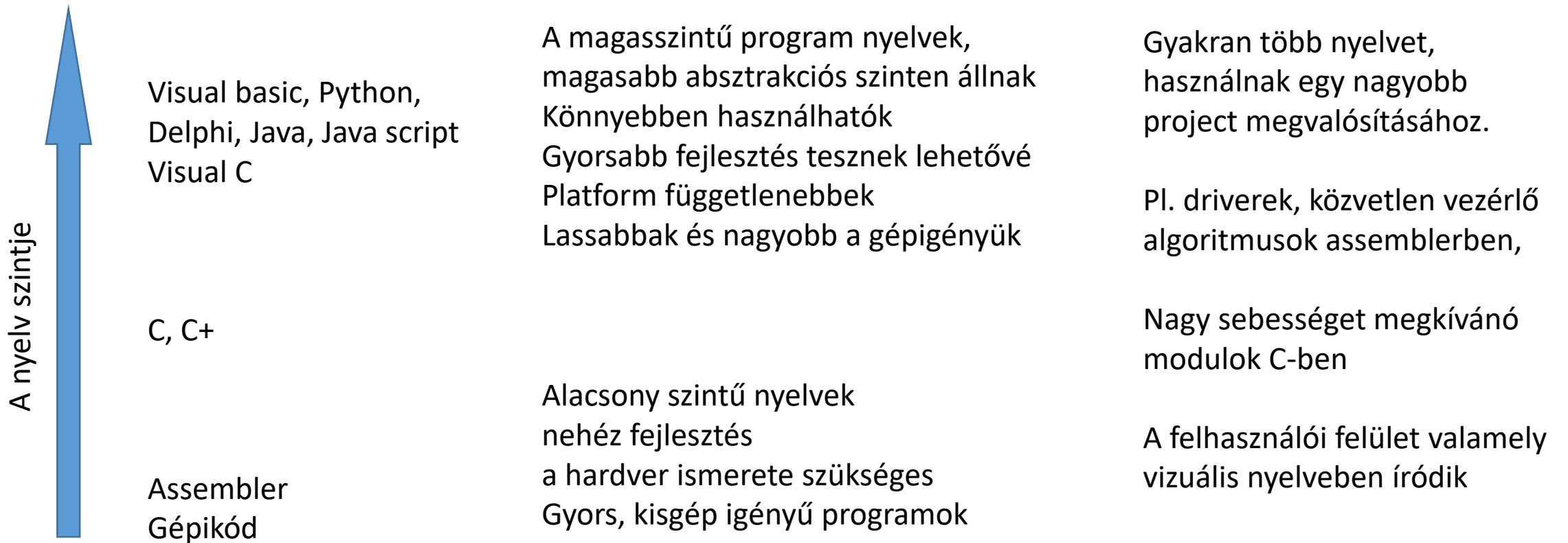


# A program nyelvek csoportosítása 1

A program nyelvek:



# A program nyelvek csoportosítása 2

Megkülönböztetünk általános és specifikus program nyelveket.

Általános célú program nyelvek:

Python, C, basic, Assembly, Java

Specifikus program nyelvek

Logo

R

SQL

Mathematica

oktatás

statisztikai számításokhoz

adatbázis kezelés

matematikai feladatok

# Fejlesztői környezet

Forrás kód:

Egy szöveg fájl, a programozó által írt kód

Közvetlenül a forráskódot a számítógép nem képes futtatni. Ezért szükséges egy fordító (compiler, interpreter).

Futtatható állomány:

Az operációs rendszerben közvetlenül futtatható fájlok:

\*.exe, \*.com,

## **A programozási vagy fejlesztői környezet (integrated development environment) alap egységei**

Szövegszerkesztő (Editor)

Fordító (Compiler vagy Interpreter)

Hiba kereső (Debugger)

Grafikus felület

a forráskód létrehozása szolgáló speciális szövegszerkesztő

a forráskódból a számítógép által értelmezhető gépkódra fordít

a forráskód hibáinak felderítésében segít modul

a vizuális nyelvekben a grafikus környezet (ablakok, gombok, menük..) létrehozására, elhelyezésre szolgál

# A program nyelvek csoportosítása 3

## Compileres, fordítóprogramos

- A forráskódot egyben fordítja le
- Létrehozható önmagában futtatható fájl
- Nehezebb hiba keresés
- A futtatható állomány operációs rendszer függő

## Interpreteres, parancsértelmező

- A forráskódot soronként értelmezi
- Csak a fejlesztői környezetben futtatható a megírt program
- Egyszerűbb hiba keresés
- Platform függetlenebb

# Programnyelv tulajdonságai

## Szintaxis

A programnyelv nyelvtana. Az ismert kifejezések gyűjteménye, a kifejezésekben alkalmazandó szórend, idézőjelek használata stb.

```
If a(1) <> 0 Then
  Cells(sor, 4) = r
  For j = 1 To 10
    Cells(sor, j + 4) = a(j)
  Next j
  sor = sor + 1
End If
```

A program nyelvekben a szintaxis szigorúan kötött, a megengedettől eltérő szintaxisú kódot nem képes értelmezni a fordító!

## Szemantika

A programban helyes szintaxissal írt kifejezések értelme

# A programnyelv felépítése

A programnyelvek egy hagyományos nyelvnek (tipikusan angol) erősen egyszerűsített kifejezésiből és matematikai, logikai kifejezésekből állnak

## Utasítások

„ige”

Amit csinálni kell az adott pontján a számítógépnek

Pl.

Matematikai műveletek  
Logikai műveletek  
Egyéb műveletek

## Változók

„tárgy” vagy „főnév”

Az utasítás „tárgya”, amin végrehajtódik a cselekvés

A számítógép memóriájának egy lefoglalt területe ahol a szükséges adatot tároljuk

# Változók

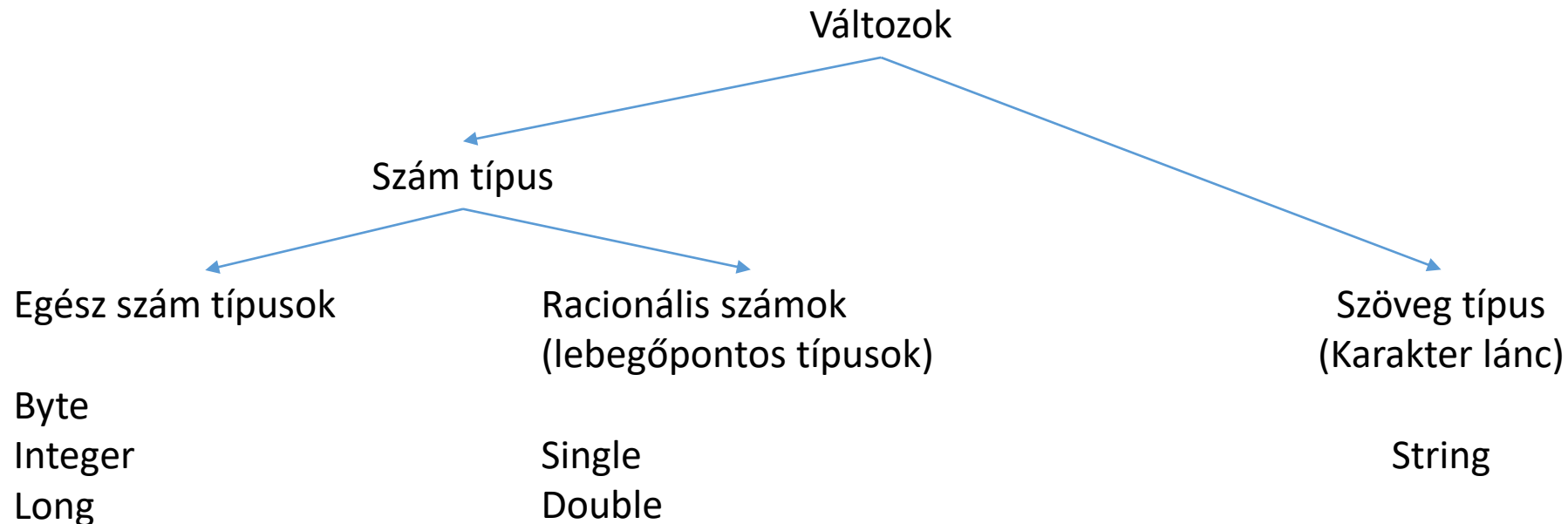
A változó a memória egy a programunk számára fenntartott szegmense, melyre a nevével tudunk hivatkozni, és a melynek értelmezését a változó típusa adja



Név: ezzel hivatkozunk rá a program írása során

Érték: Az aktuális információ amit tárol

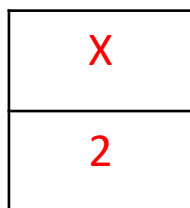
Típus: megadja az értelmezés módját



Irracionális számokkal a számítógép nem tud dolgozni.

# Változók

A változó a memória egy a programunk számára fenntartott szegmense, melyre a nevével tudunk hivatkozni, és a melynek értelmezését a változó típusa adja



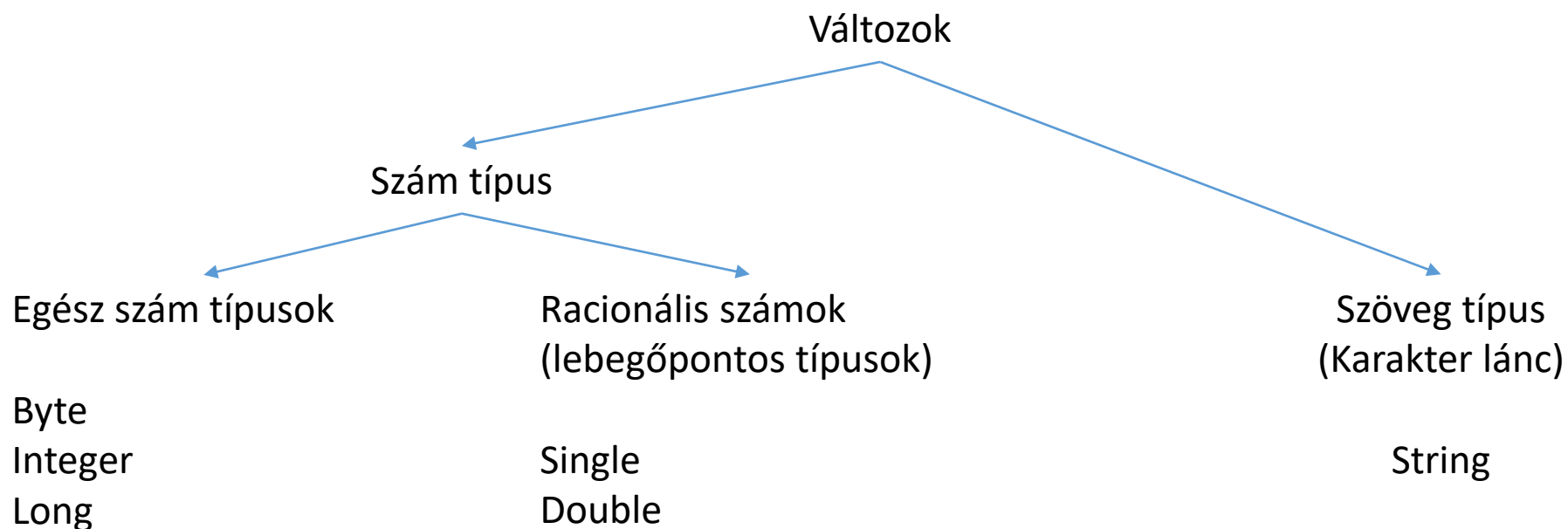
Integer

Név: ezzel hivatkozunk rá a program írása során

Érték: Az aktuális információ amit tárol

Típus: megadja az értelmezés módját

Név bármely karakter  
lánc lehet kivéve a  
fenntartott szó

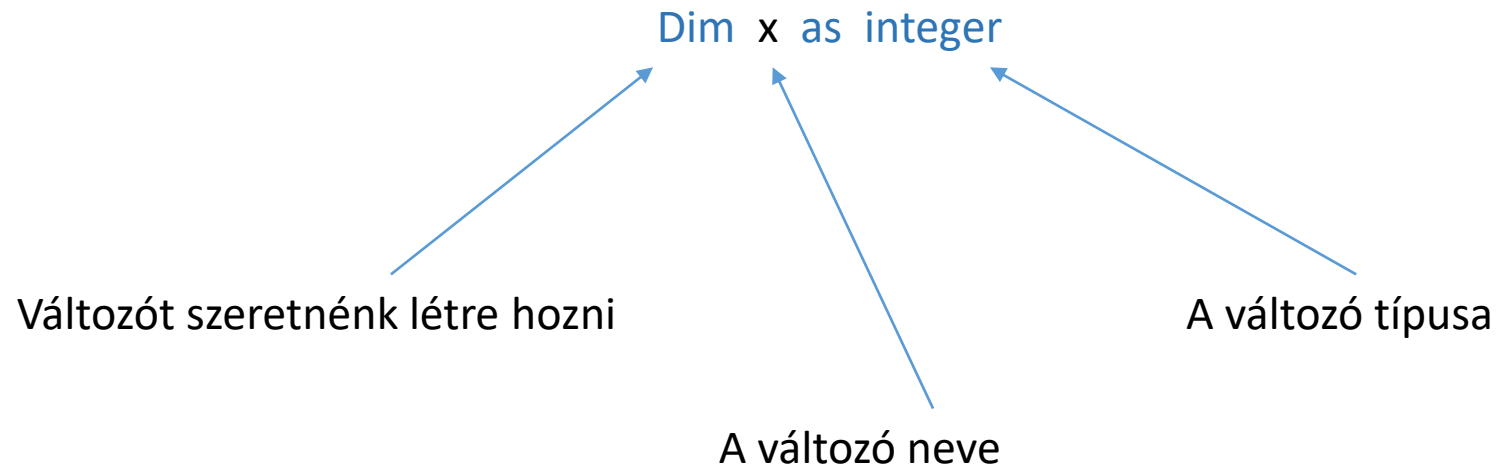


Irracionális számokkal a számítógép nem tud dolgozni.



# Változók létrehozása, deklaráció

Meghatározzuk a változó nevét és típusát



Ennek hatására lefoglal négy byte-ot a memóriában és a fordító felkészül hogy az ott szereplő értéket számként kell majd kezelnie.

# Szám típusok

## Egész szám típusok

típus	A memóriában elfoglalt hely nagysága	Felvehető értékek
byte	1 byte	$2^8$ , 0 .. 255
Integer	2 byte	$2^{15}$ , -32 768 .. 32 767
long	4 byte	$2^{31}$ , -2 147 483 648 .. 2 147 483 647

## Racionális számok (lebegőpontos típusok)

Single 4 byte-on vagyis 32 biten tárolt szám

A szám normál alakját használjuk a tárolásra

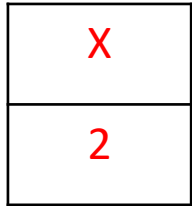
$$q * 10^x$$

$$q \text{ 24 bit} = 2^{23} = -8\,388\,608 \dots 8\,388\,607$$

$$X \text{ 8 bit} = 2^7 = -127 \dots 128$$

$$\begin{aligned} & -3,402823 * 10^{38} \dots -1,401293 * 10^{-45} \\ & 1,401293 * 10^{-45} \dots 3,402823 * 10^{38} \end{aligned}$$

# Az első művelet, érték adás



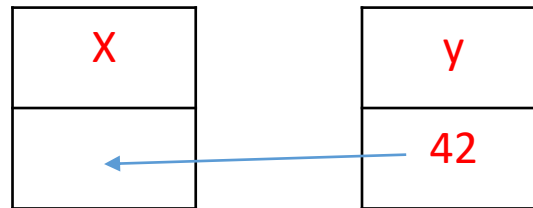
$X = 2$

Az X névvel hivatkozott memória szegmensbe 2 értéket helyezünk el

$2 = X$

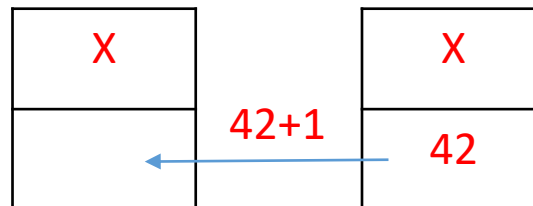
Utasítás értelmetlen, mert a 2-be nem helyezhető x

$X = Y$



X régi értéke felül íródik az Y változó értékére, Y értéke meg marad  
Nem azonos  $Y=X$  kifejezéssel

$X = X + 1$



# Érték adás String típusnál

kutya
ugat

Név: kutya

Érték: ugat

Típus: string

```
Kutya= "ugat"
```

Az idézőjel mutatja meg hogy a *kutya* változó név még az *ugat* csak egy karakter lánc

```
kutya= ugat
```

Parancs sor esetén a *kutya* nevű változóban az *ugat* nevű változó értékét próbálja betölteni

"ugat" = kutya értelmetlen parancs sor

# értékkadás

**Mindig** a bal oldalon szerepel a változó amelynek értéket adunk

$$X = Y * 3 + b23 + \text{valaminev} ^ 23$$

Jobb oldalon tetszőleges bonyolultságú matematikai kifejezés állhat

# Matematikai műveletek

Szám típusú változókra értelmezett műveletek:

+

$$X = Y + 3$$

-

$$X = 2 * X$$

\*

$$Y = X / Y$$

/

Egész osztás

Maradék képzés

String vagyis karakter lánc típusra csak a + van értelmezve ekkor összefűzi a láncokat

A="aaa"

B="bbb"

A + B = "aaabbb"

B + A = "bbbaaa"

# Visual basic for Excel specialitás

A munkalap celláira előre definiált változóval tudunk hivatkozni:

Cells(sor szám, oszlop szám)

A cells(y,x) változó speciálisan szöveg vagy szám típus is lehet.

Érték adásra példák:

Cells(1,1) = 3

Cells(2,3) = Cells(2,3) + 5

Cells(2,3) = "macska"