? please provide in a numbered list, with a brief description of each one. make sure the information is updated to today's date.”

ChatGPT said:
“Here’s a list of JavaScript keywords and reserved words (updated to today’s date), along with brief descriptions:

---

1. break

Used to exit a loop or switch statement prematurely.

2. case

Defines a condition within a switch statement.

3. catch

Handles errors in a try...catch block.

4. class

Defines a class, introduced in ES6 for creating objects with methods and constructors.

5. const

Declares a block-scoped constant variable.

6. continue

Skips the rest of the current loop iteration and moves to the next one.

7. debugger

Pauses code execution for debugging (if debugging is enabled).

8. default

Specifies the default code block in a switch statement if no case matches.

9. delete

Removes a property from an object.

10. do

Creates a do...while loop, which runs the block at least once before checking the condition.

11. else

Specifies an alternative block of code if the if condition is false.

12. enum

Reserved for future use; not implemented in JavaScript.

13. export

Used to export functions, objects, or variables from a module.

14. extends

Used in class declarations or expressions to create a child class.

15. false

Represents a Boolean false value.

16. finally

Defines a block of code that executes after try and catch, regardless of the outcome.

17. for

Creates a loop with three optional expressions (initialization, condition, and increment).

18. function

Declares a function.

19. if

Executes a block of code conditionally based on a Boolean expression.

20. implements

Reserved for future use; not implemented in JavaScript.

21. import

Used to import functions, objects, or variables from other modules.

22. in

Checks if a specified property exists in an object.

23. instanceof

Tests whether an object is an instance of a specific class or constructor.

24. interface

Reserved for future use; not implemented in JavaScript.

25. let

Declares a block-scoped variable, introduced in ES6.

26. new

Creates an instance of a class or constructor function.

27. null

Represents the absence of any value.

28. package

Reserved for future use; not implemented in JavaScript.

29. private

Reserved for future use; not implemented in JavaScript.

30. protected

Reserved for future use; not implemented in JavaScript.

31. public

Reserved for future use; not implemented in JavaScript.

32. return

Exits a function and optionally specifies a return value.

33. static

Defines a static method or property on a class (accessible without instantiating the class).

34. super

Calls the constructor of the parent class or accesses its methods.”

This was the complete answer given.

35. switch

Defines a multi-way branch statement.

36. this

Refers to the current execution context (object).

37. throw

Throws an exception.

38. true

Represents a Boolean true value.

39. try

Defines a block of code to test for errors.

40. typeof

Returns the type of a variable or expression as a string.

41. var

Declares a variable (function-scoped, less commonly used after ES6).

42. void

Returns undefined and is often used to discard a function's return value.

43. while

Creates a loop that runs while a specified condition is true.

44. with

Extends the scope chain for a statement (not recommended; considered deprecated).

45. yield

Used with generator functions to pause and resume execution.

Strict Mode Reserved Words

Some words are reserved for strict mode or future JavaScript versions:

- **arguments**: Refers to the arguments passed to a function.
- **eval**: Executes JavaScript code as a string.
- **await**: Used with async to wait for promises.

Notes:

- Some keywords, like enum, implements, package, interface, private, and protected, are reserved but not actively used.
- Modern JavaScript practices emphasize the use of let, const, class, and modules (import/export) over older constructs like var."

This was the complete answer given.