

2022_02_02

March 28, 2022

1 2022-02-02

1.0.1 Corso ITS

1.1 Magento & e-commerce software

1.2 ### JavaScript (Andrea Ribuoli)

1.2.1 seguiremo il testo JavaScript: la guida definitiva di *David Flanagan*

1.2.2 tradotto dalla 7a edizione nel 2021 per APOGEO

1.3 STRUTTURA LESSICALE

1. Il testo di un programma JavaScript
2. I commenti
3. I letterali
4. Gli identificatori e le parole riservate
5. Parole riservate
6. Unicode
7. Sequenze di escape Unicode
8. Normalizzazione Unicode
9. Il punto e virgola opzionale

1.3.1 1. Il testo di un programma JavaScript

```
[2]: !echo "prova" > prova.txt;   
echo "su nuova riga" >> prova.txt
```

```
[6]: !hexdump -c prova.txt
```

```
00000000 p r o v a \n s u n u o v a r  
00000010 i g a \n  
00000014
```

```
[3]: !hexdump prova.txt
```

```
00000000 7072 6f76 610a 7375 206e 756f 7661 2072  
00000010 6967 610a  
00000014  
  
p r o v a
```

```
70 72 6f 76 61 0a
```

```
  s u      n u o v a      r i g a  
73 75 20 6e 75 6f 76 61 20 72 69 67 61 0a
```

1.3.2 hex 14 == dec 20 ($1 * 16^1 + 4 * 16^0$)

```
[5]: !ls -la
```

```
total 60  
-rw-rw-rw-  1 jupyter 0  3952 feb  2 14:42 2022_02_02.ipynb  
-rw-rw-rw-  1 jupyter 0    20 feb  2 14:29 prova.txt  
drwxrwsrwx  3 jupyter 0  8192 feb  2 14:42 .  
drwxrwsrwx  2 jupyter 0 12288 feb  2 14:29 .ipynb_checkpoints  
drwxrwsrwx 18 jupyter 0 24576 gen 26 09:46 ..
```

1.3.3 2. I commenti

```
//  
  
/* */  
  
/*  
  
*/  
  
/*  
*  
*  
*  
*  
*/
```

1.3.4 3. I letterali

- 12
- 1.2
- "Hello world!"
- 'Una nuova string'
- true
- false
- null

```
> a = true  
true  
> True = "prova"  
'prova'  
> True === true  
false  
>
```

1.3.5 4. Gli identificatori e le parole riservate

Lettera iniziale o `_` (underscore) o `$` (dollaro)

1.3.6 5. Parole riservate

- `as`
- `async`
- `await`
- `break`
- `case`
- `catch`
- `class`
- `const`
- `continue`
- `debugger`
- `default`
- `delete`
- `do`
- `else`
- `false`
- `function`
- `for`
- `get`
- `if`
- `in`
- `import`
- `let`
- `null`
- `of`
- `return`
- `set`
- `static`
- `super`
- `switch`
- `this`
- `throw`
- `true`
- `var`
- `void`
- `while`
- `with`
- `yield`

1.3.7 Unicode

```
> euro = "€"  
'€'  
> mille = 1000
```

```

1000
> mille + euro
'1000€'

> const = 3.14
undefined
> diametro = 2 *
6.28
> sì = true
true

```

1.3.8 7. Sequenze di escape Unicode

```

> prova = "Non trovo sulla tastiera il simbolo \u20ac"
'Non trovo sulla tastiera il simbolo €'

> idea = "Sarebbe ora di un buon caffè"
'Sarebbe ora di un buon caffè'
> a = "\u00e8"
'è'
> idea === "Sarebbe ora di un buon caff\u00e8"
true
> idea === 'Sarebbe ora di un buon caff\u00e8'
true

> b = "\\u00e8"
'\\u00e8'
> console.log(b)
\u00e8
undefined
> console.log(a)
è
undefined

> caffè = 1.10
1.1
> caffè\u00e8
1.1

```

1.3.9 8. Normalizzazione Unicode

```

> c = "café"
'café'
> d = "cafe\u0301"
'café'
> c === d
false
> e = "caf\u00e9"
'café'
> e === c
true

```

```
true
```

```
[ ]: c = "café"  
d = "cafe\u0301"  
e = "caf\u00e9"  
console.log(c + d + e)
```

```
[9]: !node normalizzi.js > moltocaffe.txt
```

```
[13]: !hexdump -c moltocaffe.txt
```

```
00000000  c  a  f           c  a  f  e   201  c  a  f  
00000010  \n  
00000011
```

```
[14]: !hexdump moltocaffe.txt
```

```
00000000 6361 66c3 a963 6166 65cc 8163 6166 c3a9  
00000010 0a00  
00000011
```

1.3.10 9. Il punto e virgola è opzionale (se vado a capo!)

```
> puntoA = { x: 1.0, y: 2.0 }; puntoB = { x: 1.0, y: 2.0 }  
{ x: 1, y: 2 }  
> puntoA  
{ x: 1, y: 2 }  
> delete puntoA  
true  
> delete puntoB  
true  
> puntoA = { x: 1.0, y: 2.0 }; puntoB = { x: 3.0, y: 4.0 }  
{ x: 3, y: 4 }  
> puntoA  
{ x: 1, y: 2 }
```

1.4 TIPI, VALORI E VARIABILI

```
[ ]: -9007199254740992 --- 9007199254740992  
9007199254740992 = 2^53
```

1.4.1 Panoramica e definizioni

1.4.2 I numeri

1.4.3 I letterali interi

1.4.4 I letterali in virgola mobile

1.4.5 L'aritmetica in JavaScript

- 1.4.6 Numeri in virgola mobile binari ed errori di arrotondamento
- 1.4.7 Numeri interi con precisione arbitraria grazie a BigInt
- 1.4.8 Date e orari
- 1.4.9 Il testo
- 1.4.10 I letterali stringa