**Reserved key words**

**In JavaScript, there are several reserved words that have special meanings and cannot be used as identifiers (such as variable names, function names, or labels) in your code. These reserved words are part of the JavaScript language syntax and are used for various purposes, such as defining control flow, declaring variables, defining functions, and more. Here are some examples of JavaScript reserved words:**

**Keywords:**

**break**

**case**

**catch**

**class**

**const**

**continue**

**debugger**

**default**

**delete**

**do**

**else**

**export**

**extends**

**finally**

**for**

**function**

**if**

**import**

**in**

**instanceof**

**let**

**new**

**return**

**super**

**switch**

**this**

**throw**

**try**

**typeof**

**var**

**void**

**while**

**with**

**yield**

**Reserved for Future Use:**

**enum (reserved for future use in ECMAScript)**

**Strict Mode Only:**

**implements**

**interface**

**package**

**private**

**protected**

**public**

**static**

**Contextual Keywords (used in strict mode or as future reserved words):**

**await**

**null**

**true**

**false**

***It's important to note that while these words are reserved and cannot be used as identifiers, they can still be used as property names in objects. Additionally, some words might have specific meanings or restrictions depending on the context in which they are used, such as strict mode or module code.***

***When naming variables, functions, or other identifiers in your JavaScript code, avoid using these reserved words to prevent conflicts and ensure clarity and maintainability of your code.***