Corso Javascript

JavaScript è un linguaggio di programmazione

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.

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

ruolo di js nel web design

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

Risorse disponibili online

- Javascript reference
- JS the right way
- JSinfo
- You-Dont-Know-JS
- w3schools
- Udacity Course
- html.it

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()

Strutture del linguaggio

JavaScript è **CASE SENSITIVE**

Terminare le istruzioni con punto e virgola (😉

- E' possibile l'uso del carattere (😉 per la separazione di frasi, si può omettere se queste si trovano su linee differenti (non fatelo!)
- Gli interpreti di JavaScript trattano i fine linea come fine istruzione, se la istruzione successiva non può interpretarsi come continuazione della precedente.

Convenzioni di naming:

- CamelCase,
- camelCase,
- snake_case

Identificatori

Gli identificatori in javascript cominciano con

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

seguito da

- lettere,
- numeri,
- underline,
- e caratteri di dollaro

Per esempio

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

Parole chiave riservate

-	-	-	-
abstract	else	instanceof	super
boolean	enum	int	switch
break	export	interface	synchronized

byte	extends	let	this
case	false	long	throw
catch	final	native	throws
char	finally	new	transient
class	float	null	true
const	for	package	try
continue	function	private	typeof
debugger	goto	protected	var
default	if	public	void
delete	implements	return	volatile
do	import	short	while
double	in	static	with

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.

use strict

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

const

Parola chiave per la dichiarazione di una costante. esempio

var, let

Parola chiave per la dichiarazione di una variabile.

Variabili

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

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
*/
```

Literals - Letterali (valori letterali per i tipi di dato)

tipo	esempio
Numeri interi	192
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

Per esempio

```
//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

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()

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
```

Funzioni

Le funzioni se dichiarano con la parola riservata funzione.

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

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

Objects

- JS Objects

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