



Intro în limbajul C

IoT Bootcamp 2020

Agenda sesiunilor de azi

17:30-18:30	Intro în limbajul C
18:30-19:00	Pauză
19:00	Team building online

Prezentatorii sesiunii de azi



Diana Marusic

Project Manager IoT Bootcamp



Vasile Păpăluță

Președintele Clubului
Ingineresc Micro Lab

IDE online pentru C

https://www.onlinegdb.com/online_c_compiler

Ce este C?

Un limbaj de programare:

- **procedural**
- **de uz general**
- dezvoltat de către **Dennis Ritchie**, în cadrul **Bell Labs** începând cu **1972**
- Scop inițial: scrierea **nucleului(kernel) sistemului de operare UNIX**
- Prezent - **cel mai popular limbaj de programare** pentru scrierea de **software de sistem**
- Apreciat pentru **eficiența codului generat de compilatoarele C**, și pentru **portabilitatea sa**.

De ce învățăm C?

- Arduino – bazat pe C, C++
- Embedded C – utilizat pentru programarea microcontroloarelor
- Cel mai popular limbaj pentru software de sistem
- Limbaj foarte popular in diferite domenii de aplicații
- Smart home

Github review – most popular languages 2019

Programming Language	Ratings	Change
Java	16.028%	-0.85%
C	15.154%	+0.19%
Python	10.020%	+3.03%
C++	6.057%	-1.41%
C#	3.842%	+0.30%
Visual Basic .NET	3.695%	-1.07%
JavaScript	2.258%	-0.15%
PHP	2.075%	-0.85%
Objective-C	1.690%	+0.33%
SQL	1.625%	-0.69%

Structura unui program C

```
#include <stdio.h>
```

```
/**
```

```
Comentariu pe mai multe randuri
```

```
**/
```

```
int main() {
```

```
    // Codul programului
```

```
    return 0;
```

```
}
```


Librării

```
#include <stdio.h>
```

```
/**
```

```
Comentariu pe mai multe randuri
```

```
**/
```

```
int main() {
```

```
// Codul programului
```

```
return 0;
```

```
}
```



Librării

```
#include <stdio.h>
```

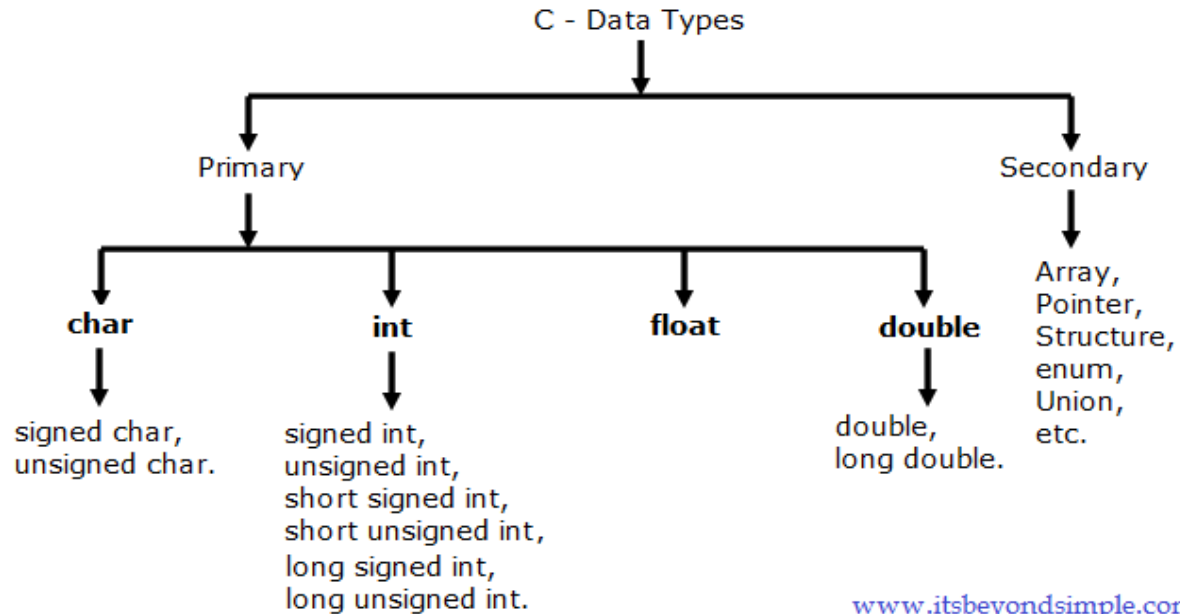
```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <Arduino.h>
```



Tipuri de date



www.itsbeyondsimple.com

fig: C Data Types

Tipuri de date

Data Type	Size in Bytes	Range	Format
signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
short signed int	2	-32768 to -32767	%d
short unsigned int	2	0 to 65535	%u
signed int	2	-32768 to -32767	%d
unsigned int	2	0 to 65535	%u
long signed int	4	-2147483648 to +2147483647	%ld
long unsigned int	4	0 to 4294967295	%lu
Float	4	3.4e-38 to 3.4e+38	%f
Double	8	1.7e-308 to 1.7e+308	%lf
long double	10	3.4e-4932 to 1.1e+4932	%LF

Variabile

- Identificatori (nume) atribuite locațiilor din memorie care stochează anumite valori.
- Variabilele stochează valori de tipurile care au fost definite

Variabile

- Identificatori (nume) atribuite locațiilor din memorie care stochează anumite valori.
- Variabilele stochează valori de tipurile care au fost definite

Exemple:

```
int i = 0;
```

```
float pi = 3.14;
```

```
char literaDinAlfabet = 'A';
```

Variabile – reguli pentru denumiri

- Un nume de variabilă poate conține doar litere (mari și mici), numere și simbolul underscore _
- Prima literă a unei variabile este întotdeauna o literă sau un underscore _
- Nu există limite la lungimea numelui unei variabile, însă unele compilatoare au probleme cu variabile mai lungi de 31 caractere

Variabile Globale și Variabile Locale

```
#include <stdio.h>
```

```
// Definim aici variabilele globale
```

```
int numar_pasi;
```

```
int main() {
```

```
    //Putem defini aici variabilele locale
```

```
    int rezultat = 10;
```

```
    printf("%d", rezultat);
```

```
    return 0;
```

```
}
```


Variabile Globale și Variabile Locale

```
#include <stdio.h>
```

```
// Definim aici variabilele globale
```

```
int numar_pasi;
```

```
int funcSuma() {
```

```
    //Putem defini aici variabilele locale
```

```
    int rezultat = 10;
```

```
    return rezultat;
```

```
}
```

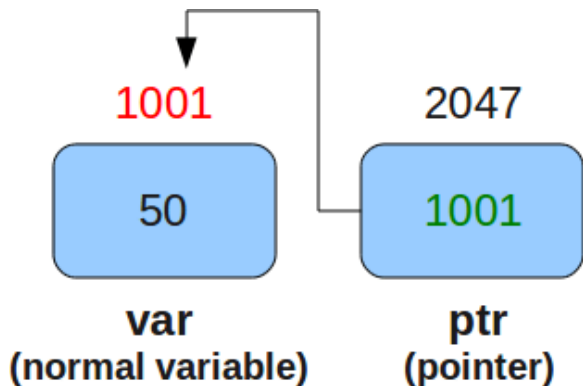
```
int main() {
```

```
    return 0;
```

```
}
```

Adrese, pointeri și referințe

- Fiecare locație din memore are o **adresă unică**.



C - Pointers

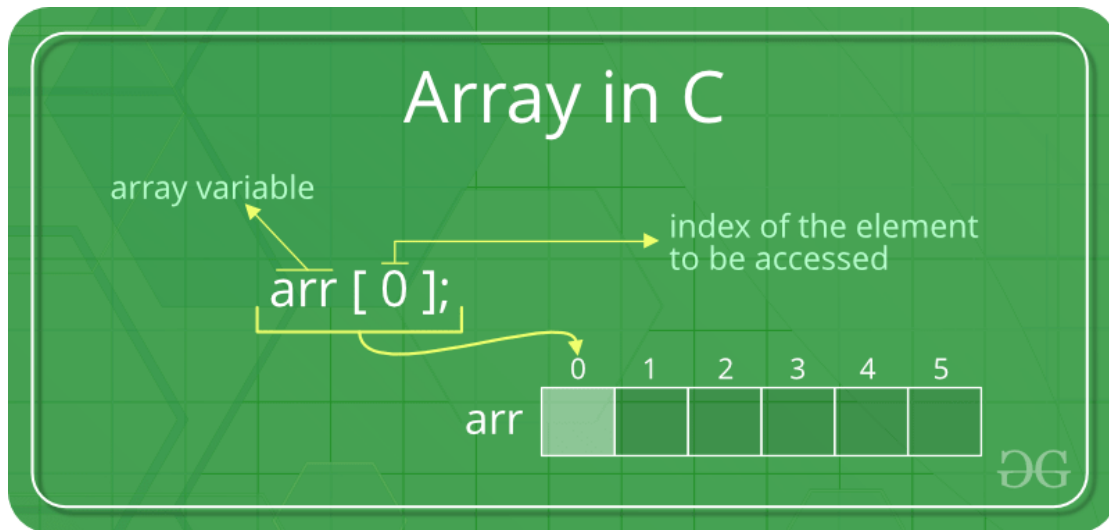
```
int var = 10;  
int *p;  
p = &var;
```

The diagram shows a pointer variable 'p' at memory address 0x7fff5ed98c50. It contains the address 0x7fff5ed98c4c. An arrow points from this address to the variable 'var' at memory address 0x7fff5ed98c4c, which contains the value 10. The text 'Beginnersbook.com' is visible in the background.

P is a pointer that stores the address of variable var.
The data type of pointer p and variable var should match because an integer pointer can only hold the address of integer variable.

Array (Tablou)

- O colecție de elemente de același tip stocate în locații continue de memorie.
- Elementele pot fi accesate printr-un index (indice)



Array (Tablou)

Array Declaration in C

`int a[3];`

2192	451	13918
------	-----	-------

`int a[3]={1, 2, 3};`

1	2	3
---	---	---

`int a[3]={1, 1, 1};`

1	1	1
---	---	---

`int a[3]={ };`

0	0	0
---	---	---

`int a[3]={ 0 };`

0	0	0
---	---	---

`int a[3]={ 1 };`

1	0	0
---	---	---

`int a[3]={ [0...1]=3 };`

3	3	0
---	---	---

`int a[]={ [0...1]=3 };`

3	3
---	---

`int *a;
int* a;
int * a;
int*a;`



Operații aritmetice

Operația	Explicație
+	Adunare
-	Scădere
*	Înmulțit
/	Împărțit
%	Modulo (restul împărțirii)

Operatori Increment și Decrement

Operația	Explicație
++	Incrementare (+1)
--	Decrementare (-1)

Operatori de asignare

Operația	Explicație
=	Atribuire / Asignare
+=	$X += 2 \iff X = X + 2$
-=	$X -= 2 \iff X = X - 2$
*=	$X *= 2 \iff X = X * 2$
/=	$X /= 2 \implies X = X / 2$

If

```
if( /** Conditie **/ ) {  
    // Se executa daca Conditie adevarata  
}
```


If, Else

```
if( /** Conditie **/ ) {  
    // Se executa daca Conditie adevarata  
} else {  
    // Se executa daca Conditie falsa  
}
```

If, Else

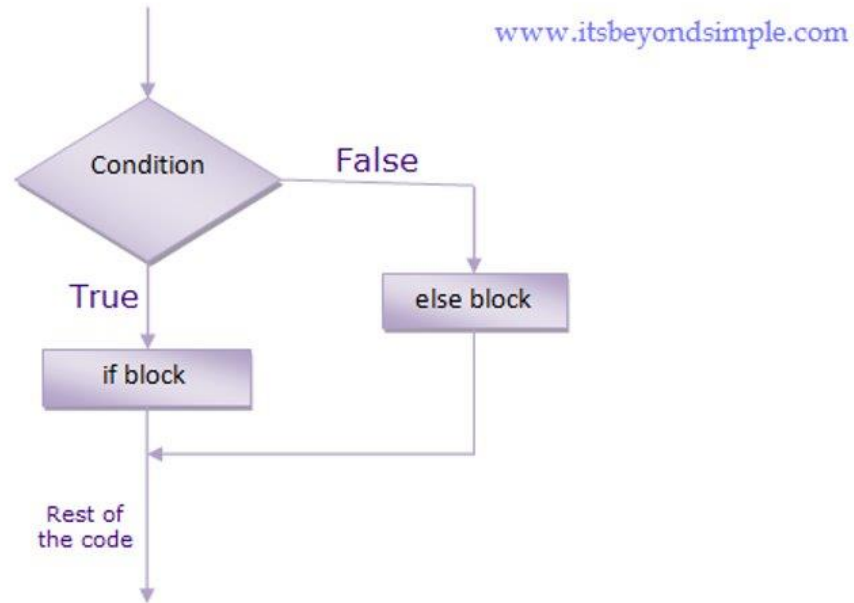


figure : *if-else* statement flowchart

For

```
int i = 0;
```

```
for(i=0; i<nr; i++) {  
    // Se executa de nr ori  
}
```

While

```
int i = 0;
```

```
while( i<10) {
```

```
    // Se executa de nr ori
```

```
    i++;
```

```
}
```

Operații logice – and, or, not

Operator	Explicație
&&	AND (ȘI) Adevărat doar dacă ambele condiții sunt adevărate
	OR (SAU) Adevărat dacă măcar una din condiții e adevărată
!	NOT (NEGARE) Valoarea negată (inversă)

Tabele de adevăr

Expresie	Rezultat
True && True	True
True && False	False
False && False	False

Tabelul and (și)

Expresie	Rezultat
True True	True
True False	True
False False	False

Tabelul or (sau)

Expresie	Rezultat
!True	False
!False	True

Tabelul not(negare)

#define, simboluri

Sintaxa:

```
#define NUME_SIMBOL 10
```

Exemplu:

```
#define PIN_LED 2
```

```
PIN_LED + 3; // Va avea valoarea 2+3=5
```

Structuri

- **Structura** este un tip de date definit de către utilizator pentru a stoca date eterogene (de diferite tipuri) într-o singură variabilă.

The diagram shows a C struct definition with annotations. A vertical arrow points from the word 'struct' to the label 'Struct Keyword'. A diagonal arrow points from the word 'student' to the label 'tag or structure tag'. A right-facing curly brace groups the three member declarations, with a label 'Members or Fields of Structure' pointing to it. A watermark 'Sitesbay.com' is visible in the background.

```
struct student
{
    char student_name[10];
    int roll_no[5];
    float percent;
};
```

Struct Keyword

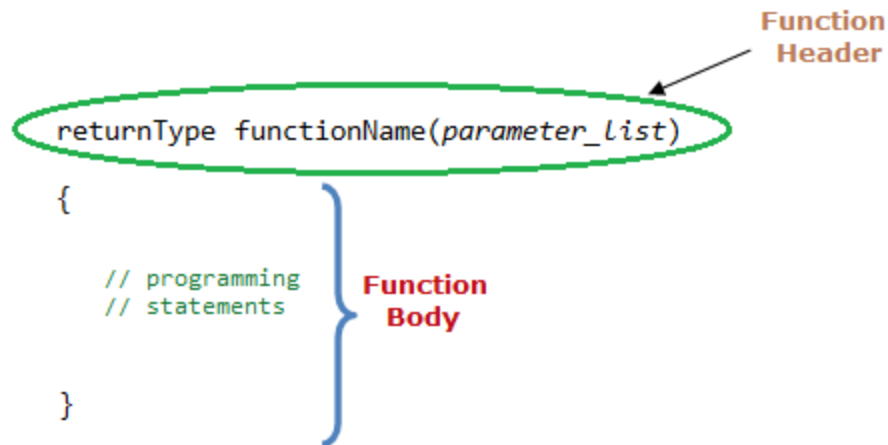
tag or structure tag

Sitesbay.com

Members or Fields of Structure

Funcții

- O parte din program care execută un anumit task și de obicei returnează un rezultat



Referințe (pointeri) la funcții

```
#include <stdio.h>
// A normal function with an int parameter
// and void return type
void fun(int a)
{
    printf("Value of a is %d\n", a);
}

int main()
{
    // fun_ptr is a pointer to function fun()
    void (*fun_ptr)(int) = &fun;

    /* The above line is equivalent of following two
       void (*fun_ptr)(int);
       fun_ptr = &fun;
    */

    // Invoking fun() using fun_ptr
    (*fun_ptr)(10);

    return 0;
}
```

Mai multe resurse

- <https://learnxinyminutes.com/docs/c/>
- <https://www.learn-c.org/>
- <https://www.tutorialspoint.com/cprogramming/index.htm>
- <https://www.itsbeyondsimple.com>

Pauză
Team building la **19:00**