

## Stringhe di caratteri

```
char nome[3] = "Pippo";
```

P	i	p	p	o	\0
---	---	---	---	---	----

```
char cognome[100];
```

```
printf("inserisci cognome: ");
```

```
scanf("%s", cognome);
```

\0:

carattere  
terminatore

inserisci cognome: Rossi: invio

R	o	s	s	i	\0	.	.	-	f	.	.	.	.
---	---	---	---	---	----	---	---	---	---	---	---	---	---

```
printf("cognome: %s\n", cognome);
```

Rossi

```
for(int i=0; i<100; i++) {
```

```
    printf("cognome [%d] = %c\n", i, cognome[i]);  
}
```

```
cognome[0] = 'R'
```

```
cognome[1] = 'o'
```

:

```
cognome[4] = 'i'
```

```
cognome[5] = '\0'
```

```
cognome[6] = '.'
```

:

```
cognome[93] = '.'
```

}

imprevedibile

```
cognome = "Rehefou";
```

Errone

## Funzione sprintf

```
sprintf(cognome, "Rahatlou");
```

R|a|h|a|t|l|o|u| | . . .

```
int ncifre;
```

```
printf("inserisci #decimeli: ");
```

```
scanf("%d", &ncifre);
```

```
char formato[100];
```

formato = "x = %3lf\n"  
          ncifre

```
sprintf(formato, "%s %d %s", "x = ", ncifre, "lf\n");
```

formato = { x | = | : | 7 | l f | \n | \0 | }

```
formato[0] = 'N';
```

"N = %7lf\n"

```
double x = -1.123456;
```

```
printf(formato, x);
```

```
printf("x = %7lf\n", x);
```

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

```
printf(" - - ", dati);
```

*Non funziona.*

```
char gioco[3][3] = { "", "", "", "", "", "", "",  
                     "", "" };
```

gioco[0][0] →

	⊗	
	○	
		X

gioco[0][1] = "X";

gioco[1][1] = "O";

Estraggo  $i, j \in [0, 2]$

rand(68)/3  
if (gioco[i][j] == "") {

// riemp: casella

}

int vuoto = 0;  
1 Controlla Casella

for (i

for (j

if (gioco[i][j] == "") {

vuoto++;

}

?

int i;

char nome[100];

scanf("i.d", &i);

scanf("i.s", nome);

Puntatori:

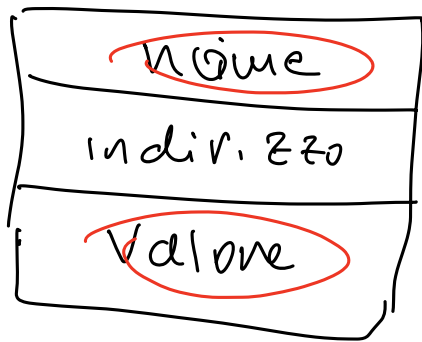
int i = 3;

int dati[4] = {1, 2, 3, 4}



dati[0]	dati[1]	dati[2]	dati[3]
1	2	3	4

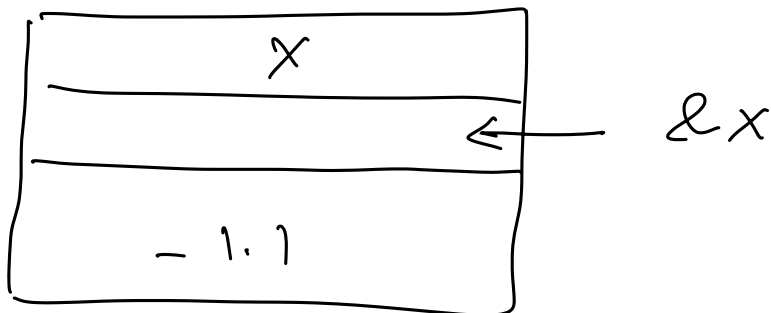
Variable



int i = 3;

&i is indir. zero

double x = -1.1;



float pippo;



&pippo

Dichiarazione puntatori:

```
int i = 3;
int * ptr;
ptr = &i;
```

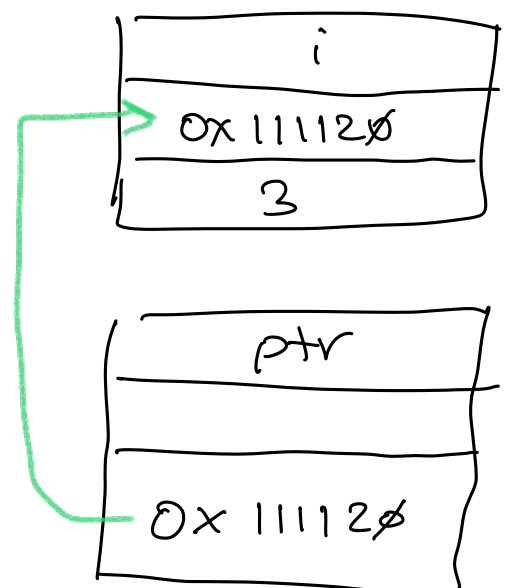
printf("&i = %p \n", &i);

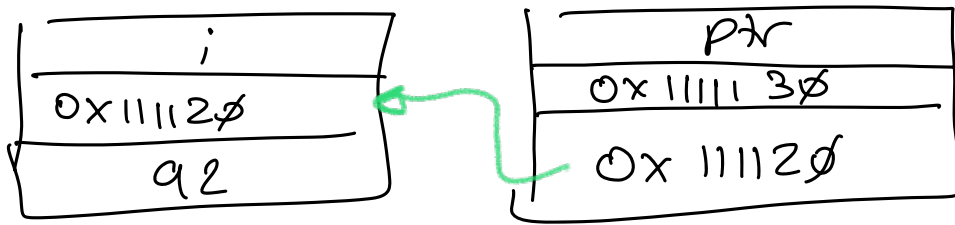
Output: &i = 0x111120

printf("ptr = %p \n", ptr);

\*ptr = 92;

dereferenzia ptr





\*ptr: variabile cui in indirizzo è ptr

```
printf("&ptr = %p\n", &ptr);
```

```
int j = 4;
```

```
ptr = &j;
```

```
*ptr = -2;
```

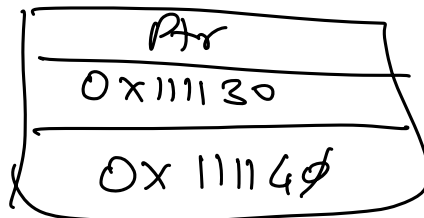
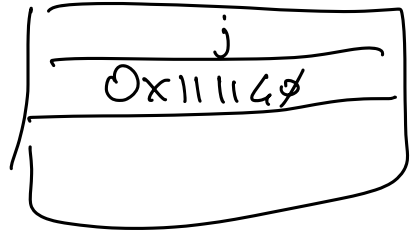
```
double x;
```

```
double * p2;
```

```
p2 = &x;
```

```
ptr = &x;
```

Errore di compilazione  
tipo puntatore errato



cambia j

## Puntatori e Array

```
int dati[3] = { 1, 2, 3 }
```

```
for (int i=0; i<3; i++) {
```

```
    printf("dati[%d] = %d it &dati[%d] = %p\n",  
           i, dati[i], i, &dati[i] );
```

```
}
```

dati[0]	dati[1]	dati[2]
0x--48	0x--4c	0x--50
1	2	3

$\text{dati}[0] = 1$        $\&\text{dati}[0] = 0x16f0077$  **48**  
 $\text{dati}[1] = 2$        $\&\text{dati}[1] = 0x16f0077$  **4C**  
 $\text{dati}[2] = 3$        $\&\text{dati}[2] = 0x16f0077$  **50**

int: 32 bit 4 byte

double x[100] = { 1, 2, ..., 99, 100 };

double: 64 bit = 8 byte

double \* ptr = &x[0];  
printf( "%ptr = %lf\n", \*ptr );

output: \*ptr = 1.

ptr = &x[37];

output: \*ptr = 38

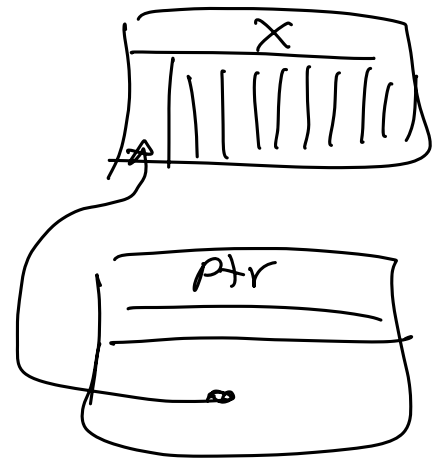
ptr = &x[92];

ptr = ptr + 7;

printf( "%ptr = %lf\n", \*ptr );

ptr → x[92]

ptr → x[99]



int voti[3];

voti: puntatore alla 0-esima cella.

voti[0]	voti[1]	voti[2]
0x...48	0x...4C	0x...50
30	27	29

$\& \text{voti}[0]$   
 $\text{voti}$  } la stessa cosa.

$\& \text{voti}[2]$

$\text{voti} + 2$  : pointer a  $\text{voti}[2]$

$\text{printf}("i.d\n", \text{voti}[2]);$

$\text{printf}("i.d\n", * (\text{voti} + 2));$  } <sup>29</sup>  
sullo  
scremo.

$\text{double } x[100];$

$\text{double* } p3 = x + 37;$

$\hookrightarrow \& x[0]$   
 $\hookrightarrow x[37]$

$p3 += 8 ; \longrightarrow x[45]$

$\text{for}(p3 = x; p3 < x + 100; p3++) \{$

$\text{printf}("x = %lf\n", *p3);$

$\}$

$\text{voti}[0]$	$\text{voti}[1]$	$\text{voti}[2]$
-5	2	3

$* \text{voti} + 2$  intero = -3 Dereferenzio voti poi somma.

$* (\text{voti} + 2)$  3 prima somma pointer poi Dereferenz.

$* \text{voti} \equiv \text{voti}[0]$

$* (\text{voti} + 2) \equiv \text{voti}[2]$

```
scanf("%d", &i)
```

puttatore a i  
indirizzo di i

```
char nome[100];
```

```
scanf("%s", nome);
```

memoria di &nome[0]

Nome di array: puttatore speciale

```
float dati[37];
```

dati  $\equiv$  &dati[0]

## I/O con file in C

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

```
int main() {
```

```
    char fname[100];
```

```
    printf("inserisci nome file (max 100): ");
```

```
    scanf("%s", fname);
```

"esempio.txt"

```
FILE * fp;
```

w: write  
in Scrittura.

```
fp = fopen(fname, "w");
```

crea file con  
nome fname

```
if (fp == NULL) {
```

```
    printf("errore con file %s\n", fname);
```

```
    exit(-1);
```

```
}
```



"esempio.txt" ✓

" / esempio.txt" ✗ errore

```
srand48( time(0) );
```

```
for( int i = 0; i < 1000; i++ ) {
```

```
    double x = -1. + 2. * lrand48() / RAND_MAX;
```

```
    fprintf(fp, "%d\t%.10lf\n", i, x);
```

```
}
```

```
fclose(fp);
```

le stesse regole di printf

gcc -o app.exe esempio.c -lm

• ./app.exe

inserisci nome file (max 100 caratteri): Pippo.dat

ls: appare nuovo file Pippo.dat