|  |  |
| --- | --- |
| Object Literal | |
|  | Object literal is a comma-seperated list of name-value pairs wrapped in curly braces. |
|  | Example  const car = {  make: ‘Renault’,  model: ‘Duster’,  year : 2016  } |
|  | Object Literal propery values can be of any data type, numbers, string, Boolean, array, functions or even nested object literal |
|  | Property name need to be enclosed in quotes if it contains spaces, special characters, or a reserved keyword or starts with a number |
|  | There should not be a comma after last name-value pair. (This will throw error in older browsers) |
|  | Object Properties can be accessed in two ways.   * Using dot   car.make // Renault  car.model // Duster   * Using Brackets   car[“make”] // Renault  car[“model”] // Duster |
|  | Object propertis can be accessed only using bracket notation if property name is not a valid variable definition means if they contain space, special characters, reserved keywords or starts with numbers |
|  | Object Properties whose value is a function is called Object method. So Method is a function stored as a property |
|  | Object methods can be executed using paranthese  car.start( )  car[‘start car’]( ) |
|  | There are 3 native ways to list/travesrse object properties   * \* [for...in](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for...in) loops This method traverses all enumerable properties of an object and its prototype chain * \* [Object.keys(obj)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/keys) This method returns an array with all the own (not in the prototype chain) enumerable properties' names ("keys") of an object obj. * \* [Object.getOwnPropertyNames(obj)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/getOwnPropertyNames) This method returns an array containing all own properties' names (enumerable or not) of an object obj. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |