

Le funzioni in JavaScript

Le Funzioni: Cos'è una Funzione?

Una funzione è come una "ricetta" che possiamo riutilizzare nel nostro codice:

- È un blocco di codice che esegue un compito specifico
- Ha un nome che descrive cosa fa
- Può essere usata tutte le volte che vogliamo

Vantaggi delle Funzioni?

- **Riutilizzo del codice:** Scriviamo il codice una volta e lo usiamo molte volte
- **Organizzazione:** Il codice è più ordinato e facile da capire
- **Manutenzione:** Se dobbiamo fare modifiche, le facciamo in un solo punto

Come Funziona una Funzione?

È come una scatola magica:

1. **Input:** Diamo alla funzione dei valori
2. **Elaborazione:** La funzione fa qualcosa con questi valori
3. **Output:** La funzione ci restituisce un risultato



Il Nostro Primo Esempio

```
// Definiamo una funzione che calcola l'area di un rettangolo
function calcolaArea(base, altezza) {
  area = base * altezza;
  return area;
}

// Usiamo la funzione
let area1 = calcolaArea(5, 3); // area = 15
let area2 = calcolaArea(10, 4); // area = 40

console.log("L'area del primo rettangolo è:", area1);
console.log("L'area del secondo rettangolo è:", area2);
```

Anatomia di una Funzione

```
function nomeFunzione(parametro1, parametro2) {  
    // Il codice della funzione va qui  
    // Facciamo qualcosa con i parametri  
  
    return risultato; // Restituiamo un valore  
}
```

- `function` : parola chiave per creare una funzione
- `nomeFunzione` : nome che scegliamo noi
- `parametri` : valori che la funzione riceve
- `return` : valore che la funzione restituisce

Esempio Pratico: Funzione per il Saluto

```
function saluta(nome) {  
    return "Ciao " + nome + "!";  
}  
  
// Usiamo la funzione  
let messaggio1 = saluta("Mario");  
let messaggio2 = saluta("Luigi");  
  
console.log(messaggio1); // Stampa: Ciao Mario!  
console.log(messaggio2); // Stampa: Ciao Luigi!
```

Esercizio: Creiamo una Funzione

Scriviamo insieme una funzione che calcola il perimetro di un rettangolo:

Funzione per il Perimetro

```
// Definiamo la funzione
function calcolaPerimetro(base, altezza) {
  return 2 * (base + altezza);
}

// Proviamo la funzione con diversi valori
let perimetro1 = calcolaPerimetro(5, 3);
let perimetro2 = calcolaPerimetro(10, 4);

console.log("Perimetro 1:", perimetro1); // Perimetro 1: 16
console.log("Perimetro 2:", perimetro2); // Perimetro 2: 28
```

Invocazione di funzioni

Come abbiamo visto nella slide precedente una funzione può essere chiamata (invocata) scrivendo due parentesi alla fine del nome della variabile che la contiene.

Eventuali valori inclusi tra parentesi, chiamati **argomenti**, sono disponibili per il codice della funzione stessa.

Esempio:

```
alert("Buongiorno!"); // La stringa "Buongiorno!" è l'argomento
```

```
calcolaPerimetro(5, 3); // 5 e 3 sono gli argomenti
```

Valori di ritorno

Le funzioni possono:

- Produrre effetti sullo schermo (come `alert`)
- Restituire un valore di ritorno

Ad esempio la funzione `Math.max` prende due numeri e restituisce il maggiore:

```
console.log(Math.max(2, 4));  
// → 4
```

Una funzione che produce un valore è una espressione e può essere utilizzata in un contesto più complesso:

```
console.log(Math.min(2, 4) + 100);  
// → 102
```

La funzione console.log

- Disponibile in molti browser moderni
- Permette di mostrare a video il valore di espressioni
- Il testo è mostrato nella Javascript console
- Di solito questa parte del browser è nascosta:
 - F12 in molti sistemi
 - Command-Option-I su Mac OS
 - Cerca "web console" o "developer tools" nel menu

Esercizio: Area e perimetro

Scrivere un programma che calcoli l'area e il perimetro di un quadrato dato il lato:

```
var a = prompt("Dammi la lunghezza del lato");  
var area = a * a;  
var perimetro = a * 4;  
  
console.log("L'area del quadrato è " + area);  
console.log("Il perimetro del quadrato è " + perimetro);
```

08. Funzioni in JavaScript

```
function calcolaArea(lato) {  
    return lato * lato;  
}  
  
function calcolaPerimetro(lato) {  
    return lato * 4;  
}  
  
var a = prompt("Dammi la lunghezza del lato");  
var area = calcolaArea(a);  
var perimetro = calcolaPerimetro(a);  
  
console.log("L'area del quadrato è " + area);  
console.log("Il perimetro del quadrato è " + perimetro);
```

Variabili globali e locali

- Una **variabile globale** è visibile in tutto il programma
 - A meno che non venga mascherata
- Una **variabile locale** è visibile solo in alcuni punti del programma
 - Ad esempio all'interno di una funzione

Name masking (shadowing)

- Una variabile globale può essere nascosta da una variabile locale che ha lo stesso identificatore.
- Una variabile nascosta non è accessibile nel blocco dove è stata mascherata.

```
var x = 10; // globale

function esempio() {
  var x = 20; // locale, maschera la globale
  console.log(x); // stampa 20
}


console.log(x); // stampa 10
```


Un po' di pratica!

Esercizio 1: La Funzione Termometro

Obiettivo: Scrivi una funzione `controllaTemperatura` che prende come parametro una temperatura e restituisce:

- "Troppo freddo!" se la temperatura è sotto i 15 gradi
- "Temperatura ok" se è tra 15 e 25 gradi
- "Troppo caldo!" se è sopra i 25 gradi

 **Suggerimento:** Usa le istruzioni condizionali (if) per controllare i diversi intervalli di temperatura


08. Funzioni in JavaScript

```
function controllaTemperatura(temp) {  
    // Controlliamo prima il caso del freddo  
    if (temp < 15) {  
        return "Troppo freddo!";  
    }  
    // Poi controlliamo il caso del caldo  
    else if (temp > 25) {  
        return "Troppo caldo!";  
    }  
    // Se non è né troppo freddo né troppo caldo...  
    else {  
        return "Temperatura ok";  
    }  
}  
  
// Proviamo la funzione con diversi valori  
console.log(controllaTemperatura(10)); // "Troppo freddo!"  
console.log(controllaTemperatura(20)); // "Temperatura ok"  
console.log(controllaTemperatura(30)); // "Troppo caldo!"
```

Esercizio 2: Il Calcolatore del Prezzo

Obiettivo: Crea una funzione `calcolaPrezzo` che:

- Prende come parametri il prezzo base e l'età del cliente
- Applica uno sconto del 20% per clienti sotto i 18 anni
- Applica uno sconto del 40% per clienti sopra i 65 anni
- Restituisce il prezzo finale

 **Suggerimento:** Ricorda che per calcolare uno sconto del 20% puoi moltiplicare per 0.8


08. Funzioni in JavaScript

```
function calcolaPrezzo(prezzoBase, eta) {  
  // Iniziamo assumendo che non ci sia sconto  
  let prezzoFinale = prezzoBase;  
  
  // Controlliamo se il cliente è giovane  
  if (eta < 18) {  
    prezzoFinale = prezzoBase * 0.8; // Sconto del 20%  
  }  
  // Controlliamo se il cliente è senior  
  else if (eta >= 65) {  
    prezzoFinale = prezzoBase * 0.6; // Sconto del 40%  
  }  
  
  return prezzoFinale;  
}  
  
// Proviamo la funzione  
console.log(calcolaPrezzo(100, 15)); // 80 (sconto giovani)  
console.log(calcolaPrezzo(100, 30)); // 100 (nessuno sconto)  
console.log(calcolaPrezzo(100, 70)); // 60 (sconto senior)
```

Esercizio 3: Il Contatore di Vocali

Obiettivo: Scrivi una funzione `contaVocali` che:

- Prende come parametro una stringa
- Conta quante vocali (a,e,i,o,u) ci sono nella stringa
- Restituisce il numero totale di vocali

 **Suggerimento:** Puoi trasformare la stringa in minuscolo con `toLowerCase()` per semplificare il controllo


08. Funzioni in JavaScript

```
function contaVocali(testo) {  
  // Convertiamo il testo in minuscolo per semplificare  
  let testoMinuscolo = testo.toLowerCase();  
  
  // Inizializziamo il contatore  
  let numeroVocali = 0;  
  
  // Controlliamo ogni carattere  
  for (let i = 0; i < testoMinuscolo.length; i++) {  
    let carattere = testoMinuscolo[i];  
    if (carattere === 'a' || carattere === 'e' ||  
        carattere === 'i' || carattere === 'o' ||  
        carattere === 'u') {  
      numeroVocali++;  
    }  
  }  
  
  return numeroVocali;  
}  
  
// Proviamo la funzione  
console.log(contaVocali("Ciao")); // 3  
console.log(contaVocali("JavaScript")); // 2  
console.log(contaVocali("AEIOU")); // 5
```

Esercizio 4: Il Generatore di Iniziali

Obiettivo: Crea una funzione `generalIniziali` che:

- Prende nome e cognome come parametri
- Restituisce le iniziali in maiuscolo seguite da punti
- Esempio: "Mario Rossi" → "M.R."

 **Suggerimento:** Puoi usare il metodo `charAt(0)` per ottenere il primo carattere di una stringa


```
function generalIniziali(nome, cognome) {  
    // Prendiamo la prima lettera del nome e la rendiamo maiuscola  
    let inizialeNome = nome.charAt(0).toUpperCase();  
  
    // Prendiamo la prima lettera del cognome e la rendiamo maiuscola  
    let inizialeCognome = cognome.charAt(0).toUpperCase();  
  
    // Combiniamo le iniziali con i punti  
    return inizialeNome + "." + inizialeCognome + ".";  
}  
  
// Proviamo la funzione  
console.log(generalIniziali("mario", "rossi")); // "M.R."  
console.log(generalIniziali("anna", "verdi")); // "A.V."  
console.log(generalIniziali("luigi", "bianchi")); // "L.B."
```