Mauro Bogliaccino

Corso Javascript

- Convenzioni del linguaggio e dialogs
- Variabili
- Operatori
- Costrutti condizionali e iterativi
- Tipi primitivi
- Tipi reference
- Boolean
- Function
- Number
- String
- Array
- Object
- Timers

Fondamenti del linguaggio

Javascript è un linguaggio di scripting (supporta gli script) per le pagine Web, ma viene utilizzato anche in ambienti diversi dal browser. È un linguaggio di scripting potente, leggero, interpretato, con funzioni di prima classe (ovvero il linguaggio supporta il passaggio di funzioni come argomenti ad altre funzioni).

Per aggiungere interattività dinamica alle pagine Web, Javascript è incorporato in Hypertext Markup Language (HTML). Poiché funziona sul lato client del Web, può essere utilizzato per progettare/programmare il comportamento delle pagine Web al verificarsi di un particolare evento. Questo è il motivo per cui è ampiamente utilizzato per il comportamento delle pagine web.

JavaScript è un linguaggio di programmazione

Javascript è utilizzato comunemente come parte dell'esperienza di navigazione, permette di creare interazioni con l'utente, controllare la navigazione, gestire la comunicazione asincrona, e modificare il contenuto del documento.

Alcune cose che puoi fare con javascript

- Generare finestre di dialogo
- redirigere una pagina
- aprire nuove finestre
- intercettare eventi dell'utente: mouse, tastiera, ...
- leggere i moduli dell'utente

- modificare le pagine html . ...e molto altro!

___Fonte: https://en.wikipedia.org/wiki/JavaScript___

---

ruolo di js nel web design

- Html: contenuto
- Css: presentazione
- Js: comportamento

Risorse disponibili online

- ES2015 - ecma
- Mozilla Dev - Javascript reference
- JS the right way
- You-Dont-Know-JS
- w3schools
- html.it
- it.html.net tutorials - html - css
- web-link.it/web/html5

Fondamenti del linguaggio

- Struttura sintattica
- Tipi, valori e variabili
- Espressioni e operatori
- Statements - istruzioni
- Oggetti
- Arrays
- Funzioni
- Classi e moduli

Hello, World

Hello, World! nel browser

```
alert('Hello, World!')
```

Hello, World! nel terminale

```
console.log('Hello, World!')
```

Esecuzione del codice

- Interpretazione in una pagina web
- Interpretazione in una console del browser
- Interpretazione in terminale (node.js)

Javascript incorporato in una pagina html

```
<!doctype html>
<html>
    <head></head>
    <body>
        <script>
            alert('Hello world')
        </script>
    </body>
</html>
```

Javascript collegato ad una pagina html

```
<!doctype html>
<html>
    <head></head>
    <body>
        <script src="esempio.js"></script>
    </body>
</html>
```

```
// file esempio.js
alert('Hello world')
```

Come mostrare a video i dati

- window
    - window.alert()
    - window.prompt()
    - window.confirm()
- document
    - document.write()

    - document.writeln()
- innerHTML
- console.log()

Convenzioni di naming

- JavaScript è **CASE SENSITIVE**
- Terminare le istruzioni con punto e virgola (😉
- NomeClasse, nomeFunzione, nomeVariabile, nome_variabile, NOME_COSTANTE

Identificatori

Gli identificatori in javascript cominciano con

- una lettera,
- una underline (_),
- un carattere di dollaro ($);

seguito da

- lettere,
- numeri,
- underline,
- $

Per esempio

```
var contatore;
var _indice;
var $indice;
var $__$__$;
```

Parole chiave riservate

abstract, boolean, break, byte, case, catch, char, class, const, continue, debugger, default, delete, do, double, else, enum, export, extends, false, final, finally, float, for, function, goto, if, implements, import, in, instanceof, int, interface, let, long, native, new, null, package, private, protected, public, return, short, static, super, switch, synchronized, this, throw, throws, transient, true, try, typeof, var, void, volatile, while, with

---

Variabili

- identificatore e visibilità (scope)
- dichiarazione e inizializzazione di variabili

- esempi

---

const

Parola chiave per la dichiarazione di una costante. esempio

var, let

Parola chiave per la dichiarazione di una variabile.

---

operatori

- precedenza degli operatori
- operatori di assegnamento
- operatori di comparazione
- operatori unari
- operatori logici AND
- operatori logici OR
- operatori su stringhe

Commenti

Esistono due tipi di commenti:

```
//  commento su una riga

//puoi usarlo per commentare un'istruzione:
var a = 5; //assegno la variabile

/*
    commento
    su diverse
    righe
*/
```

---

use strict

Direttiva per l'interprete di JavaScript, che indica l'uso del modo strict.

Literals

tipo	esempio
Numeri interi	192

tipo	esempio
Numeri float	1.4
Stringhe di testo	"Hello World!", 'Hello World!'
Valori logici	true, false
espressioni regolari	/[A-Za-z]/
Valore nullo	null
Valore undefined	undefined

Strutture del linguaggio

Costrutti di controllo del flusso

- Strutture condizionali
  - esempio if
  - esempio switch

Costrutti di iterazione (cicli)

- For Loop
  - esempi
- While Loop
  - esempi

Tipi di dato

Boolean

- Boolean è la rappresentazione di tipo oggetto di una variabile logica.
- Booleans
  - operazioni logiche
  - comparazione tra numeri e valori booleani
  - undefined e null
  - Boolean() verifica se un'espressione è booleana

```
//Valori logici
var a=true
var b=false
```

Sono valori falsi i seguenti

```
undefined
null
0
-0
NaN
''
```

Altri esempi

- esempio boolean
- esempio AND Logico
- esempio OR Logico
- esempio OR Logico

Objects

- JS Objects

instanceof

Verifica se un oggetto è instanza di qualche prototipo.

---

typeof

Ritorna una stringa indicante il tipo di dato che ha una variabile. esempio

delete

Operatore che rimuove proprietà di un oggetto.

Number

Number è la rappresentazione di tipo oggetto di un tipo numerico.

JS numbers

- i numeri in JS sono SEMPRE float a 64-bit
- il numero massimo di decimali è 17 e la virgola mobile non è sempre accurata
- il prefisso 0x permette di usare i numeri esadecimali
- In JS esiste Infinity e -Infinity
- NaN not a number: p.es. operazioni aritmetiche con le stringhe restituiscono NaN

---

proprietà e metodi principali

```
* Number()
* parseFloat()
* parseInt()
* toString()
* toFixed()
* toPrecision()
* valueOf()
```

non dichiarare stringhe, numeri e booleans come oggetti

Per esempio

```
  12 // numero intero in base decimale.
0345 // numero intero in base ottale.
0xFF // numero intero in base esadecimale.

3.141592654 // numero decimale.
.234955     // numero decimale.
6.023e23    // numero decimale in notazione esponenziale.
```

Math Object

Per lavorare con i Number, puoi usare Math che è l'oggetto che concentra molte constanti e funzioni matematiche.

- Math Object

String

String è la rappresentazione di tipo oggetto di una stringa.

- String Object

---

valore nullo e valore undefined

Rappresentano l'assenza di un valore in una variabile o nel ritorno di una function.

```
var a=null
var b=undefined
```

In JavaScript, `null` e `undefined` sono due valori speciali che rappresentano la mancanza di valore o dati non definiti, ma hanno differenze chiave:

1. **`undefined`**: Questo valore viene assegnato automaticamente a una variabile che è stata dichiarata ma non inizializzata con un valore. In altre parole, se hai dichiarato una variabile ma non le hai assegnato alcun valore, il suo valore sarà `undefined`. Ad esempio:

   ```
   let variabileNonInizializzata;
   console.log(variabileNonInizializzata); // Stampa: undefined
   ```

   `undefined` indica che la variabile esiste, ma non ha un valore definito. È importante notare che `undefined` è anche il valore restituito quando si cerca di accedere a una proprietà di un oggetto che non esiste.

2. **`null`**: Questo valore è assegnato esplicitamente da un programmatore per indicare l'assenza intenzionale di valore o dati in una variabile o in una proprietà di un oggetto. In altre parole, `null` è un valore che indica che non c'è nulla o che il valore è stato azzerato. Ad esempio:

   ```
   let variabileNullo = null;
   console.log(variabileNullo); // Stampa: null
   ```

   `null` è spesso utilizzato quando si desidera indicare che una variabile o una proprietà di un oggetto è vuota o non ha un valore valido.

In sintesi, la differenza principale è che `undefined` indica che una variabile esiste ma non è stata inizializzata, mentre `null` indica che è stata assegnata intenzionalmente l'assenza di valore. Entrambi sono valori falsy in JavaScript, il che significa che valutano a `false` in un contesto booleano.

Funzioni

Le funzioni si dichiarano con la parola riservata funzione.

```
function f(x,y){
    return x+y
}
```

- Appunti sulle Funzioni

Date Object

- Date è l'oggetto utilizzato per la rappresentazione di date.
- Internamente, questa rappresentazione è un numero che rappresenta i millisecondi trascorsi dalla data: 1 di gennaio del 1970.

- mostrare le date
- creare l'oggetto Date()
- formati e metodi per le date
- metodi get e metodi set

codice esempio

es data

Array

- JS array

Timers

Sono funzioni invocate dopo un tempo determinato.

funzione	significato
`setTimeout()`	Pianifica la invocazione dopo un tempo determinato
`setInterval()`	Pianifica l'invocazione dopo un intervallo di tempo
`clearTimeout()`	Resetta i timer
`clearInterval()`	Resetta i timer

Per esempio

```
setTimeout(function(){
    alert('asdf')
    }, 10000);

setInterval(function(){
    alert('asdf')
    }, 10000);
```

Ricapitolando

- Convenzioni del linguaggio e dialogs
- Variabili
- Operatori
- Costrutti condizionali e iterativi
- Tipi primitivi
- Tipi reference

- Booleans
- Function
- Number
- String
- Array
- Object
- Timers