

PUNTATORI: consentono di realizzare in modo implicito il passaggio per riferimento

Fondamentali per la costruzione di **strutture dati dinamiche** (liste, code, stack, alberi)

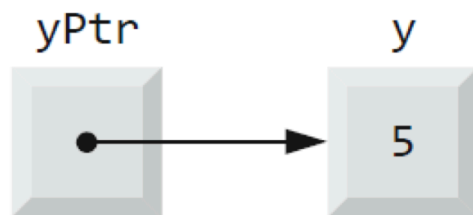
A differenza delle altre variabili (nome, tipo, locazione, valore) un puntatore è una variabile i cui valori sono indirizzi di memoria

&: otteniamo l'indirizzo di memoria

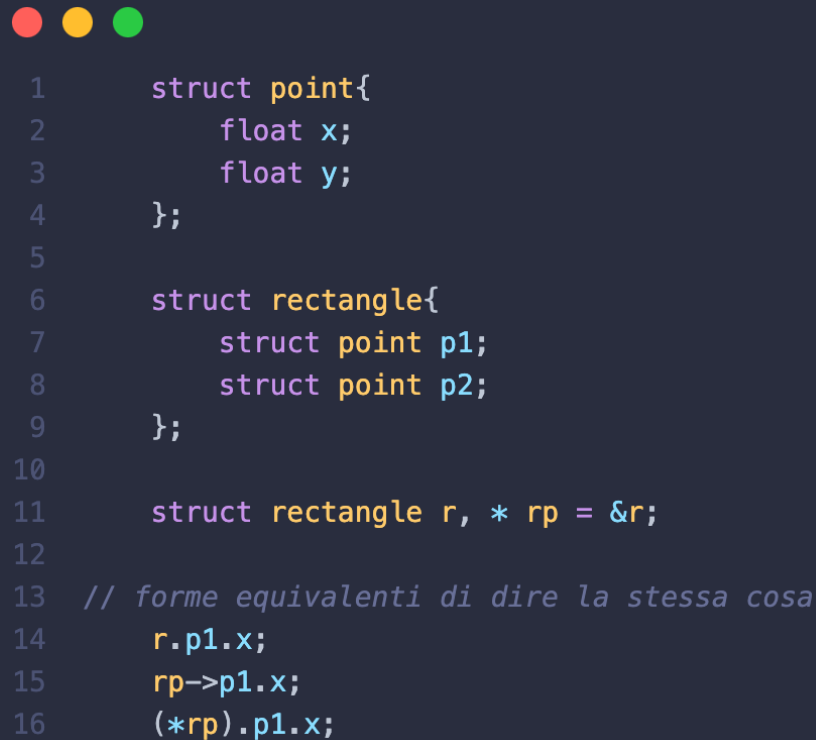
```
int y=5;  
int * yPtr;
```

*//Assegna l'indirizzo della variabile y
// alla variabile puntatore yPtr*

```
yPtr = &y;
```



```
int x=1, y=2, z[10];  
int * ip; //ip è un puntatore ad un intero  
ip = &x; //ora ip punta a x  
y= *ip //ora y vale 1  
*ip = 0; // ora x vale 0  
ip = &z[0] // ora ip punta a z[0]  
*ip = *ip +10 //incrementa *ip e cioè z[0] di 10  
y = *ip +1; // assegna a y il valore di z[0] più 1  
*ip += 1; //incrementa di uno l'oggetto puntato da ip e cioè z[0]  
(*ip)++; // incrementa di uno l'oggetto puntato da ip e cioè z[0]
```



```
1    struct point{
2        float x;
3        float y;
4    };
5
6    struct rectangle{
7        struct point p1;
8        struct point p2;
9    };
10
11    struct rectangle r, * rp = &r;
12
13    // forme equivalenti di dire la stessa cosa
14    r.p1.x;
15    rp->p1.x;
16    (*rp).p1.x;
```

Passare il un puntatore anziché un valore è inoltre un processo che risparmia di gran lunga sul tempo

Anche un array è un tipo di puntatore