



**UNIVERSITÀ
DI PARMA**

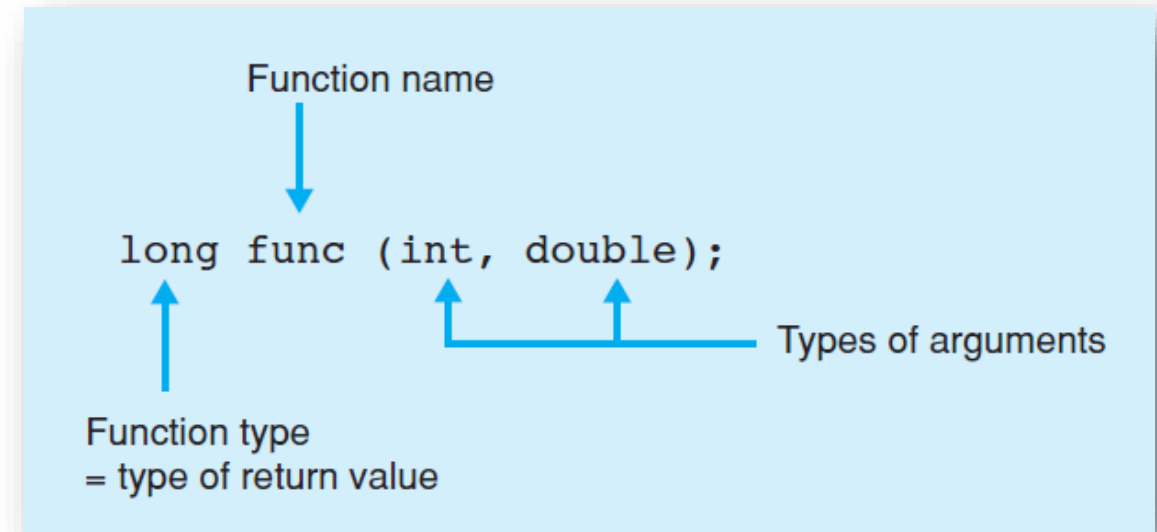
C++ funzioni

Alberto Ferrari

<http://en.cppreference.com>

FUNZIONI

- caratterizzate da *nome*, *parametri* (numero, ordine e tipo) e *tipo* di ritorno
- le funzioni hanno un prototipo
- il prototipo non è necessario se la definizione della funzione appare prima del suo utilizzo
- nel prototipo i parametri possono non avere nome, ma per chiarezza in genere lo si mette



- la direttiva **#include** permette di importare i prototipi di funzioni delle librerie standard
- ogni libreria standard ha un file *header* contenente la definizione di funzioni, di tipo di dati e di costanti
- **#include <cmath>**
 - per utilizzare funzioni matematiche

```
double sin (double);           // Sine
double cos (double);          // Cosine
double tan (double);          // Tangent
double atan (double);         // Arc tangent
double cosh (double);         // Hyperbolic Cosine
double sqrt (double);         // Square Root
double pow (double, double);  // Power
double exp (double);          // Exponential Function
double log (double);          // Natural Logarithm
double log10 (double);        // Base-ten Logarithm
```

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double y, x = 5.22;
    // la funzione pow ha prototipo double pow( double, double)
    // y = pow("x", 3.0); // error: no matching function
                          // for call to 'pow(const char [2], double)'
    // y = pow(x + 3.0); // error: no matching function for call to 'pow(double)'
    y = pow(x, 3.0);      // ok!
    y = pow(x, 3);        // ok! Il compilatore converte l'intero 3 in double
    cout << x << " elevato al cubo vale: " << y << endl;
    return 0;
}
```

```
#include <iostream>
using namespace std;
double media(int v1, int v2);
int main() {
    int val1, val2;
    cout << "Inserire due valori interi separati da spazio ";
    cin >> val1 >> val2;
    cout << "la media aritmetica fra " << val1 << " e " << val2
          << " = " << media(val1,val2) << endl;

    return 0;
}
double media(int v1, int v2) {
    double med;
    med = (v1 + v2) / 2.0;
    return med;
}
```

- call-by-value
 - l'argomento può essere una variabile o un'espressione
 - il parametro formale viene inizializzato al valore dell'argomento
 - la funzione riceve una copia del valore dell'argomento
 - le azioni sui parametri formali non si ripercuotono sui parametri attuali
- call-by-reference (*"&" precede il tipo del parametro formale*)
 - l'argomento deve essere una variabile
 - il parametro formale viene sostituito con la variabile argomento
 - le azioni sui parametri formali si ripercuotono sui parametri attuali
 - vantaggio: migliori prestazioni
 - svantaggio: minore modularità, la funzione chiamata può corrompere i dati della chiamante (*side effects*)

- call-by-pointer
 - l'indirizzo del parametro attuale viene copiato nel parametron formale
 - all'interno della funzione l'indirizzo è utilizzato per accedere al dato "puntato" dal parametron attuale
 - i parametri formali devono essere definiti come puntatori ai quali assegnare gli indirizzi dei parametri attuali
 - le azioni sui parametri formali si ripercuotono sui parametri attuali


```
void scambiaVal(int a, int b) {  
    int temp;  
    temp = a;  
    a = b;  
    b = temp;  
}  
  
int main() {  
    ...  
    int x,y; x = 7; y = 5;  
    cout << "scambio i valori x = " << x << " y = " << y << endl;  
    cout << "passaggio per valore : ";  
    scambiaVal(x,y);  
    cout << "x = " << x << " y = " << y << endl;  
    ...  
}
```

```
scambio i valori x = 7 y = 5  
passaggio per valore : x = 7 y = 5
```

```
void scambiaRef(int &a, int &b) {  
    int temp;  
    temp = a;  
    a = b;  
    b = temp;  
}  
  
int main() {  
    ...  
    cout << "scambio i valori x = " << x << " y = " << y << endl;  
    cout << "passaggio per riferimento : ";  
    scambiaRef(x,y);  
    cout << "x = " << x << " y = " << y << endl; ...  
}
```

```
scambio i valori x = 7 y = 5  
passaggio per riferimento : x = 5 y = 7
```

```
void scambiaInd(int *a, int *b) {  
    int temp;  
    temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
int main() {  
    ...  
    cout << "scambio i valori x = " << x << " y = " << y << endl;  
    cout << "passaggio per indirizzo : ";  
    scambiaInd(&x, &y);  
    cout << "x = " << x << " y = " << y << endl;  
    ...  
}
```

```
scambio i valori x = 7 y = 5  
passaggio per indirizzo : x = 5 y = 7
```