JAVASCRIPT

Cristian Carrino

ENGIM PIEMONTE
Formazione Orientamento
Cooperazione Lavoro

IMPLEMENTAZIONE DEL CODICE

Cristian Carrino

ENGIM
PIEMONTE
Formazione Orientamento
Cooperazione Lavoro

Implementare (cioè realizzare, inserire) il codice Javascript all'interno di una pagina HTML può essere fatto in due modi:

- scrivendolo direttamente nella pagina html dentro il tag <script>...</script>
- scrivendolo in un file esterno e poi importando quel file nella pagina html

index.html > ...

```html
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Javascript 01</title>
</head>
<body>
    <h1>Guarda la console del browser</h1>

    <script>
        // definizioni variabili
        const A = 13;
        const B = 4;
        var result;
        console.log('A vale: ' + A, 'B vale: ' + B);

        // somma
        result = A + B;
        console.log('Il risultato di ' + A + ' + ' + B + ' è: ' + result);

        // sottrazione
        result = A - B;
        console.log('Il risultato di ' + A + ' - ' + B + ' è: ' + result);

        // moltiplicazione
        result = A * B;
        console.log('Il risultato di ' + A + ' * ' + B + ' è: ' + result);

        // divisione
        result = A / B;
        console.log('Il risultato di ' + A + ' / ' + B + ' è: ' + result);

        // resto dellA divisione
        result = A % B;
        console.log('Il risultato di ' + A + ' % ' + B + ' è: ' + result);
    </script>
</body>
</html>
```

Dentro il tag <script>, posso scrivere codice Javascript.

Il tag <script> va aperto e chiuso:

```
<script>
/*
codice Javascript
*/
</script>
```

In un file esterno includendo il file attraverso il tag <script src="..."></script>

index.html > ...

```html
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Javascript 02</title>

    <script src="app.js"></script>
</head>
<body>
    <h1>Guarda la console del browser</h1>
</body>
</html>
```

app.js > ...

```js
// definizioni variabili
const NUMERO = 4;

// somma
var a = 9;
console.log("a vale: " + a);

a += NUMERO;
console.log("Dopo aver fatto 'a += " + NUMERO + "', adesso a vale: " + a + "\n\n");

// sottrazione
var b = 6;
console.log("b vale: " + b);

b -= NUMERO;
console.log("Dopo aver fatto 'b -= " + NUMERO + "', adesso b vale: " + b + "\n\n");

// moltiplicazione
var c = 3;
console.log("c vale: " + c);

c *= NUMERO;
console.log("Dopo aver fatto 'c *= " + NUMERO + "', adesso c vale: " + c + "\n\n");

// divisione
var d = 12;
console.log("d vale: " + d);

d /= NUMERO;
console.log("Dopo aver fatto 'd /= " + NUMERO + "', adesso d vale: " + d + "\n\n");

// resto della divisione
var e = 10;
console.log("e vale: " + e);

e %= NUMERO;
console.log("Dopo aver fatto 'e %= " + NUMERO + "', adesso e vale: " + e);
```

Cristian Carrino

CONDIZIONI

IF, ELSE, ELSE IF, SWITCH

Cristian Carrino

ENGIM PIEMONTE
Formazione Orientamento
Cooperazione Lavoro

Gli operatori

- if (in italiano: se)
- else (in italiano: altrimenti)
- else if (in italiano: altrimenti se)

servono per eseguire o un gruppo di istruzioni oppure un altro, in base al valore di una variabile booleana (o di una espressione)

Esempio di if

```
> var condizione = true; // oppure false

  if (condizione) {
      /*
      le istruzioni qui dentro vengono
      eseguite solo quando condizione
      è vera
      */
  }
```

Esempio di if, else

```
> var condizione = false; // oppure false

  if (condizione) {
      /*
      le istruzioni qui dentro vengono
      eseguite solo quando condizione
      è true
      */
  } else {
      /*
      le istruzioni qui dentro vengono
      eseguite solo quando condizione
      è false
      */
  }
```

Esempio di if, else if, else

```
> var condizione1 = false;
  var condizione2 = true;

  if (condizione1) {
      /*
      le istruzioni qui dentro vengono eseguite solo quando condizione1 è true
      */
  } else if (condizione2) {
      /*
      le istruzioni qui dentro vengono eseguite solo quando condizione1 è false e condizione2 è true
      */
  } else {
      /*
      le istruzioni qui dentro vengono eseguite solo quando sia condizione1 che condizione2 sono false
      */
  }
```

NB: Qualsiasi numero (anche negativo) è considerato true.
Tranne lo 0 che è l'unico numero considerato false.

```
> if (2) {
      console.log('E\' true');
  } else {
      console.log('E\' false');
  }
  E' true
```

```
> if (-2) {
      console.log('E\' true');
  } else {
      console.log('E\' false');
  }
  E' true
```

```
> if (0) {
      console.log('E\' true');
  } else {
      console.log('E\' false');
  }
  E' false
```

NB: Qualsiasi stringa è true.
Tranne la stringa vuota che è false.

```
> if ('Ciao') {
      console.log('E\' true');
  } else {
      console.log('E\' false');
  }
  E' true
```

```
> if ('') {
      console.log('E\' true');
  } else {
      console.log('E\' false');
  }
  E' false
```

NB: null è false.

```
> if (null) {
      console.log('E\' true');
  } else {
      console.log('E\' false');
  }
  E' false
```

Lo switch è un operatore speciale che riassume una serie lunga di if, else if, ... , else.

```
> var valore = 10;

  switch (valore) {
      case 'Ciao':
          console.log('Il valore è la stringa "Ciao"');
          break;
      case 4:
          console.log('Il valore è il numero 4');
          break;
      case 10:
          console.log('Il valore è il numero 10');
          break;
      case null:
          console.log('Il valore è null');
          break;
      case 4:
          console.log('Il valore è false');
          break;
      default:
          console.log('Il valore non è in nessuno dei casi precedenti');
  }
  Il valore è il numero 10
```

L’istruzione break; non è obbligatoria. Semplicemente se non la mettiamo, il codice continuerà anche le istruzioni successive.

```
> var valore = 10;

  switch (valore) {
      case 'Ciao':
          console.log('Il valore è la stringa "Ciao"');
      case 4:
          console.log('Il valore è il numero 4');
      case 10:
          console.log('Il valore è il numero 10');
      case null:
          console.log('Il valore è null');
      case 4:
          console.log('Il valore è false');
      default:
          console.log('Il valore non è in nessuno dei casi precedenti');
  }
  Il valore è il numero 10
  Il valore è null
  Il valore è false
  Il valore non è in nessuno dei casi precedenti
```

Se esistono più case che hanno stesso risultato (cioè che devono eseguire le stesse istruzioni), si possono riunire come in figura, mettendo tutti i case uno sotto l'altro.

```
> var mese = 'Aprile';

  switch (mese) {
      case 'Gennaio':
      case 'Marzo':
      case 'Maggio':
      case 'Luglio':
      case 'Settembre':
      case 'Novembre':
          console.log('mese è un mese dispari');
          break;
      case 'Febbraio':
      case 'Aprile':
      case 'Giugno':
      case 'Agosto':
      case 'Ottobre':
      case 'Dicemrbe':
          console.log('mese è un mese pari');
          break;
      default:
          console.log('Il mese inserito non esiste');
  }

  mese è un mese pari
```

NB: il default non è obbligatorio. Semplicemente se non lo mettiamo, e nessuno dei case precedenti è soddisfatto, lo switch non eseguirà nessuna istruzione.

```
> var mese = 'Ciao';

  switch (mese) {
      case 'Gennaio':
      case 'Marzo':
      case 'Maggio':
      case 'Luglio':
      case 'Settembre':
      case 'Novembre':
          console.log('mese è un mese dispari');
          break;
      case 'Febbraio':
      case 'Aprile':
      case 'Giugno':
      case 'Agosto':
      case 'Ottobre':
      case 'Dicemrbe':
          console.log('mese è un mese pari');
          break;
  }
```

L’OPERATORE TERNARIO

Cristian Carrino

ENGIM
PIEMONTE
Formazione Orientamento
Cooperazione Lavoro

Questi due pezzi di codici fanno esattamente la stessa cosa:

```
> var condizione = true;

  if (condizione) {
      var a = 5;
  } else {
      var a = 10;
  }

  console.log('a = ' + a);
  a = 5
```

```
> var condizione = true;

  var a = condizione ? 5 : 10;

  console.log('a = ' + a);
  a = 5
```

Solo che quello a destra è molto più compatto.

La sintassi è:
condizione ? valore per il true : valore per il false
Se condizione è true, verrà restituito il valore dopo il ?, se è false verrà restituito il valore dopo il :.

```
> var oggi = 10;

  var testo = 'Oggi è un giorno ' + (oggi % 2 == 0 ? 'pari' : 'dispari');

  console.log(testo);
  Oggi è un giorno pari
```

ITERAZIONI

FOR, WHILE, DO-WHILE

Cristian Carrino

ENGIM PIEMONTE
Formazione Orientamento
Cooperazione Lavoro

Il ciclo for serve a ripetere un gruppo di istruzioni un numero determinato di volte.
Si devono impostare:

- una variabile che funziona da contatore
- una condizione che determina quando finirà il ciclo
- un valore di incremento/decremento della variabile

```
> var i; // index
  for (i = 0; i < 10; i++) {
      console.log('i = ' + i);
  }
  i = 0
  i = 1
  i = 2
  i = 3
  i = 4
  i = 5
  i = 6
  i = 7
  i = 8
  i = 9
```

Il ciclo while esegue un gruppo di istruzioni finché la sua condizione è vera.

NB: Dobbiamo far in modo che la sua condizione dopo un po' diventi falsa o entreremo in un loop infinito.

NB: il codice nell'esempio è esattamente identico al for() di prima

```
> var i = 0;
  while (i < 10) {
      /* istruzioni eseguite finchè
         la condizione del while
         è vera */
      console.log('i = ' + i);
      i++;
  }
  i = 0
  i = 1
  i = 2
  i = 3
  i = 4
  i = 5
  i = 6
  i = 7
  i = 8
  i = 9
```

Il ciclo do-while è identico al while tranne che per una piccola ma decisiva differenza: come prima, l'istruzione dentro il do viene eseguita finché la condizione del while è vera ma viene eseguita comunque almeno una volta perché il controllo della condizione viene fatto dopo.

```
> var i = 0;
  do {
      /* questa istruzione viene
         eseguita almeno una volta */
      console.log('i = ' + i);
      i++;
  }
  while (i < 10);
  i = 0
  i = 1
  i = 2
  i = 3
  i = 4
  i = 5
  i = 6
  i = 7
  i = 8
  i = 9
```

Esempio di differenza tra while e do-while

```
> var i = 20;
  while (i < 10) {
      if (i >= 10) {
          i = 0;
      }
      console.log('i = ' + i);
      i++;
  }
```

```
> var i = 20;
  do {
      if (i >= 10) {
          i = 0;
      }
      console.log('i = ' + i);
      i++;
  }
  while (i < 10);
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
```

BREAK e CONTINUE

Cristian Carrino

ENGIM
PIEMONTE
Formazione Orientamento
Cooperazione Lavoro

Utilizzo:

- break; termina il ciclo (o lo switch)
- continue; passa alla iterazione successiva

```
> var i;
  for (i = 0; i < 10; i++) {
      if (i == 3) {
          break;
      }
      console.log('i = ' + i);
  }
  i = 0
  i = 1
  i = 2
```

```
> var i;
  for (i = 0; i < 10; i++) {
      if (i == 3) {
          continue;
      }
      console.log('i = ' + i);
  }
  i = 0
  i = 1
  i = 2
  i = 4
  i = 5
  i = 6
  i = 7
  i = 8
  i = 9
```