Assume you have studied javascript documentation on dictionaries methods where you know all the names of methods then Provide only all the methods you studied from beginner to advance without leaving any method where I will be satisfied by the response and can able to solve any leetcode problems

Here are all the methods related to JavaScript dictionaries (known as **objects**) that you should be familiar with from beginner to advanced:

Beginner Level

1. **Object.keys(obj)** - Returns an array of a given object's own property names.
2. **Object.values(obj)** - Returns an array of a given object's own property values.
3. **Object.entries(obj)** - Returns an array of the object's enumerable property [key, value] pairs.
4. **Object.hasOwnProperty(prop)** - Returns `true` if the object has the specified property as its own property.
5. **Object.is(obj1, obj2)** - Determines if two values are the same value.
6. **Object.assign(target, ...sources)** - Copies all enumerable own properties from one or more source objects to a target object.

Intermediate Level

7. **Object.getOwnPropertyDescriptor(obj, prop)** - Returns the descriptor of a property in an object.
8. **Object.getOwnPropertyNames(obj)** - Returns an array of all properties (including non-enumerable) of the object.

9. **Object.getPrototypeOf(obj)** - Returns the prototype of the specified object.
10. **Object.defineProperty(obj, prop, descriptor)** - Defines a new property or modifies an existing property of an object.
11. **Object.defineProperties(obj, props)** - Defines new properties or modifies existing properties of an object based on a given descriptors object.
12. **Object.freeze(obj)** - Freezes an object, making it immutable.
13. **Object.seal(obj)** - Seals an object, preventing new properties from being added and existing properties from being removed, but allows modifications to existing properties.
14. **Object.preventExtensions(obj)** - Prevents new properties from being added to an object.

Advanced Level

15. **Object.isExtensible(obj)** - Checks if an object is extensible (can have new properties added).
16. **Object.isSealed(obj)** - Checks if an object is sealed (cannot have properties added or removed).
17. **Object.isFrozen(obj)** - Checks if an object is frozen (its properties cannot be modified).
18. **Object.create(proto, propertiesObject)** - Creates a new object with the specified prototype and optional properties.
19. **Object.fromEntries(entries)** - Transforms a list of key-value pairs into an object.
20. **Object.prototype.toString()** - Returns a string representing the object.
21. **Object.prototype.hasOwnProperty()** - Tests if the object has the specified property as its own (not inherited).
22. **Object.prototype.propertyIsEnumerable()** - Tests if the specified property is enumerable.

ES2022 (Modern) and Advanced Techniques

23. **Object.getOwnPropertyDescriptors(obj)** - Returns all properties of an object along with their descriptors (getter, setter, enumerable, writable).

24. **Object.fromEntries(iterable)** - Transforms a list of key-value pairs into an object. Useful for arrays and maps.

25. **Object.setPrototypeOf(obj, prototype)** - Sets the prototype (internal `[[Prototype]]`) of a specified object.

Other Advanced Techniques and Methods

26. **Object.assign()** - Shallow copies properties from source objects to the target object, good for cloning.

27. **Object.entries()** - Useful in iteration and converting key-value pairs into an array.

28. **Object.values()** - Useful for transforming object values to an array.

29. **Object.getOwnPropertySymbols()** - Returns an array of all symbols found as properties on the object.

By mastering these methods, you'll be able to efficiently work with objects (dictionaries) and solve Leetcode problems, as these methods help in manipulating, querying, and modifying object properties, making them very useful for algorithmic problem-solving.

Let me know if you need further explanations or examples for any of these methods!