

FONDAMENTI DI PROGRAMMAZIONE

Giovanni Pellegrini

JAVASCRIPT

STRUTTURA DEL CODICE









2024

Instant Book 2

INSTANT BOOK 3

STRUTTURA DEL CODICE

1.  **ISTRUZIONI** pag. 3
2.  **PUNTO E VIRGOLA**
3.  **COMMENTI** pag. 4
-  **Informazioni**
-  **Annotazioni** pag. 5
4.  **ESERCIZI**

● ISTRUZIONI

Le istruzioni sono dei costrutti sintattici e comandi che permettono di eseguire azioni.

Abbiamo già visto un'istruzione `alert('Hello, world!')`, che mostra il messaggio "Hello world!".

All'interno del codice possiamo avere tutte le istruzioni che desideriamo. Le istruzioni possono essere separate da un punto e virgola.

Ad esempio, qui dividiamo il messaggio in due `alert`:

```
alert('Hello'); alert('World');
```

Di solito ogni istruzione viene scritta in una riga separata per rendere il codice molto più leggibile:

```
alert('Hello');  
alert('World');
```

● PUNTO E VIRGOLA

Un punto e virgola può essere omesso nella maggior parte dei casi quando si interrompe una riga.

Questo funzionerà ugualmente:

```
alert('Hello')  
alert('World')
```

In questo caso, JavaScript interpreta la fine della riga come un punto e virgola "implicito". Viene anche chiamata inserimento automatico del punto e virgola.

In molti casi la nuova riga viene interpretata come un punto e virgola implicito. Ma "in molti casi" non significa "sempre"!

Ci sono casi in cui la nuova riga non implica una punto e virgola, per esempio:

```
alert(3 +  
1  
+ 2);
```

Il codice stampa 6 perché, in questo caso, JavaScript non inserisce un punto e virgola. E' abbastanza ovvio che se la riga finisce con un "+", allora è un "espressione incompleta", quindi il punto e virgola sarebbe errato. Per questo, nell'esempio sopra, tutto funziona come dovrebbe.

E' consigliato quindi, di inserire il punto e virgola fra ogni istruzione, anche se vengono scritte in righe diverse. Questa è una regola largamente adottata dalla community. Ripetiamolo nuovamente : è possibile omettere il punto e virgola la maggior parte delle volte, ma è più sicuro, specialmente per un novizio, inserirlo al termine di ogni istruzione.

● COMMENTI

Con il passare del tempo, i programmi sono diventati sempre più complessi. E' diventato, pertanto, necessario aggiungere commenti per poter descrivere i comportamenti del codice.

I commenti possono essere messi in qualsiasi punto dello script. Infatti non hanno alcun effetto sull'esecuzione del codice, poiché il motore JavaScript semplicemente li ignora.

I commenti su una singola linea si inseriscono con due caratteri di slash //.

Il resto della linea è il commento. Può occupare un'intera linea oppure essere posta in seguito ad un'istruzione.

Vediamo un esempio:

```
// Questo commento occupa un'intera riga  
alert('Hello');  
  
alert('World'); // Questo commento segue un'istruzione
```

I commenti multilinea incominciano con un singolo carattere di slash ed un asterisco /* e finiscono con un asterisco ed un carattere di slash */.

Come nell'esempio:

```
/* Un esempio con due messaggi.  
Questo è un commento multilinea.  
*/  
alert('Hello');  
alert('World');  
Il contenuto dei commenti viene ignorato, quindi se inseriamo codice al suo interno  
/* ... */ non verrà eseguito.
```

Qualche volta può essere utile per bloccare temporaneamente qualche porzione di codice:

```
/* Commentiamo il codice  
alert('Hello');  
*/  
alert('World');
```



Usate le scorciatoie da tastiera !

In molti editor una linea di codice può essere commentata con la combinazione da tastiera dei tasti Ctrl+Shift+/
Per i commenti multilinea (selezionate prima una parte di codice e poi utilizzate la combinazione di tasti).

Su Mac utilizzate la combinazione Cmd al posto di Ctrl.



I commenti annidati non sono supportati!

Non si possono inserire `/*...*/` all'interno di altri `/*...*/`.

Questo codice genererebbe un errore:

```
/*  
/* commento annidato ?? */  
*/  
alert( 'World' );
```

Non abbiate paura di utilizzare i commenti nel codice.

I commenti aumenteranno il peso finale dello script, ma questo non sarà un problema. Ci sono tantissimi strumenti che possono minimizzare(“minify”) il codice prima di pubblicarlo nel server. Questi strumenti rimuovono i commenti, quindi non appariranno nello script che verrà eseguito. Perciò i commenti non hanno alcun effetto negativo sul codice prodotto.



ESERCIZI

1. Crea un commento

Riduci il seguente script a commento multilinea

```
alert('BENVENUTI');  
alert('ALL'ITS ACADEMY CUCCOVILLO');
```