Tecnologie Web: Javascript

Prof. Raffaele Montella, PhD
raffaele.montella@uniparthenope.it

Sommario

- Introduzione
- JavaScript integrato in HTML5
- Sintassi

Introduzione

- JavaScript è un linguaggio di scripting tipicamente client side
- Permette di creare pagine web attive
- E’ stato introdotto con Netscape Navigator 2.0 nel 1995
- E’ indipendente dalla piattaforma
- Accesso al Document Object Model

Introduzione

- L’interprete JavaScript gira all’interno del browser
- Con NodeJS è possibile utilizzare JavaScript lato server.
  https://nodejs.org
- È usato per sviluppare applicazioni mobile ibride.
  https://www.nativescript.org
- Oggi è usato insieme a librerie (es. JQuery).
  https://jquery.com
- È il runtime per sovralinguaggi, come Typescript.
  https://www.typescriptlang.org

Introduzione

- E’ un linguaggio di programmazione orientato agli oggetti
- E’ basato sui prototipi
  - Non esiste una distinzione tra classi ed istanze: solo oggetti
  - Si basa sul concetto di prototipo: un oggetto funge da prototipo da cui prendere le proprietà iniziali di un nuovo oggetto
  - Qualsiasi oggetto può specificare le sue proprietà alla creazione e al runtime
- Non è correlato in alcun modo a Java!

JavaScript Integrato in HTML5

- Attraverso l'elemento <script> è possibile integrare JavaScript in HTML5

```
<script language="JavaScript">
        …codice JavaScript…
</script>
```

- Se il supporto JavaScript è assente o disabilitato è possibile mostrare un messaggio alternativo

```
<noscript>Javascript non attivo</noscript>
```

JavaScript Integrato in HTML5

- Il codice JavaScript può essere inserito nell'elemento <head> o <body>
  - In <head> sono definite le funzioni
  - In <body> il codice che è eseguito quando la pagina è caricata nel browser
  - Conviene inserire gli script come ultima parte del <body>

- Il codice JavaScript può essere incluso a partire da un file differente
  <script src="myfunctions.js" />

Sintassi

- Ogni segmento di codice JS è costituito da uno o più statement:
  - Ogni statement è separato dagli altri tramite ;
  - L'interruzione di riga determina la fine dello statement

- JS è case sensitive

- E' possibile usare commenti a linea singola e multipla secondo le sintassi C++ e Java

```
// commento
/*
Questo è un commento
*/
```

Variabili

- Le variabili sono dichiarate mediante la keyword var
  var pi=3.14, x,y, name=“Pippo”;
- I nomi delle variabili devono iniziare con un carattere alfabetico o con underscore
- Le variabili non hanno controllo di tipo
- Le variabili possono cambiare di tipo al runtime

Variabili

- Visibilità
  - Le variabili dichiarate all’interno di una funzione hanno visibilità limitata alla funzione
  - Le variabili dichiarate all’esterno di una funzione sono globali

Tipi

- JavaScript ha tre tipi di dati primitivi: number, string e boolean
- Tutti gli altri elementi sono oggetti
- I valori numerici sono sempre memorizzati in virgola mobile
- I numeri esadecimali cominciano con 0x
- Valori booleani:
  - 0,”0”, stringhe vuote, undefined, null, NaN -> false
  - Tutto il resto è true

Stringhe

- Le stringhe possono essere racchiuse tra apici singoli e doppie
- Le stringhe possono contenere sequenze di escape: \n \” \\
- La concatenazione di stringhe avviene con l’operatore +
  var concat = “Stringa 1” + “Stringa 2”;

Operatori

- Molto simili al C
- Aritmetici (floating point)
  + - * / % ++ --

- Confronto o relazionali
  < <= == != >= > === !==
- == e != valutano l'uguaglianza di valori anche tra variabili di tipo diverso effettuando le opportune conversioni
- === e !== considerano gli operandi diversi se sono di tipo diverso

Operatori

- Logici
  && || !
- Bitwise
  & | ^ ~ << >> >>>
- Assegnazione
  += -= *= /= %= <<= >>= >>>= &= ^= !=
- Concatenazione di stringhe
  +
- Assegnazione condizionale
  (condizione) ? (valore vero) : (valore falso)

Operatori

- Nuovo oggetto:
  new Class()
- Tipo di dato
  typeof(x)
- Confronto di istanze
  a instanceof A
- Dellocazione di un oggetto
  delete x

Funzioni

- Vanno definite all'interno dell'elemento <head>
- Sintassi:

```
function verbObject (arg1, arg2, …, argn) {
     … blocco di codice ..
}
```

- Una funzione può restituire un valore con return value;
- Per invocare una funzione:

```
verbObject(arg1, arg2, …, argn)
```

Istruzioni

- La sintassi di molte istruzioni JS è simile al C
  - Assegnazioni
  - Blocchi di istruzioni
  - Costrutti condizionali ed iterativi
- Iterazione sugli attributi di un oggetto

```
for (var i in obj) {
    print(i+" = " obj[i]);
}
```

Istruzioni

- Gestione delle eccezioni
- Per lanciare un'eccezione
  throw
- Per catturare le eventuali eccezioni sollevate da un blocco di codice

```
try {
     … blocco di codice …
} catch (e) {
     … blocco che gestisce l'eccezione …
}
```

Conclusioni

- Il Javascript è un linguaggio di scripting

- Prevalentemente lato client
- Si usa NodeJS sul server per applicazioni high-throughput (es. IoT)

- Sintassi simile al C/C++/Java
- In nessun modo legato a Java