

2022_02_23

March 28, 2022

1 2022-02-23

1.0.1 Corso ITS

1.1 Magento & e-commerce software

1.2 ### JavaScript (Andrea Ribuoli)

```
[6]: !curl -o c19.html https://www.worldometers.info/coronavirus/
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current	
			Dload	Upload	Total	Spent	Left	Speed
0	0	0	0	0	0	0	0	0

```
[7]: !grep "<table" c19.html
```

```
<table id="main_table_countries_today" class="table table-bordered table-hover
main_table_countries" style="width:100%;margin-top: 0px
!important;display:none;">
<table id="main_table_countries_yesterday" class="table table-bordered table-
hover main_table_countries" style="width:100%;margin-top: 0px
!important;display:none;">
<table id="main_table_countries_yesterday2" class="table table-bordered table-
hover main_table_countries" style="width:100%;margin-top: 0px
!important;display:none;">
```

```
[ ]: !node dom.js
```

```
[ ]: !node dom_c19.js
```

```
const { JSDOM } = require("jsdom")

let level = 0
let posizioni = [1]
let posizioni_tot = [1]

function myF(e) {
  console.log(level.toString() + ' [' + posizioni.join() + ']' + Object.prototype.toString)
}
```

```

function myF2(e) {
    if (level == 4 &&
        posizioni[level] == 1 &&
        Object.prototype.toString.call(e) == '[object HTMLAnchorElement]')
        console.log(e.innerHTML)
}

function attraversa(elem, f) {
    f(elem)
    level++
    posizioni[level] = 0
    posizioni_tot[level] = elem.childElementCount
    for(let figlio of elem.children) {
        posizioni[level]++
        attraversa(figlio, f)
    }
    posizioni.pop()
    posizioni_tot.pop()
    level--
}

function analizza(d) {
    let b = d.window.document.querySelector("#main_table_countries_today")
    attraversa(b, myF2)
}

JSDOM.fromFile('c19.html').then(analizza)

```

1.3 function declaration .vs. function expression

```

function awesomeFunction(coolThings) {
    // ..
    return amazingStuff;
}

// let awesomeFunction = ..
// const awesomeFunction = ..
var awesomeFunction = function(coolThings) {
    // ..
    return amazingStuff;
};

```

Quest'ultima è una espressione che viene assegnata alla variabile **awesomeFunction**. In modo differente dalla dichiarazione di funzione essa non è associata all'identificatore fintanto che quella istruzione non venga eseguita (runtime).

Non tutti i linguaggi di programmazione trattano le funzioni come *valori*, ma questo aspetto è cruciale perchè un linguaggio di programmazione supporti il *pattern* di programmazione **funzionale**.

1.3.1 NON USARE PIU' document.write()

###

1.3.2 <script defer ...

1.3.3 <script async ...

<body>

<script > vfvffvf </script> (SINCRONA o bloccante)

<body>

<script defer> </script>

<body>

<script async> </script>

<body>

<script defer async> </script>

<body>

<script defer src="ritardato.js"> </script>

<script async src="asincrono.js"> </script>

```
function range(from, to) {
  let r = Object.create(range.methods)
  r.from = from
  r.to = to
  return r
}

range.methods = {
  includes(x) { return this.from <= x && x <= this.to; },
  *[Symbol.iterator]() {
    for(let x = Math.ceil(this.from); x <= this.to; x++) yield x;
  },
  toString() { return "(" + this.from + "... " + this.to + ")"; }
}
```

[31]: !node intervallo.js

```
[3.1, 7.2] include 2.9 è false  
[3.1, 7.2] include 3.7 è true  
[4°5°6°7]  
4  
5  
6  
7  
from  
to  
[object Object]
```

[38]: !node provaIteratore.js

```
Il nostro primo punto è: 3.2 + 4.1 i  
Il nostro secondo punto è: 2.4 + 7.4 i  
Il nostro prodotto è: -22.66 + 33.52 i
```