

Chapter - 3

JavaScript Basic and Element of JavaScript

(JavaScript बेसिक्स या JavaScript के तत्व)

JavaScript में लगभग वे सभी [प्रोग्रामिंग](#) क्षमताएं होती हैं जो अधिकांश [प्रोग्रामिंग भाषाओं](#) में पाई जाती हैं जैसे variable, control structure, constants, user define functions आदि इन सभी [Programming](#) तकनीकों का प्रयोग किसी भी HTML डॉक्यूमेंट में डाले गए JavaScript कोड में किया जा सकता है JavaScript की इन तकनीकों के कारण HTML की क्रियाशीलता बढ़ जाती है और वेब पेज interactive बन जाते हैं।

JavaScript के चरों (Variable), नियतांको (Constants), फ़ंक्शन (Function) आदि को परिभाषित करने के लिए HTML डॉक्यूमेंट का <head> section सबसे आदर्श स्थान है। इसका कारण यह है कि यह सेक्शन हमेशा <body> section से पहले प्रोसेस किया जाता है JavaScript code में उपयोग किए जा रहे Variables को <head> सेक्शन में परिभाषित करने से उनका आगे स्वतंत्रतापूर्वक उपयोग किया जा सकता है यह बहुत महत्वपूर्ण है, क्योंकि किसी तत्व को बिना घोषित या परिभाषित किए उपयोग करने पर गलत संदेश प्राप्त होगा।

1. Data type and Literals/Constant (डेटा टाइप और अचर)

JavaScript में किसी Variable का Data type पहले से घोषित नहीं किया जाता अतः आप एक ही Variable को अलग अलग समय पर अलग-अलग प्रकार का डाटा स्टोर करने के लिए भी उपयोग कर सकते हैं। वैसे JavaScript में Variables को 4 प्राथमिक प्रकार का Constant Data दिया जा सकता है जो निम्न प्रकार हैं-

- **Number (संख्या)**

संख्याएं सामान्यता पूर्णांक (integer) अथवा फ्लोटिंग पॉइंट (floating point) हो सकती हैं पूर्णांक में कोई दशमलव बिंदु नहीं होता जबकि फ्लोटिंग पॉइंट संख्या में दशमलव बिंदु का प्रयोग किया जाता है फ्लोटिंग पॉइंट संख्याओं में घात (exponent) भी हो सकता है जो E अक्षर के बाद दिया जाता है उदाहरण के लिए 12, 0, -4, 333 आदि सभी पूर्णांक संख्याएं हैं और 1.0, 5.345, -56.3, 24.4E4 यह सभी फ्लोटिंग पॉइंट संख्याएं हैं इनके अतिरिक्त JavaScript में एक विशेष NaN (not a Number) मान भी होता है।

- **Boolean (बुलियन)**

बुलियन चर या अचर के दो मान हो सकते हैं true and false बुलियन व्यंजको में लॉजिकल ऑपरेटर जैसे AND, OR, NOT आदि का प्रयोग किया जा सकता है JavaScript बुलियन मानो true और false को संख्यात्मक व्यंजनों में प्रयोग किए जाने वाले अपने आप क्रमशः 1 और 0 में बदल देता है।

ध्यान रहे कि जावा स्क्रिप्ट में 1 और 0 को बुलियन मान नहीं माना जाता।

- **String (स्ट्रिंग)**

सिंगल (Single) या डबल (Double) कोटेशन चिन्हों में रखे गए शून्य या अधिक चिन्हों को स्ट्रिंग कहा जाता है उदाहरण के लिए "Ashok", 'Ram' यह सभी स्ट्रिंग हैं यदि किसी संदर्भ चिह्न को स्ट्रिंग में शामिल करना है तो उससे पहले एक backslash (\) लगाना चाहिए। उदाहरण के लिए, यदि किसी string में आप Don't रखना चाहते हैं तो इसे या तो "Don't" लिखें या 'Don\'t' इस तरह लिखें।

- **Null (नल)**

यह केवल एक मान null को व्यक्त करता है जो खालीपन दिखाता है जावा स्क्रिप्ट में सामान्यतया इसका उपयोग चरों को प्रारंभ में घोषित करते समय प्रारंभिक मान रखने में किया जाता है यदि ऐसा ना किया जाए तो किसी variable में अनपेक्षित मान भी भरा हो सकता है।

2. Variable (चर)

Variable (चर) विभिन्न मानो को स्टोर करने के लिए प्रयोग किए जाते हैं और उनका कोई नाम भी रखा जाता है, जिनके द्वारा उनका संदर्भ दिया जाता है चरों के नाम ऐसे रखे जाने चाहिए कि उनसे इसका पता चलता हो कि किस चर का क्या उपयोग किया जाएगा अर्थात उस variable में कौन सा मान स्टोर किया जाएगा variables के नाम अंग्रेजी वर्णमाला के अक्षरों a से z तथा A से Z अथवा अंडरस्कोर (_) से प्रारंभ हो सकते हैं।

JavaScript में अंग्रेजी वर्णमाला के छोटे और बड़े अक्षरों को अलग-अलग माना जाता है अर्थात JavaScript केस सेंसिटिव है जैसे सुविधा की दृष्टि से हम चरों के नाम हमेशा छोटे अक्षरों में रखते हैं यदि नाम बड़ा है तो प्रत्येक शब्द का पहला अक्षर केपिटल कर देते हैं और शेष अक्षर छोटे रहते हैं जैसे FirstName, DateOfBirth, TotalMarks आदि आप इनमे से किसी भी परंपरा को अपना सकते हैं HTML में space का कोई महत्व नहीं है परंतु JavaScript में इनका महत्व होता है।

Creating Variable (चर बनाना)

जावा स्क्रिप्ट में चरों को पहले से बना लेना अर्थात घोषित करना अनिवार्य नहीं है परंतु ऐसा करना अच्छी प्रोग्रामिंग परंपरा है जैसे हम बिना घोषित किए भी किसी चर का उपयोग कर सकते हैं चर घोषित करने के लिए जावा स्क्रिप्ट में var आदेश का उपयोग किया जाता है इसका सामान्य रूप निम्न प्रकार है-

```
Var<variable nam>=value;
```

जहां <variable name> उस चर का नाम है और Value उसका प्रारंभिक मान है बरबा चिन्ह (=) का प्रयोग उस चर का मान निर्धारित करने के लिए किया जाता है इसलिए इस ऑपरेटर को एक असाइनमेंट ऑपरेटर (assignment operator) कहा जाता है चरों का प्रारंभिक मान रखना वैकल्पिक (optional) है।

```
var first_name = "Ashish Kumar";
```

```
var roll_no;
```

```
var phone_no = 256485;
```

3. Operator and Expressions (ऑपरेटर और व्यंजक)

किसी ऑपरेटर का प्रयोग एक या अधिक मानों को केवल एक मान में परिवर्तित करने के लिए किया जाता है जिन मानों पर ऑपरेटर को लागू किया जाता है उन्हें Operands कहा जाता है किसी ऑपरेटर और उसके operands के संयोग को मुद्रा या व्यंजक (expression) कहा जाता है किसी व्यंजक का मान निकालने के लिए उसमें दिए गए ऑपरेटरों को उनके operands के न्यूनतम मान पर लागू किया जाता है और अंत में एक परिणामी मान निकाला जाता है।

JavaScript में उपयोग किए जाने वाले ऑपरेटर निम्न प्रकार हैं-

- **Arithmetic Operators (अंकगणितीय ऑपरेटर)**

इनका प्रयोग गणितीय क्रियाएं अर्थात गणनाएं करने के लिए किया जाता है JavaScript के अंकगणितीय ऑपरेटर निम्न सारणी में दर्शाए गए हैं-

Operator	Result
+	Addition (also unary plus)
-	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
-=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
--	Decrement

जिस ऑपरेटर के लिए केवल एक operand की आवश्यकता होती है उसे यूनेरी (unary) ऑपरेटर कहा जाता है और जिसके लिए दो operands की आवश्यकता होती है उसे Binary Operator कहते हैं सभी प्रचलित अंकगणितीय ऑपरेटर Binary हैं, जबकि ++ और -- यूनेरी ऑपरेटर हैं।

वृद्धि (++) और कमी (-) ऑपरेटरों को दो प्रकार से उपयोग में लाया जा सकता है operand पहले और operand के बाद उदाहरण के लिए ++X देने पर x का मान पहले एक से बढ़ाया जाएगा फिर परिणाम लौटाया जाएगा जबकि x++ देने पर पहले x का मान घटाया जाएगा फिर उसे एक से बढ़ा दिया जाएगा इसी प्रकार --X देने पर x का मान पहले एक से घटाया जाएगा और फिर परिणाम लौटाया जाएगा जबकि X-- देने पर पहले x का मान लौटाया जाएगा फिर उसे एक से घटा दिया जाएगा उदाहरण के लिए निम्नलिखित निर्धारण कथनों पर ध्यान दीजिए-

```
X=3;
Y=X++;
Z=++X;
```

यहां पहले कथन के कारण Variable x का मान 3 रख दिया जाएगा। दूसरे कथन से Variable Y का मान पहले Variable X के बराबर अर्थात 3 रखा जाएगा। फिर X को एक से बढ़ा दिया जाएगा अर्थात अब x का मान 4 हो जाएगा। तीसरे कथन से पहले variable X का मान 1 से बढ़ाया जाएगा अर्थात x का मान 5 हो जाएगा फिर वह मान variable Z में लौटाया जाएगा अर्थात Z का मान 5 होगा। इस प्रकार कथनों के परिणाम स्वरूप X का मान 3, Y का मान 3 और Z का मान 5 हो जायेगा ।

- **Logical Operators (तार्किक ऑपरेटर)**

इन ऑपरेटर का उपयोग Boolean Operands पर Boolean Operations करने के लिए किया जाता है | Java Script में केवल तीन तार्किक ऑपरेटर होते हैं, जो निम्न सारणी में दर्शाए गए हैं |

Logical Operator	Java Operator
AND	&&
OR	
NOT	!

- **Comparison Operators (तुलनात्मक ऑपरेटर)**

इन ऑपरेटर्स का प्रयोग दो मानों की तुलना करने के लिए किया जाता है और इनका परिणाम Boolean मानों अर्थात True अथवा False में होता है। JavaScript में उपयोग किये जाने वाले तुलना ऑपरेटर निम्न सारणी में दर्शाए गए हैं-

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True
===	Equal value and same type	5 === 5	True
		5 === "5"	False
!==	Not Equal value or Not same type	5 !== 5	False
		5 !== "5"	True

- **Assignment Operators (निर्धारण ऑपरेटर)**

किसी निर्धारण ऑपरेटर का उपयोग किसी चर का मान बदलने अर्थात उसका नया मान रखने के लिए किया जाता है | मूल निर्धारण ऑपरेटर केवल एक है - '=', जिसे कुछ अंकगणितीय ऑपरेटर्स के साथ मिलकर अन्य निर्धारण ऑपरेटर बनाये गए हैं। जावास्क्रिप्ट में उपलब्ध निर्धारण ऑपरेटर निम्न सारणी में दर्शाए गए हैं -

- **String Operator (स्ट्रिंग ऑपरेटर)**

इस ऑपरेटर का प्रयोग केवल स्ट्रिंगो पर क्रियाये करने के लिए किया जाता हैं | जावास्क्रिप्ट में ऐसा केवल एक ऑपरेटर हैं +, जिसे स्ट्रिंग योग (String concatenation) ऑपरेटर कहा जाता हैं | इस ऑपरेटर का उपयोग दो स्ट्रिंगो को मिलाकर एक स्ट्रिंग बनाने के लिए किया जाता हैं| उदाहरण के लिए, "abc"+"opq" का परिणाम "abcopq" होगा|

- **Special Operator (विशेष ऑपरेटर)**

जावास्क्रिप्ट में कई ऑपरेटर भी हैं , जो ऊपर बताई गई किसी श्रेणी में नहीं आते | इन्हें विशेष ऑपरेटर कहा जाता हैं| ऐसे तीन प्रमुख ऑपरेटर हैं, जिनका परिचय नीचे दिया गया हैं-

Delete इस ऑपरेटर का प्रयोग किसी array के किसी तत्व को हटाने के लिए किया जाता हैं |

new इस ऑपरेटर का प्रयोग किसी array के किसी तत्व को जोड़ने के लिए किया जाता हैं|

void यह ऑपरेटर कोई मान नहीं लौटाता| सामान्यतया इसका उपयोग किसी URL को खली मान देने के लिए किया जाता हैं|