Chapter – 5

जावास्क्रिप्ट में कण्ट्रोल फ्लो स्टेटमेंट क्या है ?

Control Flow statements

Control statements program के flow को control करते है। जैसे की आप control statements की मदद से आप चुन सकते है की आप कौन सा स्टेटमेंट execute करवाना चाहते है और कौन सा नहीं करवाना चाहते है। Control statements की मदद से logic perform किया जाता है।

सरल शब्दों में कह सकते है की ,आप चुन सकते है की कौन-सा स्टेटमेंट किस situation में execute होगा,और साथ ही आप control statements की मदद से एक statement को कई बार execute कर सकते है।

Types of Control Flow Statements

Control statements को तीन केटेगरी में बांटा गया है। ये केटेगरी control statements के tasks को भी define करती है।

- Selection statements/Conditional statements
- Looping statements
- Jump statements

Selection statements / Conditional statements

इस category के control statements program में statements को situation के अनुसार सेलेक्ट करके execute करने के लिए प्रयोग किये जाते है। ये निम्न प्रकार के होते है-

- If
- If-Else
- Nested-If
- Switch case

If Statement

If statement, condition को test करता है यदि condition true होती है तो brackets में दिए हुआ statement execute होता है और यदि condition false है तो control बाहर चला जाता है।

Example:1

```
if(10>3)
{
```

```
document.write("This will be displayed");
}
```

जैसा की आप ऊपर दिए हुए उदाहरण में देख रहे है दी हुई condition true है इसलिए brackets के अंदर का statement execute होगा। आइये इसका एक और उदाहरण देखते है।

```
if(3>10)
{
document.write("This will not be displayed");
}
```

इस उदाहरण में condition false है इसलिए brackets में दिया हुआ statement execute नहीं होगा।

If else statement

If else statement भी if statement की तरह ही होता है। बस इसमें else वाला भाग भी जोड़ दिया जाता है। Else part में आप वो statements लिखते है जो condition false होने पर execute होना चाहिए। आइये इसका उदाहरण देखते है।

```
if(num%2==0)
{
document.write("Number is Positive");
}
else
{
document.write("Number is Negative");
}
```

Else If Statement

यदि आप चाहते है की एक condition के false होने पर else part को execute ना करके किसी दूसरी condition को check किया जाये तो इसके लिए आप else if statements use कर सकते है।

Else if statements के द्वारा आप एक से अधिक conditions को check कर सकते है और सभी condition के false होने पर else part को execute करवा सकते है।इसके लिए आप elseif keyword यूज़ करते है। First condition को normal if else statement की तरह execute किया जाता है। इसके अलावा आप जितनी भी conditions add करना चाहते है उन्हें if और else part के बीच elseif keyword के द्वारा डिफाइन करते है।इसका सिंटेक्स निम्नानुसार है –

```
if(condition 1)
{
// Will be executed if above condition is true
}
else if(condition 2)
{
// Will be executed if 1st condition is false and this condition is true.
}

....
....
....
else if(condition N)
{
// Will be executed if all the conditions above it were false and this condition is true.
}
else
{
// Will be executed if all the above conditions are false
}
```

Example :2

```
if(num>0)
{
document.write("Number is Positive");
}
elseif( num==0)
{
document.write("Number is Zero");
}
else
{
document.write(" Number is Negative");
}
```

आइये अब इसे एक उदाहरण से समझने का प्रयास करते है।

Nested If Statement

यदि आप आप चाहे तो एक if condition में दूसरी if condition भी डाल सकते है। इसका structure नीचे दिया जा रहा है।

```
if(condition)
{
if(condition)
{
// Statement to be executed
}
}
else
{
// Statements to be executed
}
```

जैसा की आप ऊपर दिए गए syntax में देख सकते है एक if condition के अंदर दूसरी if condition define की गयी है। आप चाहते तो nested if में else part भी add कर सकते है। आइये अब इसे एक उदाहरण से समझने का प्रयास करते है।

Example:3

```
if(5>3)
{
if(5>6)
{
document.write("This will not be executed");
}
else
{
document.write("5 is greater than 3 but not 6");
}
}
else
{
document.write("5 is not greater than 3");
}
```

Switch Case

Switch case बिलकुल if statement की तरह होता है। लेकिन इसमें आप एक बार में एक से ज्यादा conditions को check कर सकते है। Switch case में cases define किये जाते हैं। बाद में एक choice variable के द्वारा ये cases execute करवाए जाते है। Choice variable जिस case से match करता है वही case execute हो जाता है। इसका उदाहरण नीचे दिया जा रहा है।

Example :4

```
var day=2;
// Passing choice to execute desired case
switch(day)
{
case 1:
document.write("Monday");
break;
case 2:
document.write("Tuesday");
break;
case 3:
document.write("Wednesday");
break;
case 4:
document.write("Thursday");
break;
case 5:
document.write("Friday");
break;
case 6:
document.write("Saturday");
break;
default:
document.write("Wrong input");
break;
}
```

जैसे की आप देख सकते है हर case के बाद में break statement यूज़ किया गया है। यदि आप break statement यूज़ नहीं करते है तो सभी cases one by one execute हो जाते है। इस उदाहरण में variable की value 2 है इसलिए second case execute होगा और Tuesday display हो जायेगा।

Looping Statements

Looping statements particular statement को बार-बार execute करने के लिए यूज़ किये जाते है। ये 3 प्रकार के होते है। इनके बारे में नीचे दिया जा रहा है।

While Loop Statement

इस loop में आप एक condition देते है जब तक condition true होती है block में दिए गए statements execute होते रहते है। Condition false होते ही loop terminate हो जाता है और program का execution continue रहता है।

Example :5

```
var num = 0;
// While loop iterating until num is less than 5
while(num <5)
{
document.write(“Hello”);
num++;
}
```

इस उदाहरण में जब तक num 5 से कम है तब तक loop का block execute होगा। एक चीज़ यहां पर नोटिस करने की ये है की हर बार num को increment किया जा रहा है ताकि कुछ स्टेप्स के बाद loop terminate हो जाये। यदि यहां पर ऐसा नहीं किया जाये तो loop कभी terminate ही नहीं होगा infinite time तक चलेगा। इसलिए इस situation से बचने के लिए किसी भी प्रकार के loop में loop control variable को increment किया जाता है।

Do-While Loop Statement

Do while loop भी while loop की तरह ही होता है। बस ये first time बिना condition check किये execute होता है और बाद में हर बार condition check करता है। यदि condition true होती है तो do block के statements execute कर दिए जाते है। आइये इसे एक उदाहरण से समझते है।

Example :6

```
var num=0;
// Do-while loop
do{document.write(“hello”);
num++;}
while(num>5);
```

जैसा की आप देख सकते है पहले do block execute होगा और उसके बाद condition check की जाएगी। इस loop की विशेषता ये है की चाहे condition true हो या false loop एक बार तो जरूर execute होगा। यदि condition true होती है तो loop further execute होता है नहीं तो terminate हो जाता है।

For Loop Statement

सभी loops में for loop सबसे सरल है और सबसे ज्यादा प्रयोग किया जाने वाला loop है। इसमें आप single line में ही पूरे loop को define कर देते है। यदि condition true होती है तो block में दिए गए statements execute हो जाते है। इस loop का उदाहरण नीचे दिया गया है।

```
// For loop running until i is less than 5
for(var i=0;i<5;i++)
{
document.write("This will be printed until condition is true");
}
```

For loop में condition और increment दोनों एक साथ ही define किये जाते है। साथ ही इसमें loop control variable भी define किया जाता है। Condition के false होते ही loop terminate हो जाता है।

Jump Statements

Jump statements program के execution को एक जगह से दूसरी जगह transfer करने के लिए यूज़ किये जाते है। इन statements को special cases में यूज़ किया जाता है। इनके बारे में नीचे दिया जा रहा है।

Continue Statement

Continue statement के द्वारा आप किसी भी loop की कोई iteration skip कर सकते है। जैसे की आप चाहते है की 3rd iteration skip हो जाये और compiler कोई action ना ले।

Example :7

```
for(var i=0; i<5;i++)
{
if(i==2)
{
// Skipping third iteration of loop
continue;
}
document.write("This will be displayed in iterations except 3rd");
}
```

Continue statement को प्रयोग करने से compiler 3rd iteration को skip कर देगा और कोई भी statement execute नहीं किया जायेगा। इसके बाद next iteration शुरू हो जायेगी।

Break Statement

Break statement compiler के execution को stop करने के लिए यूज़ किया जाता है। Break statement आने पर compiler execution को उस block से बाहर ला देता है। इसको एक loop के example से आसानी से समझा जा सकता है।

```
for(var i=0;i<5;i++)
{
if(i==2)
{
//Breaking 3rd iteration of loop
break;
}
document.write("This will be displayed 2 times only");
}
```

उपर दिए गए उदाहरण में जैसे ही loop की 3rd iteration आती है तो break statement के द्वारा loop terminate हो जाता है और program का execution loop के बाहर से शुरू हो जाता है।