**Data Types (Verilənlərin tipləri)**

**Paradigma**

**Arithmetic operators**(Hesab operatorları)

**/** - div bölmə adlanır , tam hissə götürülür

**%** - mod bölmə adlanır , qalıq hissə götürülür

**Increment** 🡪 num++

**Declerement** 🡪 num—

int num = 10;

num++ && num-- gedən proses özündən sonraki sətrə aid olur.

++num && --num həmin sətr daxilində artıq proses getmiş olur

Console.WriteLine(num++); yazarsaq ekrana **10** çıxacaq lakin artıq yaddaşda num **11** olaraq qalacaq və daha sonra num dəyişənini ekrana verərsək **11** kimi çıxacaq

Amma Console.WriteLine(++num); yazarsaq ekrana **11** çıxacaq çünki artıq proses həmin sətrdə getmiş olacaq

**Assigment Operators**(Mənimsətmə operatorları)

a += b 🡪 a = a + b

a -= b 🡪 a = a - b

a \*= b 🡪 a = a \* b

a /= b 🡪 a = a / b

a %= b 🡪 a = a % b

a\*\*=b 🡪 a = a \*\* b sırf JavaScript’ə məxsus bir mənimsətmə operatorudur və mənası da a üstü b deməkdir

**Comparison Operators**(Müqayisə operatorları)

Müqayisə operatorları bizə true və ya false dəyər qaytarır

== 🡪 bərabərdirmi?

!= 🡪bərabərliyin əksi, bərabər deyilmi yəni not equals

> 🡪 böyükdürmü

< 🡪 kiçikdirmi

>= 🡪böyük bərabərdirmi

<= 🡪kiçik bərabərdirmi

=== 🡪 həm dəyəri həm də tipi bərabərdirmi sırf JavaScript’ə məxsus Müqayisə operatorudur.

!== 🡪 həm dəyəri həm də tipi bərabər deyilmi sırf JavaScript’ə məxsus Müqayisə operatorudur.

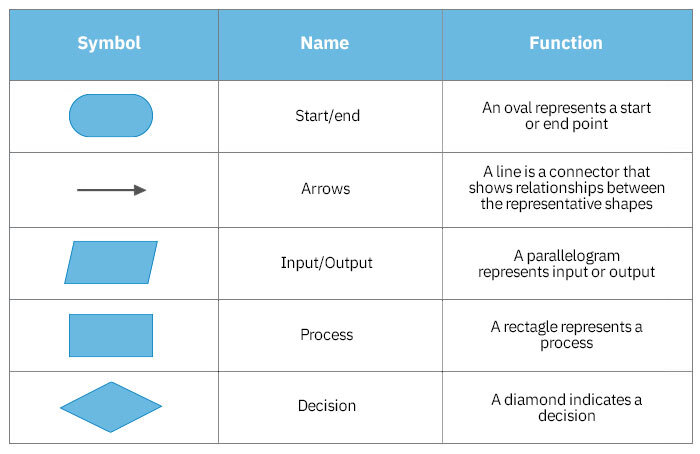
**Logical Operators**(Məntiqi operatorlar)

&& 🡪 and(və) deməkdir. Işləndiyi bütün şərtlər true olduğu halda true dəyər qaytarır

|| 🡪 or(və ya) deməkdir. Işləndiyi şərtlərin ən azı 1-I true olduğu halda true dəyər qaytarır

! 🡪 not(əks) deməkdir. Işləndiyi şərt true olarsa false , false olsarda true dəyər qaytarır.

**Flow chart**



**Condition**

İf(condition) {

Process;

}

İf – müəyyən bir şərti yoxlamaq üçün istifadə edilir. əgər daxil edilən condition true olarsa if bloku daxilindəki əməliyyatlar uyğun olaraq yerinə yetirilir. əks təqdirdə proses baş vermir

If(condition) {

Process;

}else {

Process;

}

If, else –condition true olmadığı halda uyğun olaraq else bloku daxilindəki əməliyyatlar uyğun olaraq yerinə yetirilir.

İf(condition) {

Process;

}else if(condition) {

Process;

}else {

Process;

}

İf , else if , else - əgər müəyyən bir şərti yoxladıqdan sonra əgər bu şərtimiz ödənməzsə bir başqa şərti yoxlamaq lazım olarsa else if vasitəsi ilə yeni bir condition yoxlaya bilərik.

**Switch case**

Bizim təyin edtiyimiz dəyişənin müxtəlif qiymətlərə və ya dəyərlərə bərabər olma halı oyxlanılırsa if else yerinə switch case’dən istifadə edilir.

Switch(expression) {

case value1:

Process;

break;

case value2:

Process;

break;

case valueN:

Process;

break;

default:

break;

}

əgər bizim dəyişənimizə heç bir case halı bərabər olmazsa default daxilindəki əməliyyatlar işə düşür.

**Keywords**

Break,Continue

Break keyword’ü vasitəsi ilə biz müəyyən şərt daxilində loop’u dəyandırmaq,loop’dan çıxmaq üçün istifadə edilir.

Continue keyword’ü vasitəsi ilə biz müəyyən şərt daxilində işləndiyi sətrdən sonra ki əməliyyatlar yerinə yetirilmədən loop’un əvvəlinə qayıtmaq üçün istifadə edilir.

Continue keyword’dən fərqli olaraq break keyword’ü loop’la yanaşı switch daxilində də işlənilir

**Loops**

Qarşıya qoyulmuş məsələni müəyyən şərt daxilində müəyyən say qədər yerinə yetirmək üçün istifadə edilir.

3 cür loop’dan istifadə edilir: while, do while, for

While(condition) {

Process;

}

Şərtimiz true olduğu hallarda blok daxilindəki proses dövrü olaraq davam edəcək.

Do{

Process;

}while(condition)

Do while ilə while’ın fərqi odur ki, do while zamanı şərt yoxlanılmadan ilk başda proses 1 dəfə yerinə yetirilir daha sonra şərt yoxlanılır. Bu da o deməkdir ki verilən condition false olsa, while loop’da proses heç vaxt yerinə yetirilməyəcək amma do while’da isə proses 1 dəfə yerinə yetiriləcək.

For(dataType variable; condition; process1){

Process2;

}

For loop’u 3 hissəyə bölünür 1-ci hissə dəyişən təyin etmək üçün, 2-ci hissə şərt vermək üçün və sonuncu hissə müəyyən bir əməliyyatı yerinə yetirmək üçün istifadə olunur.Şərt ödəndiyi bütün hallarda For loop’un bloku daxilindəki proses işə düşür.