# Adattípusok

## Változók

típus + név

hozzárendelhető érték!

* int a;
* a = 3;
* int b;
* int b = 3;
* int c = 2, d = 5;
* int e(4);
* int f {5};

## Elemi

* int fixpontos(egész)
* float lebegőpontos (4 byte)
* double dupla lebegőpontos (8 byte)
* bool I/O
* char betűtípus
* wchar\_t „széles” típus
* {u}char8\_t
* {u}char16\_t
* {u}char32\_t
* short short int: tárolási egység
* long long int / long: 4 byte minimum, long double: 10/8 byte
* signed signed int: legalább 2 byte
* unsigned
* long long 8 byte

## Numerikus literálok

* 19 int
* 19L long
* 19UL unsigned long
* 0.6 double
* 0.6L long double
* 0.6F float
* 6E23 6\*1023
* 6E-23 6\*10-23

## Karakterek

* char minimum 256 érték tárolására alkalmas kar. vagy num.
* signed a default
* unsigned 8 bites értékek tárolására használható

## Típuskonverzió

* int a = 15;
* double b = a; //aut. típuskonverió
* cout << (double)a;
* cout << double(a);

# Literálok

## Char

* ’A’
* ’\r’ = Carriage Return
* ’\n’ = new line
* ’\t’ = tabulátor
* ’\b’ = backspace
* ’\a\ = alert
* ’\\’ = \
* ’\”’ = „
* ’\0’ = ASCII kód: 0

## Numerikus fixpontos

* 19 : decimális
* 0x19 : hexadecimális
* 019 : oktadecimális
* 0b01011 : bináris

## Bool

* true != 0
* false = 0

## Vátozónevek

* Alfanumerikus karakterek : {a,z} {A,Z}
* Számjegy, nem az első pozíciótol
* ’\_’ (underscore)
* globális változó g\_myvar
* osztály változó c\_myvar
* objektum változó m\_myvar

## Szimbolikus állandók

* const int a = 124;
* conts double PI = 3.1415926535;
* const A = 12; //int

## Aritmetikai operátorok

* + , - , \* , / , %
* %-ot kivéve lehetnek a bemenő paraméterek fix (char is) és lebegőpontosak is
  + ’A’+2 =C
  + int a = ’A’+2
* Ha mindkét paraméter int, az eredmény int
* /: int a = 4, b = 3;
* int c = a/b; //c=1
* double d = a/b; //d=1.3333
* cout << a/b; //1
* cout << a/(double)b; //1.3333

## Relációs operátorok

* > , < , >= , <= , != , ==
* ’A’ > ’B’ => false
* ’A’ < ’B’ => true
* Kif1 > kif2
  + kifejezés : kiértékelhető
  + sorozat : operátorok, változók, konstansok
* Kif :
  + Visszatérő érték
  + Mellékhatás

## Logikai operátorok

* bemeneti paraméter : bool
* && : logikai ÉS
* || : logkai VAGY
* ! : logikai NEM

## Bitenkénti operátorok

* & : bitenkénti ÉS
* | : binetnkénti VAGY
* ^ : bitenkénti XOR
* ~ : bitenkénti NEGÁLÁS
* << : balra tolás
* >> : jobbra tolás

## Auto increment/decrement operátorok

* ++n;
* n++;
* int n = 5, a = 0;
* a = ++n; //a=6
* int n =5, a = 0;
* a =n++; //a = 5, n=6

## Értékadó operátorok

* = : lvalue = rvalue //lvalue : írható, rvalue : kiértékelhető
* int a, b, c;
* a = b = c = 3;