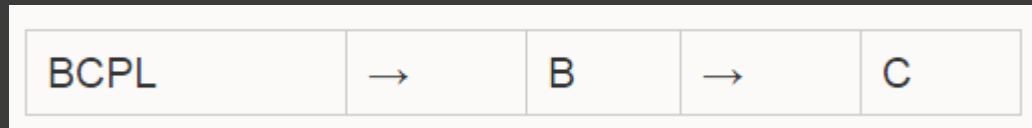


INFORMATIKA LABORATÓRIUM I.

Alapok

Bevezetés

- A C nyelv egy általános célú programozási nyelv.
- Rendszerprogramozási nyelvként is emlegetik, mivel hatékonyan használható operációs rendszerek írására.
- Természetesen használható más alkalmazói programok írására is.
- A nyelvet a BCPL és B nyelvből eredeztetik:



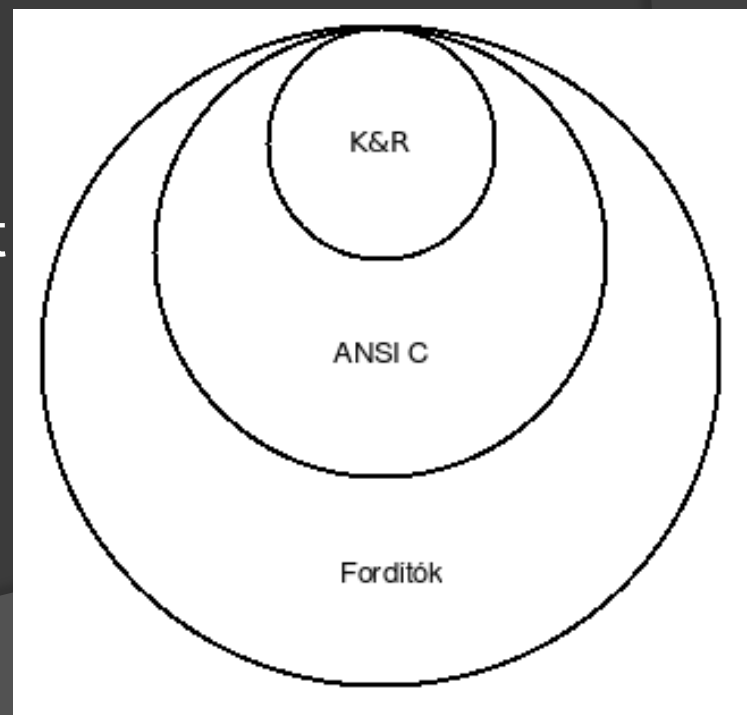
- A C nyelvet **Dennis Ritchie** az 1970-es évek elején fejlesztette ki **Ken Thompson** segítségével.
- A nyelvet UNIX operációs rendszerre tervezte

Bevezetés

- A C sok tekintetben örökölte a B nyelv tulajdonságait, azért is lett a neve az ábécé következő betűje.
- Az 1970es években a személyi számítógépeken BASIC nyelvet lassan felváltja a C nyelv.
- 1978-ban **Dennis Ritchie** és **Brian Kernighan** nevével fémjelzett „***A C programozási nyelv***” című könyv első kiadása megjelenik.
- 1983 – 1989 az Amerikai Nemzeti Szabványügyi Hivatal (angolul: American National Standards Institute, röviden ANSI) szabványnak fogadják el. 1990-ben megjelenik a szabvány: ANSI C
- Újabb szabvány 2000: C99 névvel.
- Újabb szabvány 2011: C11 névvel.

Bevezetés

- ◉ [Bjarne Stroustrup](#) és társai a Bell Labs-nél elkezdtek dolgozni objektumorientált nyelvi elemek hozzáadásán a C nyelvhez. A nyelv, amit készítették a [C++](#) nevet kapta, ez ma a legelterjedtebb programozási nyelv
- ◉ Az ANSI C az eredeti K&R nyelvnek egy kibővített változata. A megvalósítások ehhez még plusz kiterjesztéseket tesznek hozzá.



Jellemzése

- általános célú
- gépfüggetlen
- a periféria kezelés nem a nyelv része
- nem támogatja a párhuzamos feldolgozást (parallel C)
- gazdag függvény könyvtár
- nehéz a kezdőknek: nincs sok merev szabály, nagy a szabadság, az értelmetlen dolgokat is megengedi
- általános célú, de nem minden feladatra ideális, pl. mivel I/O lehetőségei szegényesek adatfeldolgozásra nehezkesebb, nagyszerűen használható sokféle gépközel programozási feladatra (közvetlenül címezhető memóriapozíció és I/O portok)
- a PC-s verziók gazdagabbak I/O egységek kezelésében

Fogalmak

- **Függvény**

A C nyelvben ha utasításról beszélek, gyakran azt mondjuk helyette, hogy függvény. Ez azért van mert minden utasítás valójában egy függvényként van megvalósítva.

- **Kimeneti eszköz**

Mivel számítógépről beszélünk, ezért ez monitor, nyomtató vagy más ehhez hasonló eszköz.

- **Standard Output**

A Standard Output az alapértelmezett kimeneti eszközt jelenti. Szokásos módon ez a monitor képernyője. Tehát ha azt mondom „A Standard Outputra írok”, ez azt jelenti, hogy a képernyőre írok.

Fogalmak

- **Paraméter**

Amikor írunk egy „saját” függvényt, előfordulhat, hogy a hívó programból adatokat akarunk átadni. Ezeket az adatokat a függvény neve után zárójelbe írjuk és paramétereknek nevezzük.

- **Utasítás**

Megmondjuk a gépnek mit tegyen. Ezt úgy tesszük, hogy leírjuk a nevét. Valójában ez egy függvény neve. A C nyelvben ha nincs egy utasításnak paramétere akkor is kötelező a zárójelek kitétele. Minden utasítást pontosvessző (;) zár le! Létezik üres utasítás is, ami egy pontosvessző önmagában.

Fejlesztőeszközök (IDE)

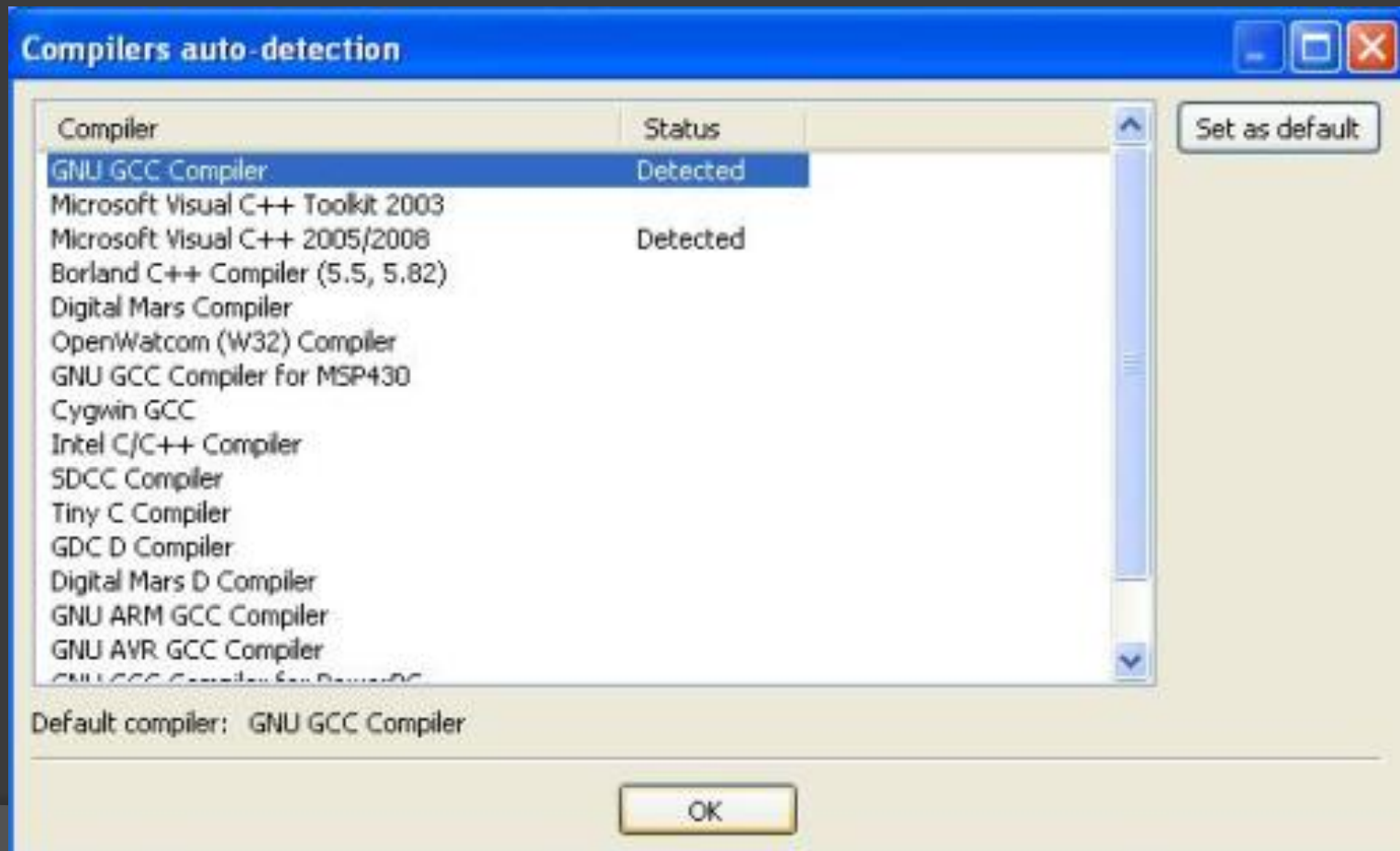
- BloodShed Dev-C++
- Eclipse CDT
- Code::Blocks
- NetBeans IDE
- Microsoft Visual Studio (Express)
- CodeLite
- Turbo C/Borland C++
- Xcode IDE

Code::Blocks

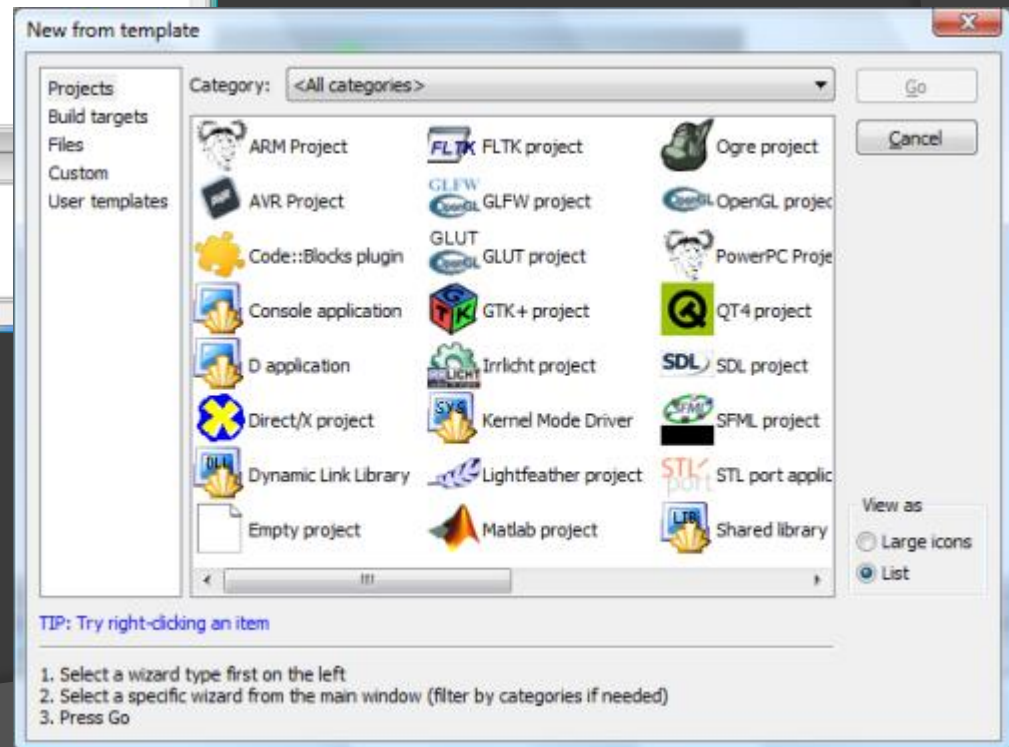
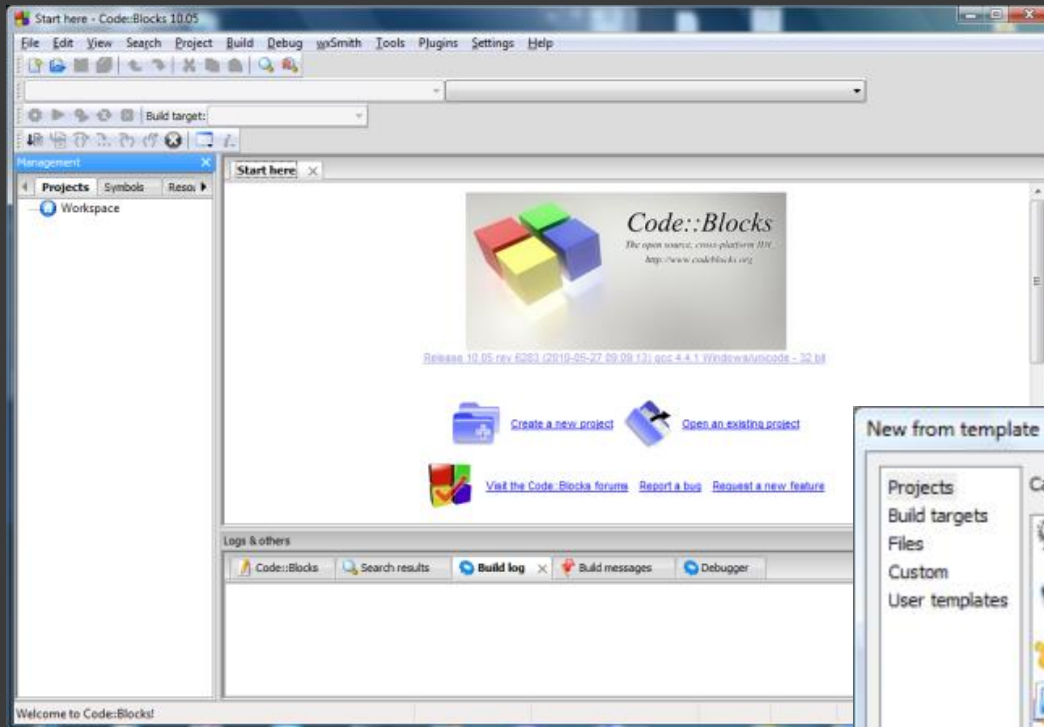
- letöltő helye: <http://www.codeblocks.org>
- használati jog: nyílt forráskódú, szabadon felhasználható
- lényege: **keretrendszer**, amely sokféle platformon (Windows, Mac, Linux), elsősorban **C++** programozási nyelvhez biztosít kényelmes programfejlesztési környezetet; többféle fordítóprogramot képes magába integrálni (javasolt: GNU GCC); lehetővé teszi nyomkövető rendszer beépülését és kényelmes használatát (javasolt: GNU GDB)
- szolgáltatásai (többek közt): több program egyidejű szerkesztése; rugalmasan módosítható szintaxis kiemelés (syntax highlighting); interaktív „kódsúgás” (code completion)

Code::Blocks

A Code::Blocks első elindításakor történik a fordítóprogram hozzárendelése. Ekkor jelzi ki, hogy mely – számára felismerhető – C vagy C++ fordítóprogramok találhatók a lemezen, és választásra kínálja, hogy melyik legyen a feltételezett.



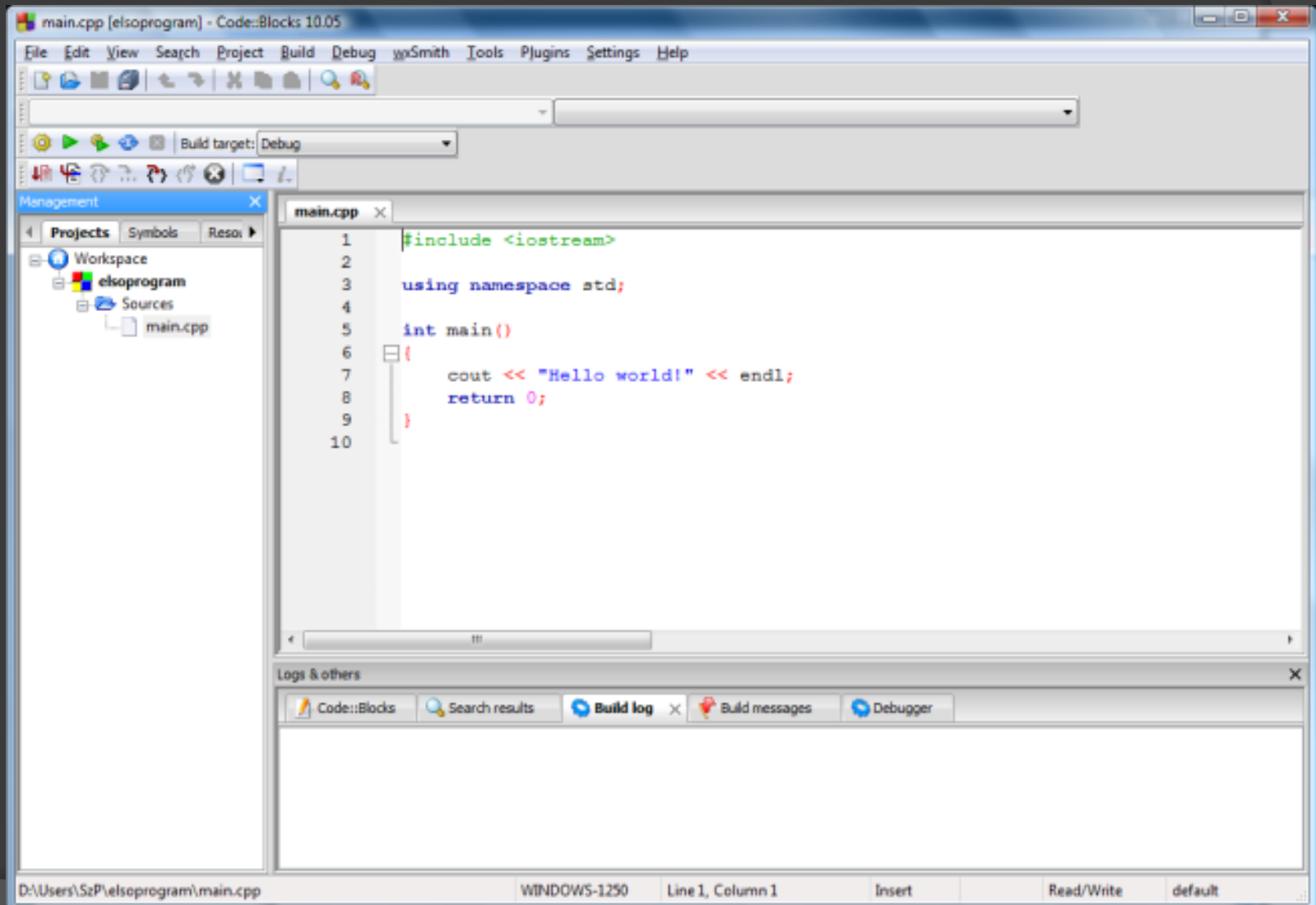
Code::Blocks



Code::Blocks



Code::Blocks



Újgyakorlatok: Hello világ,...

```
#include <stdio.h>

main()
{
    printf("Helló Világ\n");
}
```

```
#include <stdio.h>

main()
{
    printf("alma");
}
```

```
#include <stdio.h>

main()
{
    puts("alma");
}
```

A C program szerkezete

- ◉ Egy C program egy vagy több modulból áll.
- ◉ A modulok valamelyikének tartalmaznia kell egy **main** nevű függvényt. Ez lesz a program belépési pontja, vagyis ez kezd el végrehajtódni.
- ◉ **Modul:**
Minden modul külön file. A modulok önállóan lefordíthatók, úgynevezett tárgykóddá. A tárgykódú modulokat az összeszerkesztő (**linker**) állítja össze egy programmá. Egy modulus program esetén is van dolga az összeszerkesztőnek, hiszen a modulunkban használt könyvtári függvények (a C nyelvhez adott előre megírt függvények, pl. `printf()`) egy vagy több tárgykódú modulban vannak elhelyezve.
- ◉ Egy modul tartalmazhat:
 - fordítónak szóló utasításokat (pl. `#include <stdio.h>`)
 - deklarációkat, definíciókat (változó, típus, stb.)
 - függvényeket (pl. `void main() { ... }`)
 - megjegyzéseket

Fordítás

- ⦿ A C nyelv egy compileres nyelv.
- ⦿ A fordítás több menetben történik:
 - feldolgozza a fordítónak szóló utasításokat (előfordítás),
 - lefordítja a programot tárgykódúvá,
 - összeszerkeszti a többi modullal.
 - Eredménye egy futtatható file.

Állandók vagy konstansok

- ⦿ Az állandó olyan érték, amelyet a továbbiak során nem akarunk megváltoztatni.
- ⦿ Kétfajta állandót használunk: literális és nevesített.
- ⦿ Literális állandó:
lehet szám, karakter, karaktersorozat, stb.
- ⦿ Számállandó: 1234 formában leírt szám egész szám
- ⦿ Karakteres állandó: egy vagy több aposztrófok (') közzé írt karakter. Pl.: 'a' vagy 'b,
- ⦿ Karakterként nem szerepelhet a ' vagy az újsor. Ez ért ezek helyettesítésére ún. escapesorozatok (escape szekvencia) használunk.

Escape szekvencia

új sor	NL	\n
vízszintes tabulátor	HT	\t
függőleges tabulátor	VT	\v
visszalépés (backspace)	BS	\b
kocsivissza	CR	\r
lapemelés (formfeed)	FF	\f
hangjelzés (bell)	BEL	\a
backslash	\	\\
kérdőjel	?	\?
apostroóf	'	\'
idézőjel	"	\"
oktális szám	ooo	\ooo
hexadecimális szám	hh	\xhh

Állandók vagy konstansok

- Szöveges állandó

Szokásos elnevezése: „karakter sorozat”. A szöveget úgy adjuk meg, hogy maga a program azon változtatni nem tud a végrehajtás során, vagyis állandó. A C nyelvben a szöveges állandót idézőjelek között adjuk meg. Pl.: "Alma a fa alatt "

- Nevesített állandó

Az állandó számára egy azonosító nevet adunk meg:
`#define TELJESNEV "Nagy Peter"`

- A C nyelvben lehetőség van még egy módon állandó megadására a **const** módosítóval. Pl.: `const int a;`

Deklarációk

- A felhasználása előtt minden változót deklarálni kell, bár bizonyos deklarációk a programkörnyezet alapján is létrejöhetnek.
- A deklaráció egy típust határoz meg, és utána egy vagy több adott típusú változó felsorolása (listája) áll.

```
int also, felso, lepes;  
char c, sor[1000];
```

```
int also;  
int felso;  
int lepes;  
char c;  
char sor[1000];
```

```
int i = 0;  
char esc = '\\';  
int hatar = MAXSOR+1;  
float eps = 1.0e-5
```

```
const double e = 2.71828182845905;  
const char uzenet [ ] = "figyelem:";  
int strlen(const char[ ]);
```

Változók és adattípusok

Típus	Bájt	Minimális érték	Maximális érték
char, signed char	1	-128	127
unsigned char	1	0	255
short, short int, signed short int	2	-32768	+32767
unsigned short, unsigned short int	2	0	65535
int, signed int	2 vagy 4	short vagy long	short vagy long
unsigned unsigned int	ugyanígy, de unsigned	ugyanígy, de unsigned	
long, long int, signed long int	4	-2147483648	+2147483647
unsigned long, unsigned long int	4	0	4294967295
char, unsigned char	1	0	255
signed char	1	-128	127

Ellenőrző kérdések

- Milyen nyelvet tanulunk most?
- Ki találta ki ezt a nyelvet amit tanulunk?
- Mikor lett a nyelv kifejlesztve?
- Mire használjuk?
- Van-e szabványosított verziója?
- Mi az az előfordítás?
- Mivel kezdjük az előfordítói utasításokat?
- Mi az a konstans?
- Hogyan hozunk létre C nyelvben konstanst?
- Milyen az írásmódja a változóknak és a konstansoknak?
- A C nyelv kis és nagybetű érzékeny?
- A C nyelv kiírató utasítása?
- Hol nem szükséges a C nyelvben az utasítás végére (;) pontosvessző?