

Programozás Logo nyelven

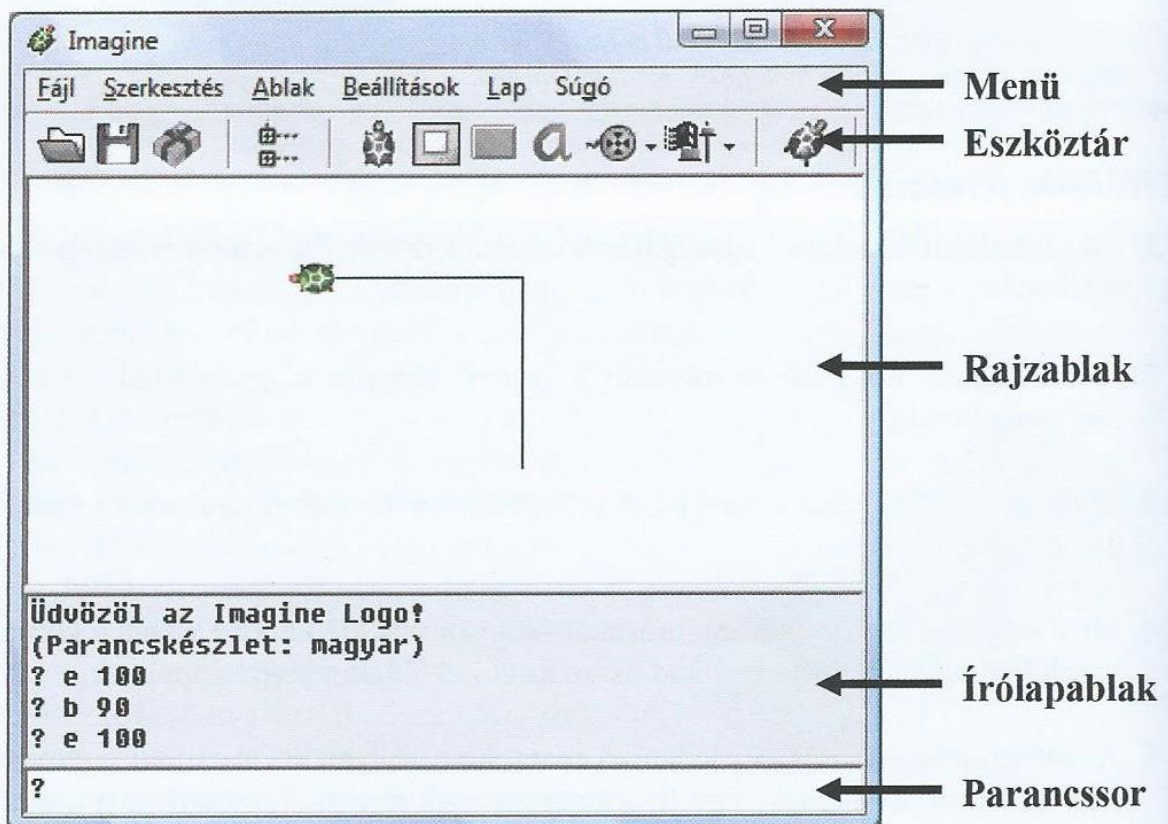
Programot sokan már az alsó tagozatban is készítettek, amikor megismerkedtek a teknőccel. Ebben a fejezetben először átismétljük ezeket az ismereteket, majd tovább bővítjük olyan új fogalmakkal, amelyekre a későbbiekben szükségünk lesz.

Teknőcgrafika

Az Imagine fejlesztői környezet

Az olyan programot, amellyel új programokat készíthetünk, *fejlesztői környezetnek* nevezik. Egy ilyen fejlesztői környezet például az Imagine (ejtsd: imedzsín), ahol egy teknőc segíti munkánkat.

Az Imagine indításakor a **teknőc** a rajzablak közepén áll, felfelé néz, és egy vékony fekete tollat tart. A teknőcöt az **írólapablak** alsó sorába, a **parancssorba** írt utasításokkal irányíthatjuk, míg ő a **rajzablakban** mozog. A korábban kiadott utasításokat az írólapablakban követhetjük nyomon. Az ablakokat az F5, F6, F7 funkcióbillentyűkkel tüntethetjük el, illetve jeleníthetjük meg.



Az Imagine Logo képernyője. A teknőc éppen egy négyzetet rajzol. Hogyan kell folytatni az utasításokat, hogy a teknőc befejezze a négyzetet?

A Logo programozási nyelv

A fejlesztői környezetbe speciális nyelven, az ún. *programozási nyelven* írhatjuk be az utasításokat. Az Imagine teknőce például a Logo nyelvet érti.

A későbbiek során más fejlesztői környezetekkel és más programozási nyelvekkel is meg fogunk ismerkedni.

A Logo programozási nyelv számos utasításával már találkozhattunk. Ezek főleg a teknőc irányítására szolgálnak: előre, hátra, balra, jobbra, tollat fel, tollat le. Cserélhetjük továbbá a toll színét és vastagságát, illetve akár törölhetjük a képernyőt is.

A Logóban kényelmes segítség, hogy az utasítások neve helyett azok rövidítését is használhatjuk. Például az `előre 100` helyett elég ennyit írunk: `e 100`.

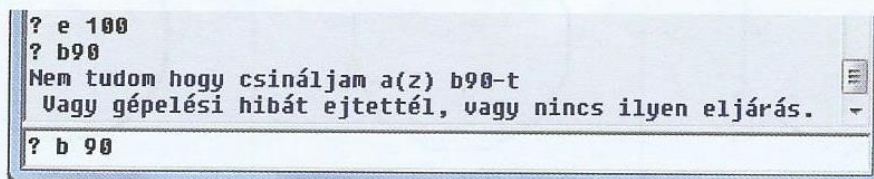
A leggyakoribb Logo utasításokat a következő táblázat foglalja össze:

Utasítás	Rövidítve	Hatása
<code>előre 100</code>	<code>e 100</code>	A teknőc 100 képpontot előrelép.
<code>hátra 50</code>	<code>h 50</code>	A teknőc 50 képpontot hátralép.
<code>balra 90</code>	<code>b 90</code>	A teknőc 90 fokkal balra fordul.
<code>jobbra 60</code>	<code>j 60</code>	A teknőc 60 fokkal jobbra fordul.
<code>tollatfel</code>	<code>tf</code>	A teknőc felemeli a tollát.
<code>tollatle</code>	<code>tl</code>	A teknőc leteszi a tollát.
<code>törölképernyő</code>	<code>törölkép</code>	Törli a képernyőt, és a teknőc alapállapotba áll.
<code>tollszín! "kék</code>	<code>tsz! "kék</code>	A teknőc kék színű tollat vesz elő.
<code>tollvastagság! 10</code>	<code>tv! 10</code>	A teknőc 10 képpont vastagságú tollat vesz elő.

Az adatok kiválasztását rábízhatjuk akár a teknőcre is, ha a megfelelő érték helyett a **tetszőleges** vagy rövidítve **tetsz** szót írjuk. Így például a teknőc tolla a `tsz! tetsz` utasítás kiadásával véletlenszerűen bármilyen színű lehet.

Hibák javítása

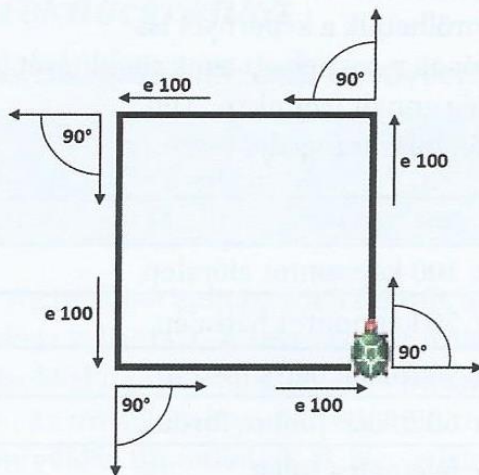
Aki dolgozik, az hibázik is: az utasításokat elírhatjuk. Gyakori hiba például, hogy elírjuk az utasítást, vagy kimarad a szóköz: `b90`. Ilyenkor a fejlesztői környezet hibaüzenetet küld, amit feltétlenül el kell olvasni. Segít azonban a hiba kijavításában az is, hogy a korábban már beírt parancsokat a nyilas gombok nyomkodásával újra megjeleníthetjük, és módosíthatjuk.



Az utasítást hibásan gépeltük be: kimaradt a szóköz. A fejlesztői környezet figyelmeztet a hibára. A nyilas gombokkal újra megjeleníthetjük és javíthatjuk a hibás utasítást. Próbáld ki!

Utasítások ismétlése. Kör

Amikor valamilyen „szabályos” alakzatot rajzolunk, a teknőcnek gyakran ugyanazokat az utasításokat kell újra és újra kiadni. Például egy négyzet rajzolásakor a teknőc négyszer előre megy, és elfordul 90 fokkal. Ilyen esetekben alkalmazhatjuk az **ismétlés**, vagy rövidítve **ism** utasítást. Az **ism** kulcsszót a darabszám, majd szögletes zárójelben a végrehajtandó utasítások listája követi. Ismétléssel minden szabályos sokszöget könnyen megrajzolhatunk.



Lépésenként:

```
e 100
b 90
e 100
b 90
e 100
b 90
e 100
b 90
```

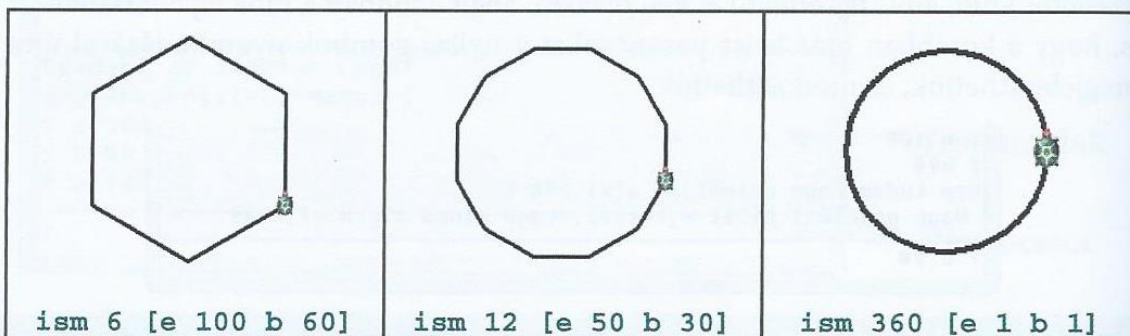
Röviden:

```
ism 4 [e 100 b 90]
```

Négyzet rajzolása lépésenként és ismétléssel. A nyíl a teknőc irányát mutatja. Hogyan kell ezek alapján szabályos nyolcszöget rajzolni?

Vajon hány fokkal kell a teknőcnek elfordulni minden lépésben, ha szabályos hatszöget szeretnénk rajzolni? Mivel ahhoz, hogy a teknőc körbeérjen, összesen 360° -ot kell megtennie, így a 6 fordulóra egyenként $360^\circ : 6 = 60^\circ$ jut. Hasonló módon, a 12 oldalú szabályos sokszög esetén egy-egy fordulóra $360^\circ : 12 = 30^\circ$ jut, 360 oldalú esetén pedig már csak 1 fok. Ezt akár már körnek is vehetjük.

Kört természetesen máshogy is rajzolhatunk. A **kör 100** utasítás kiadásával a teknőc egy 100 sugarú kört rajzol maga köré. A **pontméret 100** utasítás hatására pedig egy 100 sugarú, az éppen használt toll színével kitöltött óriási pontot kapunk. (Ugyanezt eredményezi a **töltöttkör 100** utasítás is.)



Vajon hogyan lehetne bemutatni, hogy a 360 oldalú szabályos sokszög ténylegesen nem kör?

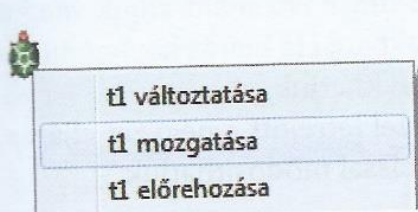
Új ablak, új teknőc

Érdeemes megismerkednünk még a következő lehetőségekkel is.

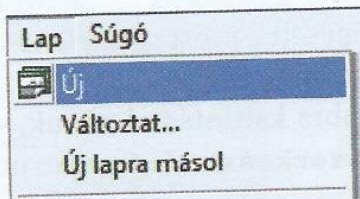
A teknőcot az egérrel is odébb tehetjük, ha a teknőcre jobb egérgombbal kattintva a helyi menüben kiválasztjuk a *teknőc (t1) mozgatása* menüpontot.

Ha több ábrát szeretnénk készíteni, ezeket külön lapra is rajzolhatjuk. Új lapot a *Lap* menü *Új* pontjával kérhetünk, a lapok között pedig a *Lap* menü alján válthatunk.

Ha új lapot kérünk, arra nem kerül rá automatikusan a teknőc, ezt az eszköztár *új teknőc* ikonjával kell megtennünk.



A teknőc áthelyezése egérrel.



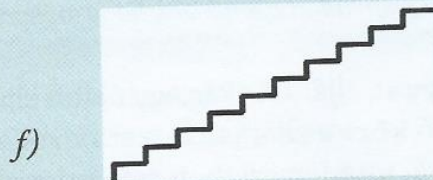
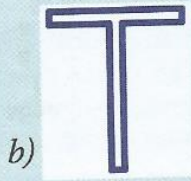
Új rajzablak kérése.



Új teknőc beillesztése.

Kérdések, feladatok

1. Készítsd el a következő ábrákat!



2. A *Fájl* menü *Demoválasztás* pontjával kész programok között válogathatunk. Indítsd el például a *bekak* nevű alkalmazást!



Eljárás, program, projekt

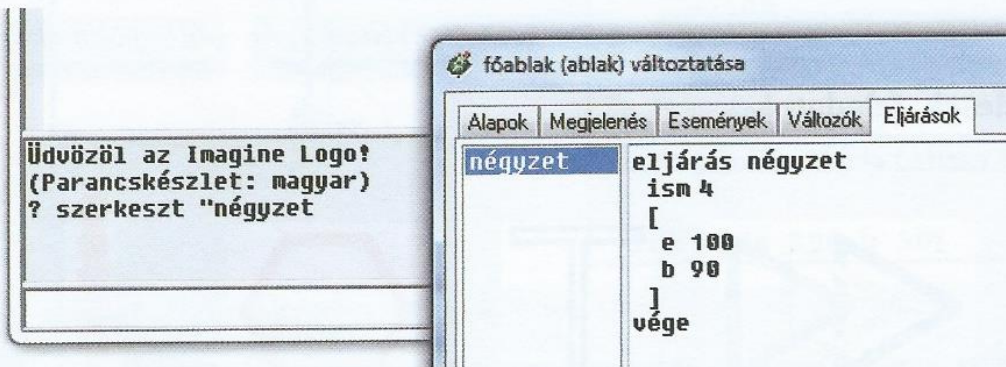
Kör rajzolására több eszközünk van, például rajzolhatunk ismétléssel egy 360 oldalú sokszöget, de akár közvetlenül kiadhatjuk a kör utasítást is.

Négyzet készítésére ugyan nincs külön utasítás, de a Logo lehetővé teszi, hogy készítsünk. Nézzük meg hogyan!

Példa:Négyzet

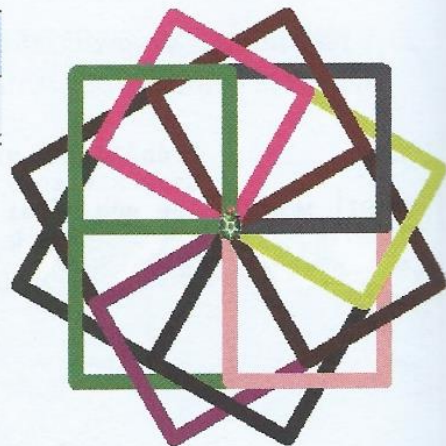
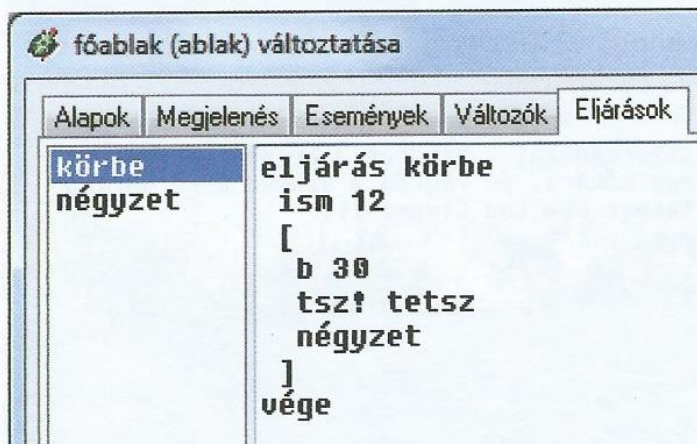
Írjuk be a parancssorba a **szerkeszt** "négyzet" utasítást, majd adjuk meg a négyzet elkészítésének lépéseit a most megjelenő ablakban! Ha beírás közben hibázunk, a hibát a **Jegyzetömbben** megszokott módon javíthatjuk.

Munkánkat az OK gombra kattintással zárjuk, s ezzel létrejött a négyzet eljárás. A kész eljárást később a **szerkeszt** "négyzet" utasítással módosíthatjuk.



Hozd létre az ábrán látható módon a négyzet eljárást!
Hogyan tudod ellenőrizni, hogy működik-e?

A továbbiakban a **négyzet** eljárást akár egy újabb eljárásban is felhasználhatjuk. A következő ábrán látható **körbe** eljárás 12-szer rajzolja ki a négyzetet, egymáshoz képest 30 fokkal elforgatva, minden esetben más színnel:



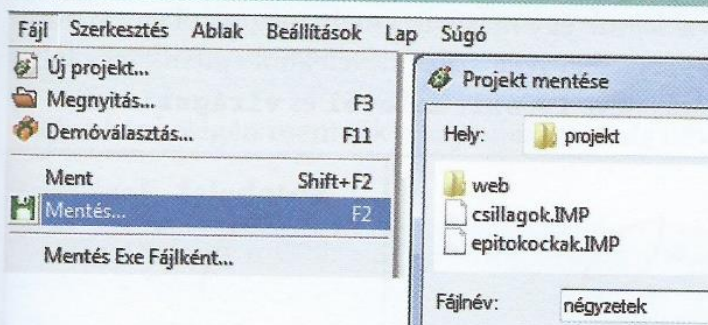
Négyzetek körbe rendezése.
Hogyan változik a kép, ha 12 helyett 120, és 30 helyett 3 szerepel?

A **program** a számítógépnek szóló utasítások sorozata, amely meghatározza, hogy a számítógép hogyan végezzen el egy adott feladatot. Egy Logo nyelvű program eljárásokra tagolódik. Példánkban a négyzetek körbe rendezését a **négyzet** és a **körbe** eljárások segítségével valósítottuk meg.

Az **eljárás** egy részfeladatot megvalósító művelet sor, melyet névvel ellátva tárolunk. Az eljáráshoz tartozó művelet sor később az eljárás nevének beírásával indítható. A **négyzet** eljárás például egy négyzetet rajzol meg.

A **projekt** egy program elkészítéséhez szükséges fájl vagy fájlok összessége. Az Imagine Logóban a projekt egyetlen fájlból áll.

A projektet a **Fájl** menü **Mentés** pontjával menthetjük el, illetve **Megnyitás** pontjával vehetjük elő. A nevét mentéskor kell megadnunk.



Projekt mentése a **Fájl** menü **Mentés** pontjával. Mentse el a munkádat négyzetek néven, és lép ki az Imagine-ből. Hogyan tudnád a projektet folytatni?



Az **F4** gomb lenyomására a képernyő jobb szélén megjelenik az **Intéző**, a projekt részeivel.

Kérdések, feladatok

- Bővítsd a projektet az **egykör** eljárással, ez rajzoljon egy 360 oldalú szabályos sokszöget! Módosítsd a **körbe** eljárást úgy, hogy a négyzet helyett a most készített **egykör** eljárást hívja meg! A teknőc most az ábrához hasonló rajzot készít. Mentse el az új projektet **körbékör** néven!



- A következő sor háromszögekből áll. Készítsd el a projektet!



- Ismétlés segítségével készítsd el a következő „pöttysort”!



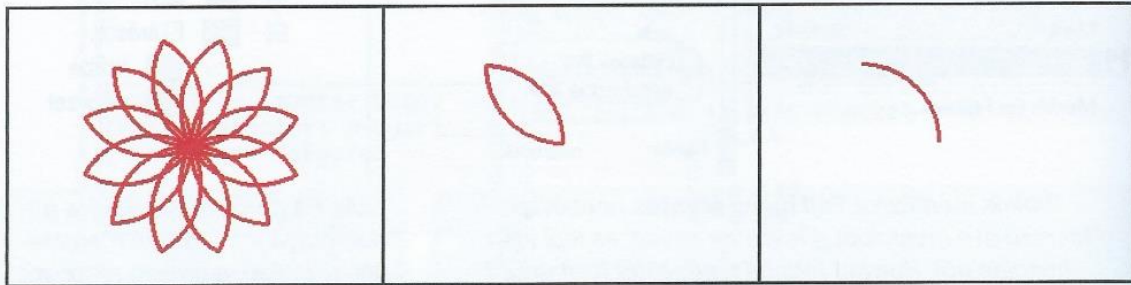
Programunkat tervezni kell!

Ha egy bonyolult ábrát kell megrajzolnunk, rossz megoldás csak úgy utasításokat írogatni az írólapablakba, aztán majd kialakul valami. Ilyenkor a feladat megoldása az ábra elemzésével, és a feladat részekre bontásával kezdődik.

Példa: Virágszirom rajzolása

Az ábrán látható virágszirom megrajzolása közben a teknőc elég bonyolult mozgást végez. Vizsgáljuk meg, hogyan is épül fel a teknőc mozgása! A virágszirom 12 szíromlevélből áll, az egyes szíromlevelek pedig 2-2 ívből.

Adódik a következő megoldás. Első lépésként készítsünk egy eljárást, amely megrajzol egy ívet. Ennek felhasználásával írjunk egy másikat, amely megrajzol egy szíromlevelet. Végül egy harmadik eljárás segítségével a szíromlevelekből építsük fel a teljes virágsziromot. Legyen a három eljárás neve **ív**, **szíromlevél** és **virágszirom**!

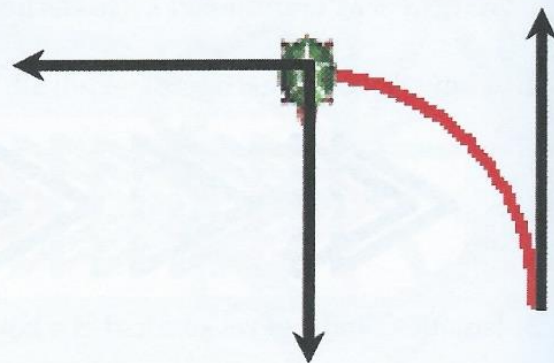


A virágszirom szíromlevelekből, a szíromlevelek két-két ívből állnak.
Hogyan lehet az ívekből a virágot felépíteni?

Az ívek negyedkörök. A korábban már megismert ismétléses eljárással ezért az ívet könnyen megrajzolhatjuk, ha a teknőc 360 helyett most csak 90-et lép.

Az első ív megrajzolása után a teknőcnek 90 fokot kell elfordulnia, hogy a második ívhez jó irányban álljon. Ezt követően újra el kell 90 fokot fordulnia, hogy visszatérjen a kiindulási helyzetébe.

főablak (ablak) változtatása			
Alapok	Megjelenés	Események	Változók
ív		eljárás ív	
szíromlevél		ism 90	
virágszirom		[
		e 1	
		b 1	
]	
		vége	



Egy-egy ív során a teknőc 90 fokot fordul. Ahhoz, hogy a szíromlevél megrajzolása során körbeérjen, az ívek után még 90-90 fokkal kell elfordulnia. Vajon miért kell erre ügyelnünk?

Programozás Logo nyelven

Végül a teljes virágszirom 12 szíromlevélből áll. Ezek mindegyike után 30 fokot kell a teknőcnek elfordulnia, hiszen $12 \times 30 = 360$ fok.

```
főablak (ablak) változtatása
Alapok  Megjelenés  Események  Változók  Eljárások
ív
szíromlevél
virágszirom
eljárás szíromlevél
ív
b 90
ív
b 90
vége
```

```
főablak (ablak) változtatása
Alapok  Megjelenés  Események  Változók  Eljárások
ív
szíromlevél
virágszirom
eljárás virágszirom
ism 12
[
szíromlevél
j 30
]
vége
```

A szíromlevél és a virágszirom eljárás.

Hogyan kell az eljárásokat módosítani, ha nem 12, hanem 18 levelű szíromot szeretnénk?

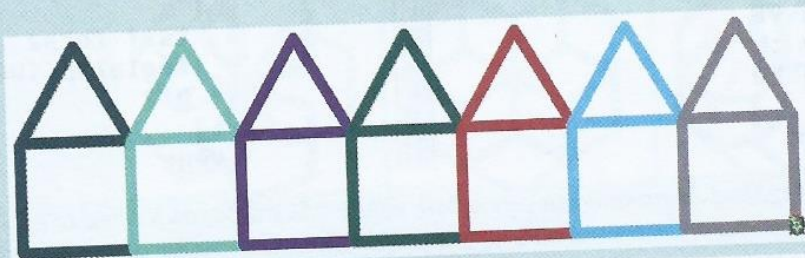
Munkánk végén mentjük el a projektet *virág* néven!

Kérdések, feladatok

1. Fejleszd úgy tovább a programot, hogy a virágnak szára és levele is legyen!
2. A projekt módosításával és kiegészítésével készíts tetszőlegesen szétszórt virágokból álló kertet!



3. A következő ábra egy házsort mutat. Hogyan lehetne elemeire bontani? Készítsd el a projektet! Ügyelj arra, hogy a házsor a rajzablak bal oldalán kezdődjön!



Paraméteres eljárás készítése

Példa: Négyzet, téglalap

A **kör** beépített utasítás mintájára nemrégiben elkészítettük a **négyzet** eljárást. Ennek azonban nagy hiányossága, hogy amíg a kör esetében megadható annak sugara, addig a négyzet eljárásunk mindig 100 pont oldalú négyzetet rajzolt.

Sok olyan eljárás van a Logóban, amelyek indításakor a teknőcnek további adatokra is szüksége van. Ilyen például az **előre** vagy a **kör** eljárás, ahol meg kell adnunk, hogy a teknőc mennyit menjen előre, illetve mekkora sugarú kört rajzoljon. Ezeket az adatokat az eljárás **paraméterének** nevezzük.

Paraméteres eljárást magunk is készíthetünk. Ilyenkor a paraméter értékét egy betűvel vagy szóval helyettesítjük. A paraméterek mindig kettősponttal kezdődnek. A paramétereket az eljárás nevében is meg kell adnunk.

The screenshot shows the 'főablak (ablak) változtatása' window with tabs for 'Alapok', 'Megjelenés', 'Események', and 'Változók'. The 'négyzet' procedure is defined as follows:

```

eljárás négyzet :a
  ism 4
  [
    e :a
    b 90
  ]
vége
    
```

To the right, a list of execution results is shown:

```

? négyzet 20
? négyzet 40
? négyzet 60
? négyzet 80
? négyzet 100
? négyzet 120
? négyzet 140
    
```

On the far right, a small window displays a series of nested squares, with the largest one having a side length of 140.

Példánkban a négyzet oldalának hosszát az :a paraméterrel jelöltük. Mit történik, ha a négyzet eljárás indításakor elfelejtjük beírni a paramétert?

Egy eljárásnak akár több paramétere is lehet. Ilyenkor a paramétereket abban a sorrendben kell megadnunk, amilyen sorrendben azok az eljárás neve után fel vannak sorolva. Például a téglalap nevű eljárás két paramétere a téglalap két oldala.

The left screenshot shows the 'főablak (ablak) változtatása' window with tabs for 'Alapok', 'Megjelenés', 'Események', 'Változók', and 'Eljárások'. The 'téglalap' procedure is defined as follows:

```

eljárás téglalap :a :b
  ism 2
  [
    e :a
    b 90
    e :b
    b 90
  ]
vége
    
```

The right screenshot shows the same window with the 'forgó' procedure defined, which uses the 'téglalap' procedure:

```

eljárás forgó
  tv! 10
  ism 120
  [
    tsz! tetsz
    téglalap 100 200
    b 3
  ]
vége
    
```

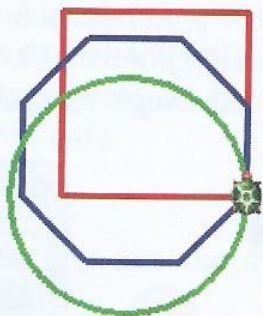
A téglalap eljárás elkészítése és meghívása egy másik eljárásból. Írd be, és indítsd el a projektet!

Sokszögek készítése

A négyzet mintájára könnyen készíthetünk olyan eljárást, amely a paraméterében megadott oldalszámú szabályos sokszöget rajzol. Legyen az eljárás neve **nszög**, a paraméter neve pedig **:n**.

Mivel a sokszögnek n oldala van, ezért a teknőcnek $:n$ -szer kell előremennie, és elfordulnia. Ha a sokszög kerülete pl. 400, akkor minden oldala $400/:n$ hosszú, ez lesz tehát az előre eljárás paramétere. Mivel a teljes szög 360 fok, ezért minden oldal után $360/:n$ fokkal kell elfordulnia balra.

főablak (ablak) változtatása	
Alapok	Megjelenés
nszög	<pre> eljárás nszög :n ism :n [e 400/:n b 360/:n] vége </pre>



```

? tsz! "piros nszög 4
? tsz! "kék nszög 8
? tsz! "zöld nszög 360
                    
```

Négyzet, nyolcszög és „kör” rajzolása az nszög eljárással.
Vajon a három alakzat közül melyiknek a legnagyobb a kerülete? Miért?

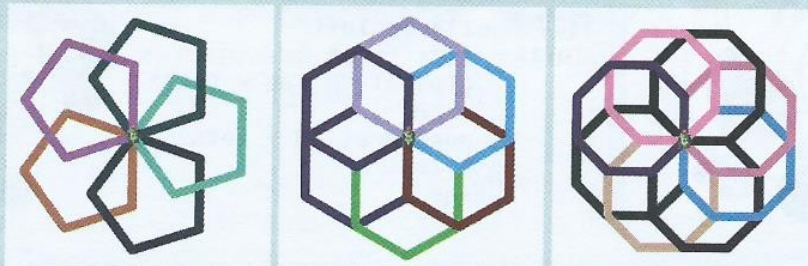
Kérdések, feladatok

1. A következő eljárások közül melyek paraméteresek? Mit jelent esetükben a paraméter?
előre, tollatfel, jobbra, törölkép, tollatle, pontméret, ismétlés

2. Készíts eljárást, amely kirajzol egy *Elsőbbségadás kötelező* táblát! A tábla nyele legyen kétszer olyan hosszú, mint a tábla oldala! Készítsd el az eljárást paraméteresen is úgy, hogy indításkor meg kelljen adni a tábla oldalát!



3. A következő ábrákon a teknőc 5 db ötszöget, 6 db hatszöget, illetve 8 db nyolcszöget rajzolt meg, a toll színét minden sokszög után módosította. Készíts egy olyan paraméteres eljárást, amely az **nszög** eljárást felhasználva képes megrajzolni mindhárom ábrát!

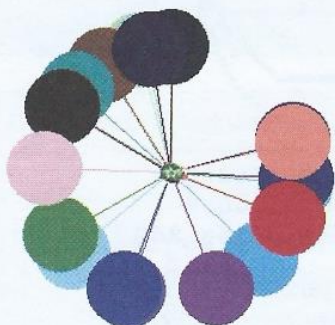


Változók használata

A paraméter egy adott eljáráshoz tartozó adat. Szükségünk lehet azonban az eljárásokhoz szorosan nem kötődő adatok tárolására is. Ezt valósíthatjuk meg a változók segítségével.

Példa: Lufik

A következő példában a **lufi** eljárás egy tetszőleges színű, 60 pont méretű lufit rajzol, amely egy 100 pont hosszú madzag végén van. A lufi megrajzolása után a teknőc visszatér a kiindulási pontba. A **lufik** eljárás pedig ezt hívja meg 20-szor, előtte azonban a teknőcot tetszőleges szöggel elfordítja. Így egy maréknyi egyforma méretű lufit kapunk.



Alapok	Megjelenés	Események
lufi	eljárás lufi	
lufik	tsz? tetsz	
	e 100	
	pontméret 60	
	h 100	
	vége	

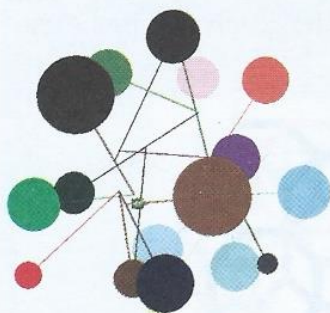
Alapok	Megjelenés	Események
lufi	eljárás lufik	
lufik	törölkép	
	ism 20	
	[
	b tetsz	
	lufi	
]	
	vége	

Készítsd el a „lufirajzó” programot! Hogyan lehetne úgy átírni, hogy a lufik mérete és a madzag hossza mindig tetszőleges legyen?

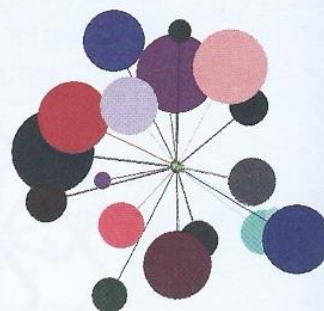
Hogyan tudnánk elérni, hogy a lufik mérete változzon? Ha a lufi méretének megadására a **pontméret 20 + tetsz** utasítást használjuk, akkor a lufik mérete tetszőleges, de legalább 20 pont nagyságú lesz. *Próbáld ki!*

Vajon el tudjuk-e ugyanígy érni azt is, hogy a madzag hossza tetszőleges legyen? Írjuk az **e 100** helyett **e 100 + tetsz**, a **h 100** helyett pedig **h 100 + tetsz** utasítást! Az eredményt az ábrán látjuk, az bizony nem egy „luficsokor”.

A hiba oka nyilván az, hogy a **tetsz** értéke más lesz, amikor a teknőc előre megy, és más, amikor hátra. Vajon hogyan tudnánk megoldani, hogy a **100 + tetsz** értéket a teknőc megjegyezze, és mindkét esetben ezt használja?



Alapok	Megjelenés	Események	Változók
lufi	eljárás lufi		
lufik	tsz? tetsz		
	globvál "m 100 + tetsz		
	e :m		
	pontméret 20 + tetsz		
	h :m		
	vége		



A **100 + tetsz** értéket az **m** változó tartalmazza. Előre és hátra így ugyannyit megy a teknőc.

A programozási nyelvekben az adatokat *változók* segítségével tárolhatjuk a memóriában. A változókat a nevük azonosítja.

A Logo nyelvben például a globvál `"m 100` létrehoz egy `m` nevű változót, melynek értéke 100. Az `e :m` utasítás hatására pedig a teknőc annyit lép előre, amennyi az `m` változó értéke.

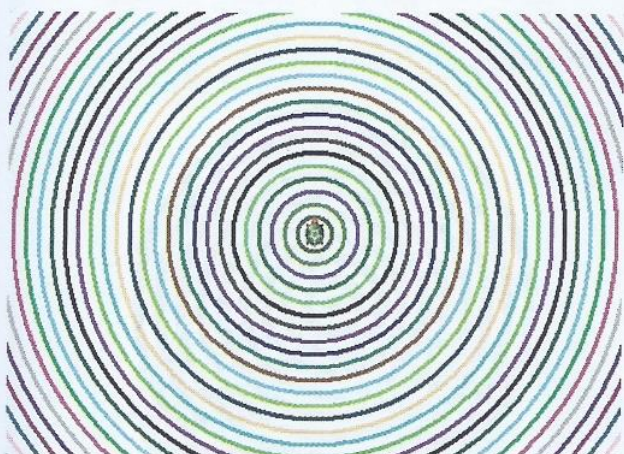
Figyeljünk arra, hogy amikor a változó értékét módosítjuk, akkor a neve elé tegyünk egy macskakörmet, amikor pedig lekérdezzük, akkor egy kettőspontot!

Ezek után már érthető az ábrán látható eljárás. A globvál `"m 100 + tetsz` sorban létrehozuk az `m` nevű változót, melynek értéke `100+tetsz`. Az `m` változó értéke mindaddig nem változik, amíg új értéket nem adunk neki. Így a teknőc a 4. és a 6. sorban ugyanannyi lépést tesz meg előre, illetve hátra.

Körök

A változó értékét az eljárásban akár módosíthatjuk is. A következő példában koncentrikus köröket rajzolunk, egyre nagyobb sugárral.

A kör sugarát az `r` változó tartalmazza. Az `r` értéke kezdetben 10, majd minden kör megrajzolása előtt 20-szal nő.



```
főablak (ablak) változtatása
Alapok  Megjelenés  Események  Változók
körök   eljárás körök
globvál "r 10
ism 100
[
globvál "r :r+20
tst? tetsz
kör :r
]
vége
```

A kör sugarát az `r` változó tartalmazza, melynek értékét minden kör megrajzolása előtt növeljük. Mit jelent a macskaköröm, illetve a kettőspont a globvál `"r :r+20` sorban?

Kérdések, feladatok

1. Készíts programot, amely kirajzol egy cseresznyét a képernyőre! Módosítsd úgy a programot, hogy a cseresznyék szára véletlen hosszú, de legalább 50 pont legyen! Egészítsd ki egy újabb eljárással a programot úgy, hogy az szórja tele a képernyőt cseresznyékkel!
2. Készíts körsort úgy, hogy a körök az ábrán látható módon, egy pontban találkozzanak!
3. Miben tud „többet” a változó, mint a paraméter?



Vegyes feladatok

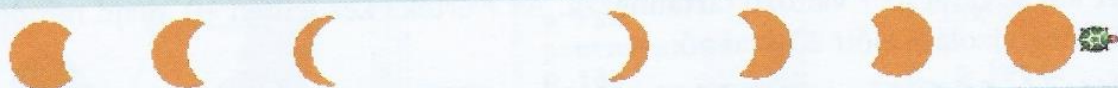
1. Hányféleképpen tudunk kört rajzolni?

2. Hogyan készülhetett az ábrán látható „láncszem”? Készíts többféle megoldást is!

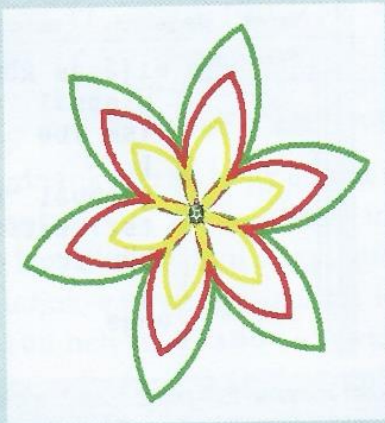
3. Készíts **lépcső** néven paraméteres eljárást! A paraméter a lépcsőfokok számát adja meg!



4. Készíts paraméteres eljárást **pontsor** néven! Az eljárás rajzoljon ki 8 darab 40 méretű pontot! Az eljárás paramétere a pontok távolsága legyen! A következő ábra a Hold változásait mutatja. Hogyan készülhetett? (Amikor a Hold C alakú, akkor „csökken”, amikor D alakú, akkor „dagad”. Amikor nem látszik, akkor beszélünk újholdról, amikor pedig teljes méretű, akkor van telihold. Nézz utána az interneten, hogy mi a Hold fázisváltozásainak az oka!)

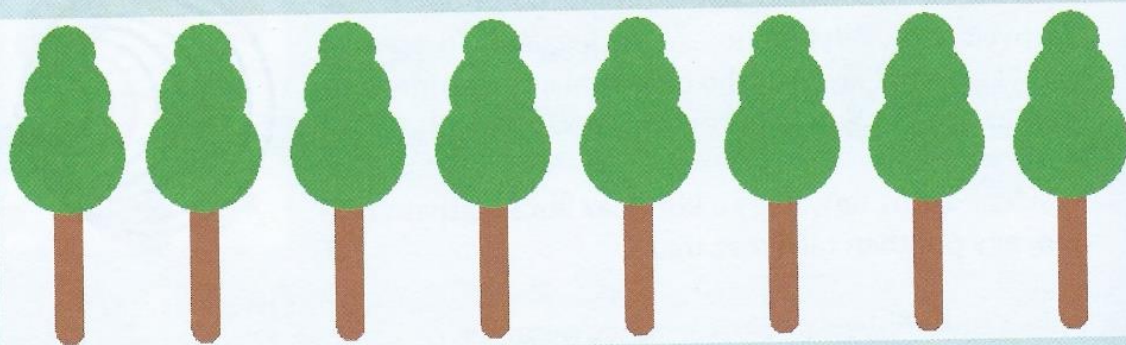


5. Rajzold meg az ábrán látható színes virágot!

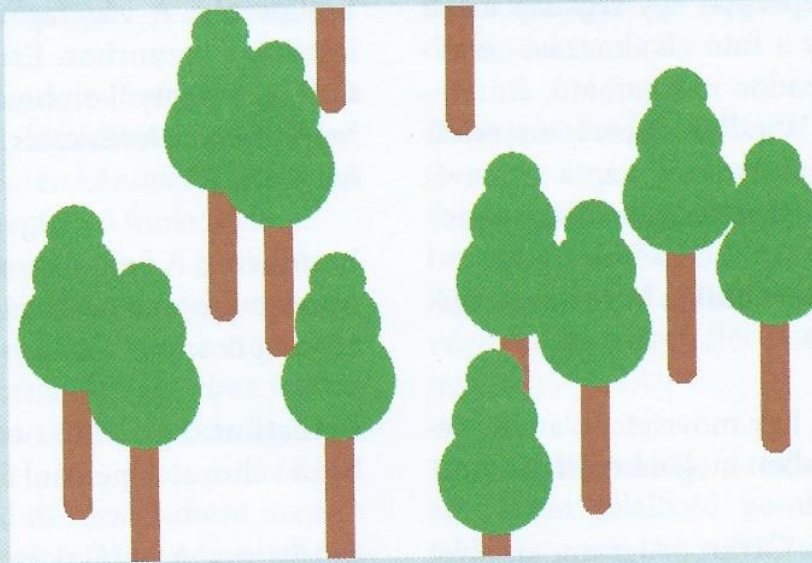


6. Készítsd el a „céltablát” ismétlés alkalmazása nélkül, majd ismétléssel is!

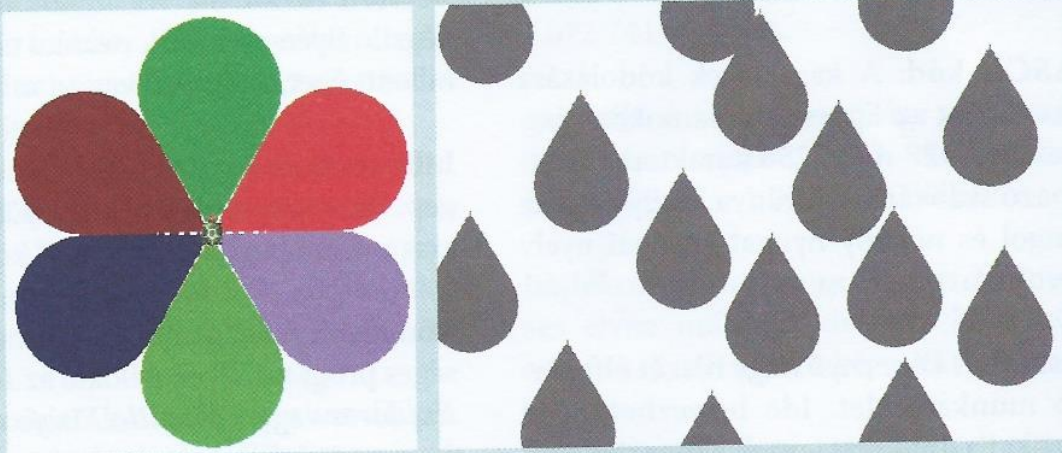
7. Készíts eljárást, amely egy fát rajzol! Az eljárás ismételt meghívásával rajzoljass a teknőccel fasort!



8. Szórj szét fákat tetszőleges módon a képernyőn úgy, hogy abból erdő legyen!
Ügyelj arra, hogy az erdőben a fák törzsei párhuzamosan álljanak!



9. Készíts eljárást, amely egy vízcseppet rajzol!
a) A „vízcsepp” elforgatásával készíts egy virágszirmot!
b) Szórd tele vízcseppekkel a képernyőt! Ügyelj arra, hogy az esőben minden vízcsepp lefelé essen!



10. Az olimpiai játékok közismert szimbóluma az olimpiai ötkarika. Elterjedt magyarázat szerint a kék karika Európát, a sárga Ázsiát, a fekete Afrikát, a zöld Óceániát, a piros pedig az amerikai kontinenst jelképezi. A fehér szín a versengő országok közötti békét és barátságot fejezi ki. A következő ábra az olimpiai ötkarika hivatalos változata. Készíts eljárást, amely a lehető legpontosabban valósítja meg ezt az ábrát!

