

Változó deklarálás	
típus azonosító;	int a;
Több ugyanolyan típusú változó deklarálása	
típus azonosító, azonosító, azonosító;	long a, b, c;
Változó deklarálása és definiálása	
típus azonosító = érték;	string duma = "valami";
	long a = 200;
	char c = 'X';
	double a = 100000.55;
	float c = 200000.15F;
Több ugyanolyan típusú változó deklarálása és definiálása	
típus azonosító = érték, azonosító = érték;	int a = 50, b=100;
Adott típusú szám beolvasása billentyűzetről	
azonosító = [azonosító típusa].Parse(Console.ReadLine());	a = int.Parse(Console.ReadLine());
	a = double.Parse(Console.ReadLine());
Karakter beolvasása billentyűzetről	
A leütött karakter a képernyőre is kiíródik	char c; c = Console.ReadKey().KeyChar;
A leütött karakter nem íródik ki a képernyőre	char c; c = Console.ReadKey(true).KeyChar;
Szöveg beolvasása billentyűzetről	
Console.ReadLine();	string s; s = Console.ReadLine();
Várakozás billentyű leütésre anélkül, hogy eltárolnánk	
A leütött karakter kiíródik a képernyőre	Console.ReadKey();
A leütött karakter nem íródik ki a képernyőre	Console.ReadKey(true);
Képernyőre írás	
Üres sor kiírása	Console.WriteLine();
Fix szöveg kiírása. A kiírást ott kezdi, ahol a kurzor villog a képernyőn. A kiírás után a kurzor átkerül a következő sor elejére.	Console.WriteLine("Ezt kiírom");
Fix szöveg kiírása. A kiírást ott kezdi, ahol a kurzor villog a képernyőn. A kiírás után a kurzor csak akkor kerül át a következő sor elejére, ha elérte a sor végét.	Console.Write("Ezt kiírom");
String típusú változó értékének kiírása	string s = "Búgócsiga"; Console.Write(s);

Szám típusú változó értékének kiírása	int a = 200; Console.Write(a);
Több részből álló szöveg összefűzése + operátorral	string s = "aki", s2 = "másnak", s3 = "vermet és maga esik bele"; Console.WriteLine(s + " " + s2 + " " + s3); /*Kiírja hogy: aki másnak vermet és maga esik bele */
Így akár számokat is összefűzhetünk	string s = "forintnak"; int a = 100, b = 50; Console.WriteLine(a + " " + s + " " + b + " a fele"); /*Kiírja hogy: 100 forintnak 50 a fele */
Szöveg kiírása behelyettesítéses módszerrel	int a = 75; Console.Write("{0}+{1}={2}",100,a,100+a); /*Kiírja, hogy: 100+75=175*/ /*A színek mutatják, hogy melyik érték hova helyettesítődik be*/
Színek	
Betűszín beállítása	Console.ForegroundColor = ConsoleColor.Red;
Háttérszín beállítása	Console.BackgroundColor = ConsoleColor.Blue;
Írás pozíciójának beállítása	
Console.SetCursorPosition(x,y); Megadjuk az x és y koordinátákat, a képernyő bal felső sarka az origó.	Console.SetCursorPosition(10,5);
Kurzor elrejtése	
Console.CursorVisible = false;	Console.CursorVisible = true; /*így viszont megjelenik*/
Speciális karakterek kiírása	
idézőjel kiírása (\ " jelenti magát az idézőjelet)	Console.Write("\ "');
idézőjel kiírása ("" jelenti magát az idézőjelet, mert a @ hatására a \-jel elvesztette speciális jelentését)	Console.Write(@" "");
\-jel kiírása (azért problémás, mert a \-jelnek speciális jelentése van, \\ jelenti magát a \-jelet)	Console.Write("\\ ");
\-jel kiírása (azért problémás, mert a \-jelnek speciális jelentése van, \\ jelenti magát a \-jelet, ha viszont kiteszük a @-ot, akkor a \-jel elveszíti a speciális jelentését, így elegendő egy darab \ jelet írni)	Console.Write(@"\ ");
\t jelenti a tabulátort \n jelenti az új sort	Console.Write("\tBeljebb kezdődöm\nEz már új sorba kerül");
Szöveges utasítások	
Fontos, hogy a stringek karaktereinek sorszámozása 0-tól indul!	
Length /*megadja a szöveg karaktereinek számát*/	string s = "valami"; int a = s.Length; /*Az a változó értéke 6 lesz, mert a valami -ben hat darab karakter van*/

Substring(m,n) /*A szöveg m . karakterétől vesz n darab karaktert*/ /*Tehát nem az n . karakterig veszi, hanem n darab karaktert vesz*/	string s = "vilá <u>eg</u> yetem", s2; s2 = s.Substring(5,3); /* s2 értéke egy lesz, s értéke nem változik*/
Substring(m) /*A szöveg m . karakterétől veszi a szöveget a végéig*/	string s = "vilá <u>eg</u> yetem"; string s2; s2 = s.Substring(5); /* s2 értéke egyetem lesz, s értéke nem változik*/
IndexOf(keresem) /*Ha a megadott szöveg (keresem) nem található -1 -et kapunk eredményül*/	string s = "kakadu"; int a; a = s.IndexOf("barack"); /*Az a változó értéke -1 lesz, mert a kakadu szöveg nem tartalmazza a barack szöveget*/
IndexOf(keresem) /*Ha a megadott szöveg (keresem) megtalálható akkor legelső előfordulásának pozícióját kapjuk eredményül*/	string s = "mama <u>ka</u> kadu"; int a; a = s.IndexOf("ka"); /*Az a változó értéke 4 lesz, mert a kakadu szöveg tartalmazza a ka szöveget a 4. pozíciótól kezdve (a 2. pozíciótól is tartalmazza, de mindig az első előfordulás pozícióját adja eredményül)*/
Remove(m) /*Törlés a szövegből az m . pozíciótól kezdve a végéig*/	string s = "mama <u>ka</u> kadu", s2; s2 = s.Remove(4); /* s2 értéke mama lesz, s értéke nem változik*/
Remove(m,n) /*Töröl a szövegből az m . pozíciótól kezdve n darab karaktert*/ /*Tehát nem az n . pozícióig törli, hanem n darabot töröl*/	string s = "mama <u>ka</u> kadu", s2; s2 = s.Remove(0,4); /* s2 értéke kakadu lesz, s értéke nem változik*/
Insert(m,szöveg) /*Beszúrja a szöveget az m . pozíciótól kezdve, az m . pozíciótól a szöveg jobbra tolódik*/	string duma = "manga <u>li</u> ca", duma2; duma2 = duma.Insert(5, "biga"); /* duma2 értéke mangabigalica lesz, duma értéke nem változik*/
Trim() /*A szöveg elejéről és végéről is törli a szóközöket*/	string duma = " mangalica ", duma2; duma2 = duma.Trim(); /* duma2 értéke "mangalica" lesz, duma értéke nem változik*/
TrimStart() /*A szöveg elejéről törli a szóközöket*/	string duma = " mangalica "; duma = duma.TrimStart(); /* duma értéke "mangalica " lesz*/
TrimEnd() /*A szöveg végéről törli a szóközöket*/	string duma = " mangalica ";

	duma = duma.TrimEnd(); /*duma értéke " mangalica" lesz*/
ToLower() /*Szöveg kisbetűssé alakítása*/	string duma = "kisMACSKa"; duma = duma.ToLower(); /*duma értéke kismacska lesz*/
ToUpper() /*Szöveg nagybetűssé alakítása*/	string duma = "nagyMacSKa"; duma = duma.ToUpper(); /*duma értéke NAGYMACSKA lesz*/
string m. karaktere [] operátor használatával	string s = "abcd"; char c; c = s[1]; /*c értéke b lesz*/;
Feltételek megadása	
Sokan eltévesztik: = jelenti az értékadást, == jelenti az összehasonlítást	
A legegyszerűbb feltétel két értéket hasonlít össze valamelyik relációs operátor használatával (ami nem is igaz, mert a legegyszerűbb feltétel a true és a false)	/*a és b legyen int típusú változó*/ true /*ez mindig igaz*/ false /*ez mindig hamis*/ 100 > 50 /*100 nagyobb, mint 50*/ a > 50 /*a nagyobb, mint 50*/ b < a /*b kisebb, mint a*/ 100 == b*a /*100 ugyanaz, mit b szorozva a-val*/ a != b /*a nem egyenlő b-vel*/ 200 - 20 >= a+b /*200-20 nagyobb vagy egyenlő, mint a és b összege*/ a <= b /*a értéke kisebb vagy egyenlő b értékénél*/
Ha több feltételt kell vizsgálnunk, akkor ezt még kombinálhatjuk a logikai operátorokkal	a == b a > 50 /*a-nak ugyanaz az értéke, mint b-nek, vagy a értéke nagyobb, mint 50*/ a != b && b < 20 /*a nem ugyanazt tartalmazza, mint b, és b kisebb, mint 20*/ !(a != b && b < 20) /*Az előző feltétel tagadása*/
Fontos, hogy az && operátornak nagyobb a precedenciája (amelyiknek nagyobb, az hajtódik végre először), mint a operátornak (lásd a *, /, +, - végrehajtási sorrendjét a matematikában, a *-nak ugyanaz a precedenciája, mint a /-nek, ezeknél kisebb a + és – precedenciája, a + és – precedenciája viszont ugyanaz) Ha el akarunk térni a precedenciától, akkor nyilván zárójelezhetünk.	
Matematikai utasítások	
Abs(szám) /*szám abszolút értéke*/	int a = -100; a = Math.Abs(a);

		/*a értéke 100 lesz*/
Sqrt(szám)	/*szám gyöke*/	double a = 100; a = Math.Sqrt(a); /*a értéke 10 lesz*/
Math.Pow(m,n)	/* hatványozás: m az n.-en*/	double a ; a = Math.Pow(2,10); /*a értéke 1024 lesz*/
Truncate(szám)	/*Szám törtrészének levágása, azaz a szám egészrészét kapjuk*/	double a = 500.978; a = Math.Truncate(a); /*a értéke 500 lesz*/
Round(szám,tizedes)	/*szám kerekítése megadot tizedesre*/	double a = 500.978; a = Math.Round(a,2); /*a értéke 500.98 lesz*/
Floor(szám)	/*A legnagyobb olyan egész számot adja, amely kisebb vagy egyenlő, mint a szám*/	double a = -500.978; a = Math.Floor(a); /*a értéke -501 lesz*/
Sin, Cos, Tan	/*Trigonometriai függvények*/	
Sign(szám)	/*A szám előjelét kapjuk (0, 1, -1)*/	int a = 0; Console.Write(Math.Sign(a)); /*0-át kapunk*/ a = 200; Console.Write(Math.Sign(a)); /*1-et kapunk*/ a = -200; Console.Write(Math.Sign(a)); /*-1-et kapunk*/
Feltétel használata logikai változó értékadásánál		
Logikai változó értékének beállítása		/*m legyen int típusú változó*/ bool a; a = true; /*a értéke igaz lesz*/ a = false; /*a értéke hamis lesz*/ a = 10 > m; /*az a változó értéke igaz lesz, ha 10 nagyobb, mint m, ellenkező esetben hamis*/
Véletlenszám előállítás		
int típusú véletlenszám előállítása		Random r = new Random(); int szam;

	<pre>szam = r.Next(); /*0-tól kb. kétmilliárdig kapunk számot*/</pre>
véletlenszám előállítása 0-tól, megadott felső értékig	<pre>Random r = new Random(); int szam; szam = r.Next(100); /*0-tól 99-ig kapunk számot*/</pre>
véletlenszám előállítása megadott intervalumban	<pre>Random r = new Random(); int szam; szam = r.Next(20,50); /*20-tól 49-ig kapunk számot*/</pre>
Feltételes elágazás if-else if-else szerkezettel	
<p>Ha a feltétel igaz hajtsdjon végre az utasítás, vagy utasítás blokk</p> <hr/> <pre>if (feltétel) utasítás;</pre> <hr/> <pre>if (feltétel) { utasítás; utasítás; }</pre>	<pre>Console.WriteLine("Kérem az első számot:"); int a = int.Parse(Console.ReadLine()); Console.WriteLine("Kérem a második számot:"); int b = int.Parse(Console.ReadLine()); if (a > b) Console.WriteLine("Az első szám nagyobb, mint a második"); /*Ha egy utasítás tartozik az if-hez, akkor nem kell kapcsos zárójel*/</pre> <hr/> <pre>if (a < b) { Console.ForegroundColor = ConsoleColor.Red; Console.WriteLine("Az első szám kisebb, mint a második"); } /*Ha több utasítás tartozik az if-hez, akkor kell a kapcsos zárójel*/</pre>
<p>Ha a feltétel igaz hajtsdjon végre az utasítás, vagy utasítás blokk, ellenkező esetben az else ág hajtsdjon végre</p> <pre>if (feltétel) utasítás; else utasítás;</pre> <hr/> <pre>if (feltétel) { utasítás; utasítás;</pre>	<pre>if (a == b) Console.WriteLine("A két szám egyforma."); else Console.WriteLine("A két szám nem egyforma.");</pre> <p>/*Ha a két szám nem egyforma, akkor az else ág hajtsdjon végre*/ /*Ha az egyes ágakhoz több utasítás tartozik, akkor nyilván ki kell tenni a kapcsos zárójeleket (egy utasításnál is kitehető)*/</p>

<pre> } else { utasítás; utasítás; } </pre>	
<p>Több különböző feltétel vizsgálata</p> <pre> if (feltétel) utasítás; else if (feltétel) utasítás; else if (feltétel) utasítás; else utasítás; </pre> <p>/*Ha egy feltétel teljesül és végrehajtottak a hozzá tartozó utasítások, akkor a többi ágot már meg sem nézi. Ha sem az if-nél sem az else if-nél megadott feltételek nem teljesülnek, akkor az else ág hajtódik végre (már ha van else ág, mert nem kötelező). Ha nincs else ág és egyik feltétel sem teljesül, akkor egyetlen utasítás sem fog végrehajtni*/</p>	<pre> if (a == 1) Console.WriteLine("one"); else if (a == 2) Console.WriteLine("dwa"); else if (a == 3) Console.WriteLine("trije"); else Console.WriteLine("llawer"); /*Elhagyható*/ </pre> <p>/*Pontosan egy ág utasításai fognak végrehajtni*/ /*Ha nincs else ág, akkor lehet, hogy egy se (pl. ha a értéke 4)*/</p>
<p>Mikor kell else if és mikor nem?</p>	<pre> if (a == 1) Console.WriteLine("one"); if (a == 2) Console.WriteLine("dwa"); if (a == 3) Console.WriteLine("trije"); </pre> <p>/*Itt is nyilván egy ág utasításai fognak csak végrehajtni (mert a feltételek egymást kizáróak) (vagy egy se, mert nincs else). Viszont minden feltételt meg fog vizsgálni, mert egymás után szépen végrehajtja az összes if-et. Ha a értéke 1, akkor kiírja, hogy one, de utána még megnézi, hogy 2 vagy 3 van-e benne. Ez nyilván felesleges, mert ha 1 van benne, akkor már nem lehet 2. Ebben az esetben nyilván az előző megoldást kell használni (else if)*/</p>

	<p>Egymást nem kizáró feltételek esetén viszont hibás lehet ha else if-et használunk.</p> <pre> if (a == 10) Console.Write("a értéke 10"); else if (b == 20) Console.Write ("b értéke 20"); /*Ha a értéke 10, akkor megtudjuk, hog mi van az a-ban, de hogy ha a b-ben 20 van már nem írja ki. Ha így akartuk, akkor jó. De ha kíváncsiak vagyunk arra is, hogy b-ben 20 van-e, akkor nem jó.*/ Ha viszont így használjuk: if (a == 10) Console.Write("a értéke 10"); if (b == 20) Console.Write ("b értéke 20"); /*Ha a-ban 10 van, akkor kiírja ezt az információ. Ha b-ben 20 van, akkor kiírja ezt az információt is, mert most nem else if-et használtunk, hanem két egymástól független if-et, tehát a második if akkor is kiértékelődik, ha az első if feltétele true.*/ </pre>
<p>Feltételes elágazás switch szerkezettel</p> <pre> switch (érték) { case konstans érték: utasítások; break; case konstans érték: utasítások; break; case konstans érték: utasítások; break; default: </pre>	<pre> int a = int.Parse(Console.ReadLine()); switch(a) { case 1: Console.Write("Egy"); break; case 2: Console.Write("Kettő"); break; default: Console.Write("Eddig már nem tudok számolni"); break; </pre>

<pre> utasítások; break; } </pre>	<pre> } </pre> <p>A switch után levő értéket hasonlítja a case utasítások után megadott konstans értékhez. Ha van egyezés, akkor azt az ágot végrehajtja, majd a break utasítás hatására kiugrik a végrehajtás a switch-ből. Ha egyik feltétel sem teljesül a case ágaknál, akkor a default ág hajtódik végre (már ha van, mert nem kötelező). A konstans érték azt jelenti, hogy nem lehet benne változó (tehát már a program futtatása előtt is egyértelmű ez az érték).</p>
Stringek összehasonlítása	
Ha egyenlőséget vizsgálunk nincs semmi probléma	<pre> string s = "eper", s2 = "teper"; if (s == s2) Console.WriteLine("Ugyanaz van bennük"); </pre>
ABC sorrendet viszont a CompareTo utasítással kell	<pre> string s = "eper", s2 = "teper"; int a = s.CompareTo(s2); if (a == -1) Console.WriteLine("s előrébb van az ABC sorrendben"); else if (a == 0) Console.WriteLine("Egyformák"); else /*ekkor a-ban 1 van*/ Console.WriteLine("s2 előrébb van az ABC sorrendben"); </pre>
Ciklusok	
<p>Ciklusokat akkor használunk ha bizonyos utasítást vagy utasításokat többször is végre akarunk hajtani. Ezt egy feltétel megadásával valósítjuk meg. A ciklusok addig működnek, amíg a feltétel igaz. Ha a feltétel hamis lesz véget ér a ciklus.</p>	
for ciklus	
<pre> for (inicializáció ; feltétel ; léptetés) { ciklusmag; } </pre> <p>Az inicializáció és a léptetés el is hagyható (viszont mindig kell a két pontosvessző). Az inicializációs rész és a léptetés rész egy utasítást vagy több vesszővel elválasztott utasítást tartalmazhat (ha nincs elhagyva).</p> <ol style="list-style-type: none"> Először az inicializációs rész hajtódik végre (és csak egyszer). Ezután a b pontot hajtja végre. Megnézi, hogy a feltétel igaz-e. Ha igaz, akkor végrehajtja a c pontot, ha hamis, akkor vége a ciklusnak. 	<pre> for (int i = 1; i <= 10; i++) Console.WriteLine("{0}, " , i); /*1-10-ig kiírja a számokat vesszővel elválasztva*/ /*A for cikluson belül deklarált változók csak a for cikluson belül használhatóak.*/ int a = 10 , b = 3; for (; a > b ;) a -= b; /*Csak a feltételt adtuk meg.*/ </pre>

<p>c. Végrehajtja a ciklusmagot, majd végrehajtja a d pontot.</p> <p>d. Végrehajtja a léptetést, majd újra a b pont jön.</p>	<pre>int a , b; for (a = 10, b = 3 ; a > b ; a++ , b++) a -= b; Console.Write (a + " " + b); /*Most a léptetés és az inicializációs rész is két vesszővel elválasztott utasítást tartalmaz*/ /*Nyomkövesse a program működését a gépén*/ int a; for (a = 100 ; a > 0 ; a--); Console.Write (a); /*A ciklusmag is elhagyható. Pontosabban a ciklusmag most az üres utasítás, ami nem csinál semmit.*/</pre>
while ciklus	
<pre>while (feltétel) { ciklusmag; }</pre> <p>Amíg a feltétel igaz végrehajtodik a ciklusmag.</p> <p>a. Megnézi, hogy igaz-e a feltétel. Ha igaz végrehajtja a b pontot, ha hamis, akkor vége a ciklusnak.</p> <p>b. Végrehajtja a ciklusmagot. Majd végrehajtja az a pontot.</p>	<pre>int a = 1; while (a <= 100) { Console.Write (a + ", "); a++; }</pre> <p>/*1-100-ig kiírja a számokat*/</p>
do while ciklus	
Ez egy hátultesztelő ciklus. A ciklusmag a feltétel előtt hajtodik végre. (Az előző ciklusok előltesztelőek voltak)	
<pre>do { ciklusmag; } while (feltétel);</pre> <p>a. Végrehajtodik a ciklusmag, majd végrehajtodik a b pont.</p> <p>b. Ha a feltétel igaz az a pont hajtodik végre, ellenkező esetben vége a ciklusnak.</p>	<pre>int szam; do { Console.Write ("Kérek egy számot 1-10 között:"); szam = int.Parse (Console.ReadLine()); } while (szam < 1 szam > 10);</pre>

Kisokos

	/*Nem engedjük a felhasználónak, hogy más számot írjon be, ha nem jó újra és újra bekérjük tőle.*/