

## Untitled

Here's the entire explanation in Hindi:

1. `var`, `let`, और `const` में अंतर:

`var`: यह एक ऐसा वेरिएबल डिक्लेयर करता है जिसका स्कोप फंक्शन-स्तरीय या ग्लोबल-स्तरीय होता है। `var` से डिक्लेयर किए गए वेरिएबल को फिर से री-डिक्लेयर और री-असाइन किया जा सकता है।

`let`: यह एक ब्लॉक-स्कोप्ड वेरिएबल डिक्लेयर करता है। इसे री-असाइन किया जा सकता है, लेकिन एक ही स्कोप में इसे री-डिक्लेयर नहीं किया जा सकता।

`const`: यह एक ब्लॉक-स्कोप्ड वेरिएबल डिक्लेयर करता है, जिसे एक बार इनिशियलाइज़ करने के बाद फिर से असाइन नहीं किया जा सकता।

2. JavaScript में Closure क्या है?

Closure एक ऐसी फंक्शन है जो अपनी Lexical Scope तक पहुँच बनाए रखती है, भले ही वह फंक्शन बाहरी स्कोप से बाहर execute हो। इसका मतलब है कि एक अंदर की फंक्शन, बाहरी फंक्शन के वेरिएबल्स तक पहुँच सकती है, भले ही बाहरी फंक्शन का execution समाप्त हो चुका हो।

3. Hoisting क्या है?

Hoisting JavaScript का वह default

## Untitled

behavior है जिसमें सभी डिक्लेरेशन्स (initializations नहीं) को उनके स्कोप के शीर्ष पर ले जाया जाता है। यह var डिक्लेरेशन्स पर लागू होता है, लेकिन let, const, या फ़ंक्शन एक्सप्रेसन्स पर नहीं।

4. this keyword का उपयोग क्या है?

this keyword उस ऑब्जेक्ट को दर्शाता है जो वर्तमान में कोड के उस हिस्से को चला रहा है। इसका मान इस बात पर निर्भर करता है कि फ़ंक्शन को किस तरह कॉल किया गया है:

Global Context में: this ग्लोबल ऑब्जेक्ट को दर्शाता है।

Method में: this उस ऑब्जेक्ट को दर्शाता है जिसमें method है।

Event Handler में: this उस element को दर्शाता है जिसने event को trigger किया।

5. == और === में अंतर क्या है?

== (Equality Operator): यह दो मानों को बराबरी के लिए तुलना करता है, प्रकार परिवर्तन (type coercion) के साथ।

=== (Strict Equality Operator): यह दोनों मानों और प्रकारों की बराबरी करता है, बिना किसी प्रकार परिवर्तन

## Untitled

के।

### 6. Event Loop क्या है?

Event Loop वह मैकेनिज़्म है जो JavaScript को non-blocking asynchronous ऑपरेशन करने की अनुमति देता है। यह कॉल स्टैक को निरंतर चेक करता है और जो फ़ंक्शन ready होते हैं, उन्हें execute करता है।

### 7. Promise क्या है?

Promise एक ऑब्जेक्ट है जो किसी asynchronous ऑपरेशन की eventual completion (या failure) को दर्शाता है। एक Promise तीन स्थितियों में हो सकता है:

Pending: प्रारंभिक स्थिति।

Fulfilled: ऑपरेशन सफलतापूर्वक पूरा हो गया।

Rejected: ऑपरेशन विफल हो गया।

### 8. Synchronous और Asynchronous प्रोग्रामिंग में अंतर क्या है?

Synchronous प्रोग्रामिंग: कोड को लाइन द्वारा लाइन execute किया जाता है। प्रत्येक ऑपरेशन अगले ऑपरेशन के शुरू होने से पहले पूरा होने का इंतजार करता है।

## Untitled

**Asynchronous** प्रोग्रामिंग: कोड स्वतंत्र रूप से चलता है, जिससे प्रोग्राम अन्य ऑपरेशन (जैसे नेटवर्क अनुरोध या टाइमर) के पूरा होने का इंतजार किए बिना `execute` हो सकता है।

9. `bind()` method क्या है?

`bind()` एक नया फ़ंक्शन बनाता है जिसका `this` keyword निश्चित रूप से उस `value` को सेट करता है जो प्रदान की गई होती है। यह मुख्य रूप से `callback functions` में `this` की सही `context` को बनाए रखने के लिए उपयोगी होता है।

10. **Higher-order functions** क्या होते हैं?

**Higher-order functions** वे फ़ंक्शन होते हैं जो एक या अधिक फ़ंक्शन्स को एरग्युमेंट के रूप में लेते हैं या किसी फ़ंक्शन को `return` करते हैं। जैसे `map()`, `filter()`, और `reduce()`।

11. `null` और `undefined` में अंतर क्या है?

`null`: यह किसी `object` के वैल्यू की जानबूझकर अनुपस्थिति को दर्शाता है। यह एक `object type` है।

`undefined`: यह उस वेरिएबल को दर्शाता है जिसे डिक्लेयर तो किया गया है लेकिन उसे कोई मान असाइन नहीं किया गया।

12. **JavaScript डेटा प्रकार** क्या हैं?

**JavaScript** में दो प्रकार के डेटा होते हैं:

## Untitled

Primitive Types: String, Number, BigInt, Boolean, Undefined, Symbol, और Null।

Non-Primitive (Reference) Types: Object (जिसमें Arrays और Functions शामिल हैं)।

13. JavaScript में Array क्या है?

Array एक ऑर्डर्ड कलेक्शन है जो विभिन्न डेटा प्रकार के मानों को रख सकता है। इसमें कई बिल्ट-इन methods होते हैं जैसे push(), pop(), shift(), unshift(), map() आदि।

14. Destructuring क्या है?

Destructuring एक तरीका है जिससे हम arrays और objects से values को अलग-अलग वेरिएबल्स में unpack कर सकते हैं। यह डेटा को अधिक संक्षिप्त और पठनीय बनाने में मदद करता है।

15. Arrow function क्या है?

Arrow function एक शॉर्टकट है जो functions को लिखने का तरीका है। यह => सिंटैक्स का उपयोग करता है और इसका अपना this नहीं होता, बल्कि यह surrounding lexical scope से inherit करता है।

## Untitled

DOM (Document Object Model) और BOM (Browser Object Model) Methods:

DOM Methods:

DOM वह इंटरफ़ेस है जो JavaScript को HTML और XML दस्तावेजों के साथ इंटरैक्ट करने की अनुमति देता है। यह दस्तावेज़ संरचना को एक ट्री के रूप में प्रस्तुत करता है, जिसमें हर नोड दस्तावेज़ के एक हिस्से का प्रतिनिधित्व करता है।

प्रमुख DOM Methods:

`getElementById()` - ID द्वारा एक एलिमेंट प्राप्त करता है।

`getElementsByClassName()` - कक्षा द्वारा एलिमेंट्स की एक लिस्ट प्राप्त करता है।

`getElementsByTagName()` - टैग नाम द्वारा एलिमेंट्स की एक लिस्ट प्राप्त करता है।

`querySelector()` - CSS सेलेक्टर द्वारा पहला मेल खाने वाला एलिमेंट प्राप्त करता है।

`querySelectorAll()` - CSS सेलेक्टर द्वारा सभी मेल खाने वाले एलिमेंट्स की `NodeList` प्राप्त करता है।

## Untitled

`createElement()` - एक नया एलिमेंट नोड बनाता है।

`appendChild()` - एक नोड को पैरेंट नोड के रूप में जोड़ता है।

`removeChild()` - एक चाइल्ड नोड को पैरेंट से हटाता है।

`setAttribute()` - एलिमेंट पर एक एट्रिब्यूट सेट करता है।

`getAttribute()` - एलिमेंट से किसी एट्रिब्यूट का मान प्राप्त करता है।

### BOM Methods:

BOM वह ऑब्जेक्ट्स होते हैं जो ब्राउज़र के साथ इंटरैक्ट करने की अनुमति देते हैं, जैसे `window`, `location`, और `navigator`।

### प्रमुख BOM Methods:

`window.alert()` - एक अलर्ट बॉक्स दिखाता है।

`window.confirm()` - एक कन्फर्म बॉक्स दिखाता है।

`window.prompt()` - एक पॉपअप बॉक्स दिखाता है जिसमें यूज़र से इनपुट लिया जाता है।

## Untitled

`window.open()` - एक नई विंडो या टैब खोलता है।

`window.close()` - वर्तमान विंडो या टैब को बंद करता है।

यदि आपको और विस्तार से किसी भी टॉपिक पर जानकारी चाहिए, तो बताइए!

You said:

oops ke all method hindi me bta

ChatGPT said:

Here is a detailed explanation of all the main Object-Oriented Programming (OOP) concepts and methods in Hindi:

1. ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग (OOP) क्या है?

ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग (OOP) एक प्रोग्रामिंग पैटर्न है, जो



## Untitled

"ऑब्जेक्ट्स" (objects) पर आधारित है। ऑब्जेक्ट्स, कक्षाओं (classes) के उदाहरण होते हैं। OOP का मुख्य उद्देश्य कोड को इस तरह से संरचित करना है कि डेटा और उस डेटा पर काम करने वाली विधियाँ (methods) एक साथ एक इकाई के रूप में हों। इससे कोड की पुनः उपयोगिता, विस्तारशीलता (scalability) और रख-रखाव (maintainability) में सुधार होता है।

### 2. OOP के मुख्य सिद्धांत

OOP के चार मुख्य सिद्धांत हैं:

**एन्कैप्सुलेशन (Encapsulation):** इसका मतलब है कि डेटा (attributes) और विधियाँ (methods) को एक साथ एक कक्षा (class) में रखना, ताकि डेटा को बाहर से सीधे एक्सेस न किया जा सके। डेटा तक पहुँच को नियंत्रित करने के लिए **getter** और **setter** का उपयोग किया जाता है।

उदाहरण: बैंक अकाउंट में सीधे बैलेंस को बदलने के बजाय, केवल डिपॉजिट और विथड्रॉइट जैसे कार्य उपलब्ध होते हैं।

**एब्स्ट्रैक्शन (Abstraction):** यह जटिलताओं को छुपाने और केवल आवश्यक विवरणों को प्रदर्शित करने का कार्य करता है। यह उपयोगकर्ता को सिस्टम की जटिलताओं से बचाता है और केवल उच्च-स्तरीय कार्यों पर ध्यान केंद्रित करता है।

## Untitled

उदाहरण: कार की शुरुआत करने या रोकने के लिए शुरू(), रोकें() जैसे मेथड होते हैं, जिनका उपयोग हम करते हैं, लेकिन हमें यह नहीं पता कि इंजन किस प्रकार काम करता है।

इनहेरिटेंस (Inheritance): इसमें एक कक्षा दूसरी कक्षा से गुण और कार्य (methods) प्राप्त करती है। इससे कोड का पुनः उपयोग संभव होता है और विभिन्न कक्षाओं के बीच संबंध स्थापित होता है।

उदाहरण: एक "Car" कक्षा "Vehicle" कक्षा से गुण प्राप्त कर सकती है, जैसे start(), stop(), आदि।

पॉलीमॉर्फिज्म (Polymorphism): यह किसी एक नाम के मेथड को विभिन्न कक्षाओं में अलग-अलग तरीके से लागू करने की क्षमता है। यह मेथड ओवरलोडिंग (method overloading) और मेथड ओवरराइडिंग (method overriding) के माध्यम से हो सकता है।

उदाहरण: एक "Shape" कक्षा का draw() मेथड होगा, लेकिन "Circle" और "Rectangle" जैसी कक्षाएँ अपने-अपने तरीके से इसे लागू करेंगी।

3. कक्षा (Class) क्या है?

कक्षा (Class) वस्तुओं (objects) को बनाने के लिए एक ब्लूप्रिंट होती है। यह उन गुणों (attributes) और विधियों

Untitled

(methods) को परिभाषित करती है जो उस कक्षा के वस्तु में होंगे।