

	C	C++	C#	Java	JavaScript
Değişkenler	int sayi = 10 ; char ch = 'a' ; const double Pi = 3.14 ;	int sayi = 10 ; char ch = 'a' ; const double Pi = 3.14 ;	int sayi = 10 ; char ch = 'a' ; const double Pi = 3.14 ; String name = "huseyin";	int sayi = 10 ; char ch = 'a' ; final double Pi = 3.14 ; String name = "huseyin"; boolean myBool = true;	let sayi = 1 ; let name = "huseyin" ; const pi = 3.14 ; var myBool = true ; let x={ name:"y",surname:"y"};
Veri Tipleri	int char float double bool long short	int char float double bool long short string	int char float double bool long short string	int char float bool boolean long short string	var let const
Diziler	double cars1[2]; int cars2[]={1,2};	string[5] cars1; string[2] cars2={"x","y"}; int[4] cars1; int[2] cars2={1,2};	string[] cars1; string[] cars2={"x","y"}; int[] cars1; int[] cars2={1,2};	String[] cars1; String[] cars2={"x","y"}; int[] cars1; int[] cars2={1,2};	let fruits=['Apple','Banana']; let years= [2020,2021];
Input	int giris1; float giris2; double giris4; scanf("%d", &giris1); scanf("%f", &giris2); scanf("%lf", &giris4); scanf("%d%f",&giris1,&giris2);	char name[50]; cin >> name;  int x; cin >> x;  string surname; cin >> surname;	string name; name = Console.ReadLine();  int age; age = Convert.ToInt32( Console.ReadLine());	String name; name = scanner.nextLine();  int sayi1; sayi1 = scanner.nextInt();  double sayi; sayi = scanner.nextDouble();	let giris1 = document. getElementById("numb"). value;  let giris2 = document. querySelector("#score");
Output	printf("Hello"); printf("Sayi %d", giris1);	cout << "Hello"; cout << "Name: "<<name<< endl;	Console.WriteLine("Hello"); Console.WriteLine(name);	System.out.println("Hello"); System.out.println(name);	console.log("Hello"); console.log(giris1); console.log(giris2.value);
Aritmetik Operatörler	+ : Toplam - : Çıkarma * : Çarpma / : Bölme % : Bölümünden kalan ++ : Bir arttır -- : Bir azalt	+ : Toplam - : Çıkarma * : Çarpma / : Bölme % : Bölümünden kalan ++ : Bir arttır -- : Bir azalt	+ : Toplam - : Çıkarma * : Çarpma / : Bölme % : Bölümünden kalan ++ : Bir arttır -- : Bir azalt	+ : Toplam - : Çıkarma * : Çarpma / : Bölme % : Bölümünden kalan ++ : Bir arttır -- : Bir azalt	+ : Toplam - : Çıkarma * : Çarpma ** : Üs alma / : Bölme % : Bölümünden kalan ++ : Bir arttır -- : Bir azalt
Atama Operatörleri	= : a = b += : a += b -= : a-= b *= : a*= b /= : a /= b %= : a %= b	= : a = b += : a += b -= : a-= b *= : a*= b /= : a /= b %= : a %= b	= : a = b += : a += b -= : a-= b *= : a*= b /= : a /= b %= : a %= b	= : a = b += : a += b -= : a-= b *= : a*= b /= : a /= b %= : a %= b	= : a = b += : a += b -= : a-= b *= : a*= b **=: a **= b /= : a /= b %= : a %= b

	C	C++	C#	Java	JavaScript
İlişkisel Operatörler	<code>== : a == b</code> <code>!= : a != b</code> <code>&gt; : a &gt; b</code> <code>&gt;= : a &gt;= b</code> <code>&lt; : a &lt; b</code> <code>&lt;= : a &lt;= b</code>	<code>== : a == b</code> <code>!= : a != b</code> <code>&gt; : a &gt; b</code> <code>&gt;= : a &gt;= b</code> <code>&lt; : a &lt; b</code> <code>&lt;= : a &lt;= b</code>	<code>== : a == b</code> <code>!= : a != b</code> <code>&gt; : a &gt; b</code> <code>&gt;= : a &gt;= b</code> <code>&lt; : a &lt; b</code> <code>&lt;= : a &lt;= b</code>	<code>== : a == b</code> <code>!= : a != b</code> <code>&gt; : a &gt; b</code> <code>&gt;= : a &gt;= b</code> <code>&lt; : a &lt; b</code> <code>&lt;= : a &lt;= b</code>	<code>===: a === b</code> <code>== : a == b</code> <code>!= : a != b</code> <code>!== : a !== b</code> <code>&gt; : a &gt; b</code> <code>&gt;= : a &gt;= b</code> <code>&lt; : a &lt; b</code> <code>&lt;= : a &lt;= b</code> <code>?</code>
Mantıksal Operatörler	<code>&amp;&amp; ( and ) : a &amp;&amp; b</code> <code>   ( veya ) : a    b</code> <code>! ( değil ) : !(a &amp;&amp; b)</code>	<code>&amp;&amp; ( and ) : a &amp;&amp; b</code> <code>   ( veya ) : a    b</code> <code>! ( değil ) : !(a &amp;&amp; b)</code>	<code>&amp;&amp; ( and ) : a &amp;&amp; b</code> <code>   ( veya ) : a    b</code> <code>! ( değil ) : !(a &amp;&amp; b)</code>	<code>&amp;&amp; ( and ) : a &amp;&amp; b</code> <code>   ( veya ) : a    b</code> <code>! ( değil ) : !(a &amp;&amp; b)</code>	<code>&amp;&amp; ( and ) : a &amp;&amp; b</code> <code>   ( veya ) : a    b</code> <code>! ( değil ) : !(a &amp;&amp; b)</code>
If Else	<code>if(koşul){</code> <code>//code</code> <code>}</code> <code>else{</code> <code>//code</code> <code>}</code>	<code>if(koşul){</code> <code>//code</code> <code>}</code> <code>else{</code> <code>//code</code> <code>}</code>	<code>if(koşul){</code> <code>//code</code> <code>}</code> <code>else{</code> <code>//code</code> <code>}</code>	<code>if(koşul){</code> <code>//code</code> <code>}</code> <code>else{</code> <code>//code</code> <code>}</code>	<code>if(koşul){</code> <code>//code</code> <code>}</code> <code>else{</code> <code>//code</code> <code>}</code>
While	<code>while (i &lt; 10 ){</code> <code>//code</code> <code>}</code>	<code>while (i &lt; 10 ){</code> <code>//code</code> <code>}</code>	<code>while (i &lt; 10 ){</code> <code>//code</code> <code>}</code>	<code>while (i &lt; 10 ){</code> <code>//code</code> <code>}</code>	<code>while (i &lt; 10 ){</code> <code>//code</code> <code>}</code>
Switch Case	<code>switch(degisken){</code> <code>case sabit1:</code> <code>//code</code> <code>break:</code>  <code>case sabit2:</code> <code>//code</code> <code>break:</code> <code>default:</code> <code>}</code>	<code>switch(degisken){</code> <code>case sabit1:</code> <code>//code</code> <code>break:</code>  <code>case sabit2:</code> <code>//code</code> <code>break:</code> <code>default:</code> <code>}</code>	<code>switch(degisken){</code> <code>case sabit1:</code> <code>//code</code> <code>break:</code>  <code>case sabit2:</code> <code>//code</code> <code>break:</code> <code>default:</code> <code>}</code>	<code>switch(degisken){</code> <code>case sabit1:</code> <code>//code</code> <code>break:</code>  <code>case sabit2:</code> <code>//code</code> <code>break:</code> <code>default:</code> <code>}</code>	<code>switch(degisken){</code> <code>case sabit1:</code> <code>//code</code> <code>break:</code>  <code>case sabit2:</code> <code>//code</code> <code>break:</code> <code>default:</code> <code>}</code>
For	<code>for(int a=0; i&lt;10; i++ ){</code> <code>//code</code> <code>}</code>	<code>for(int a=0; i&lt;10; i++ ){</code> <code>//code</code> <code>}</code>	<code>for(int a=0; i&lt;10; i++ ){</code> <code>//code</code> <code>}</code>	<code>for(int a=0; i&lt;10; i++ ){</code> <code>//code</code> <code>}</code>	<code>for(int a=0; i&lt;10; i++ ){</code> <code>//code</code> <code>}</code>
Yorum Satırı	<code>// C</code> <code>/* C</code> <code>C</code> <code>*/</code>	<code>// C++</code> <code>/* C++</code> <code>C++</code> <code>*/</code>	<code>// C#</code> <code>/* C#</code> <code>C#</code> <code>*/</code>	<code>// Java</code> <code>/* Java</code> <code>Java</code> <code>*/</code>	<code>// JavaScript</code> <code>/* JavaScript</code> <code>JavaScript</code> <code>*/</code>