

Tecnologie Cloud e Mobile

Lez. 05

JavaScript - Esempi

Giuseppe Psaila

Università di Bergamo

giuseppe.psaila@unibg.it

altervista.org

- È un sistema PaaS gratuito, decisamente basico
- Conviene usarlo per fare un po' di prove
- Per abituarsi a ragionare sul web

BACHECA

RISORSE

WORDPRESS

GUADAGNA


PROMUOVI

ACCESSORI

Applicazioni



Gestione File

 Visualizza | FTP Accedi

Risorse

SPAZIO WEB →

TRAFFICO →

Guadagni da Altervista

BONUS
ISCRIZIONE

€ 5,00

GUADAGNI
MARZO

€ 0,00

GUADAGNI
IERI

€ 0,00

GUADAGNI
TOTALI

€ 5,00

■ Euro



0.10

0.07

altervista.org

Gestione File

/



PHP


FTP

Invia files

Sposta

Copia

Cancella

<input type="checkbox"/>	File
<input type="checkbox"/>	 2017

altervista.org

NuovaCartella

Crea Cartella

NuovoFile


.

html



Crea File

altervista.org

<input type="checkbox"/>	<div><div> Messaggio_0.html</div><div>Apri Modifica Modifica con l'editor avanzato Rinomina Elimina</div></div>

Messaggio_0.html

Prendiamo il codice che formatta un messaggio a scala

- Predisponiamo nel copro un DIV nel quale inserire il testo formattato, con id="area_risultato"
- Predisponiamo una funzione gen_messaggio che riceve una stringa come parametro e restituisce una stringa
- Predisponiamo una funzione formatta che pilota la formattazione e l'inserimento nell'HTML
- Impostiamo l'evento onLoad sull'elemento BODY

Messaggio_0.html

WELCOME EVERYBODY

Messaggio_0.html

```
<body onLoad="formatta()">
```

```
<div id="area_risultato">
```

```
Qui il risultato
```

```
</div>
```

```
</body>
```



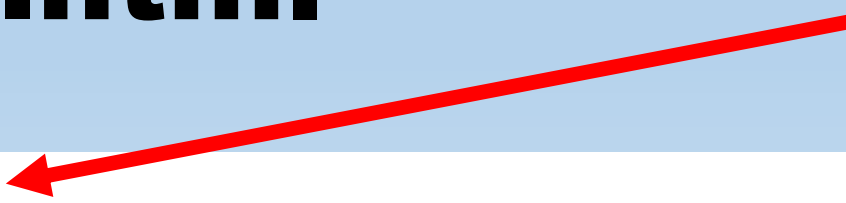
Messaggio_0.html

```
<script language = "javascript">
function gen_messaggio(message) {
    var format=message.toUpperCase()
    var size=1 //go through the message, letter by letter
    var risultato="";
    for (i=0;i<message.length;i++) {
        var m = format.charAt(i).fontSize(size).bold();
        risultato += m;
        if (size<7)
            size++;
        else
            size=1;
    }
    return risultato;
}
```

Messaggio_0.html

```
function formatta() {  
    var nodo = document.getElementById("area_risultato");  
    nodo.innerHTML = gen_messaggio("Welcome everybody");  
    return false;  
}  
</script>
```

Messaggio_0.html



```
<body onLoad="formatta()">  
  <div id="area_risultato">  
    Qui il risultato  
  </div>  
  
</body>
```

Messaggio_1.html

Modifichiamo il precedente

- Inseriamo una form
- Con un solo campo, per inserire il testo da formattare
- Il pulsante di submit avvia la formattazione
- Ma occorre fermare la richiesta HTTP

Messaggio_1.html

Formatta

Qui il risultato

Formatta

SALVE **M**o**NDO**

Messaggio_1.html

<body>

<form name="myform" onSubmit="return formatta();">

<input name="testo" type="text"/>

<input type="submit" value="Formatta"/>

</form>

<div id="area_risultato">

Qui il risultato

</div>

Messaggio_1.html

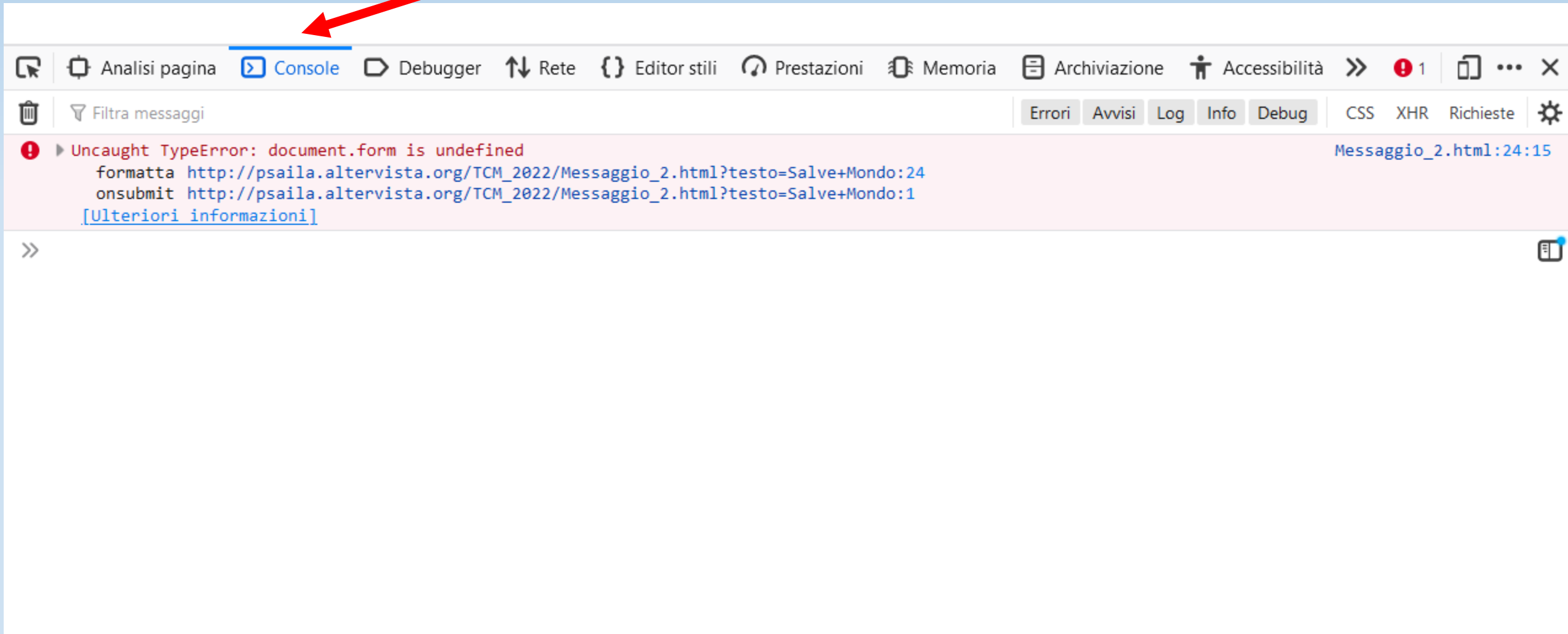
```
function formatta() {  
    var msg = document.forms.myform.testo.value;  
    var nodo = document.getElementById("area_risultato");  
    nodo.innerHTML = gen_messaggio(msg);  
    return false;  
}
```


Messaggio_2.html

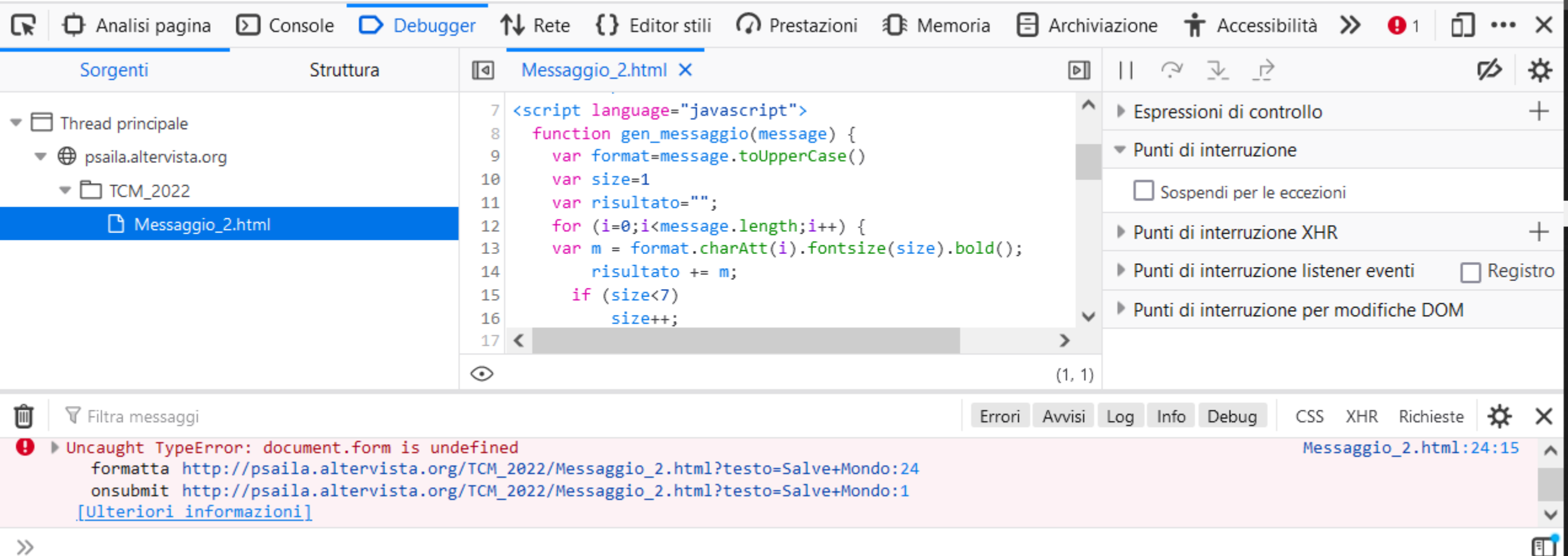
Ho introdotto un paio di errori e NON VA

- Dobbiamo usare la console e il debugger
- Su Firefox:
tasti SHIFT+F12
oppure Tasto destro -> Analizza
- Su Chrome:
tasto destro->Ispeziona
tasti CTRL+SHIFT+I

Firefox



Firefox



The screenshot displays the Firefox Developer Tools interface, specifically the Debugger tab. A red arrow points to the 'Debugger' tab in the top navigation bar. The interface is divided into three main sections: 'Sorgenti' (Sources), 'Struttura' (Structure), and 'Debugger'.

Sorgenti (Sources): Shows the file hierarchy for the current page. The file 'Messaggio_2.html' is selected.

Debugger: The main area for debugging. It contains a code editor showing the JavaScript code for 'Messaggio_2.html'. The code is as follows:

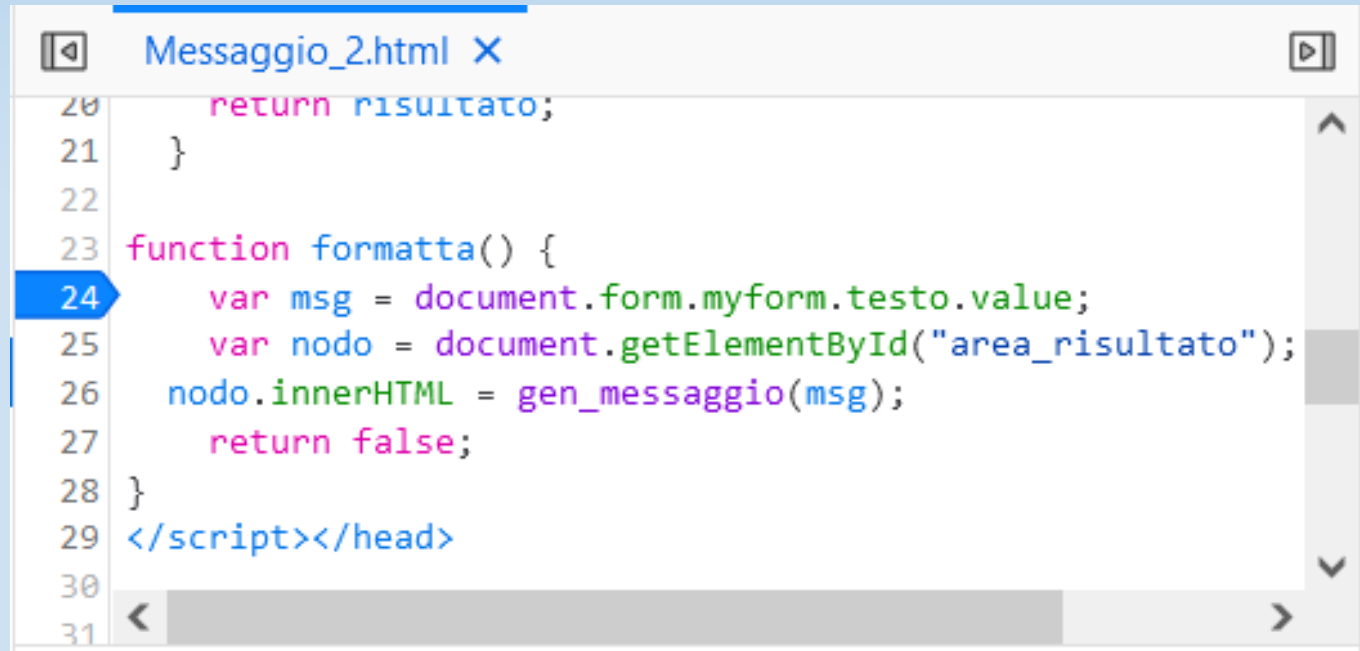
```
7 <script language="javascript">
8   function gen_messaggio(message) {
9     var format=message.toUpperCase()
10    var size=1
11    var risultato="";
12    for (i=0;i<message.length;i++) {
13      var m = format.charAt(i).fontSize(size).bold();
14      risultato += m;
15      if (size<7)
16        size++;
17    }
```

The right sidebar of the Debugger tab shows various debugging options, including 'Espressioni di controllo', 'Punti di interruzione' (Breakpoints), 'Punti di interruzione XHR', 'Punti di interruzione listener eventi', and 'Punti di interruzione per modifiche DOM'.

Console: The bottom section of the interface shows the console messages. A red error message is displayed: 'Uncaught TypeError: document.form is undefined'. The error message includes the file name and line number: 'Messaggio_2.html:24:15'. The console also shows the URL of the page: 'http://psaila.altervista.org/TCM_2022/Messaggio_2.html?testo=Salve+Mondo:24'.

Firefox

Break Point



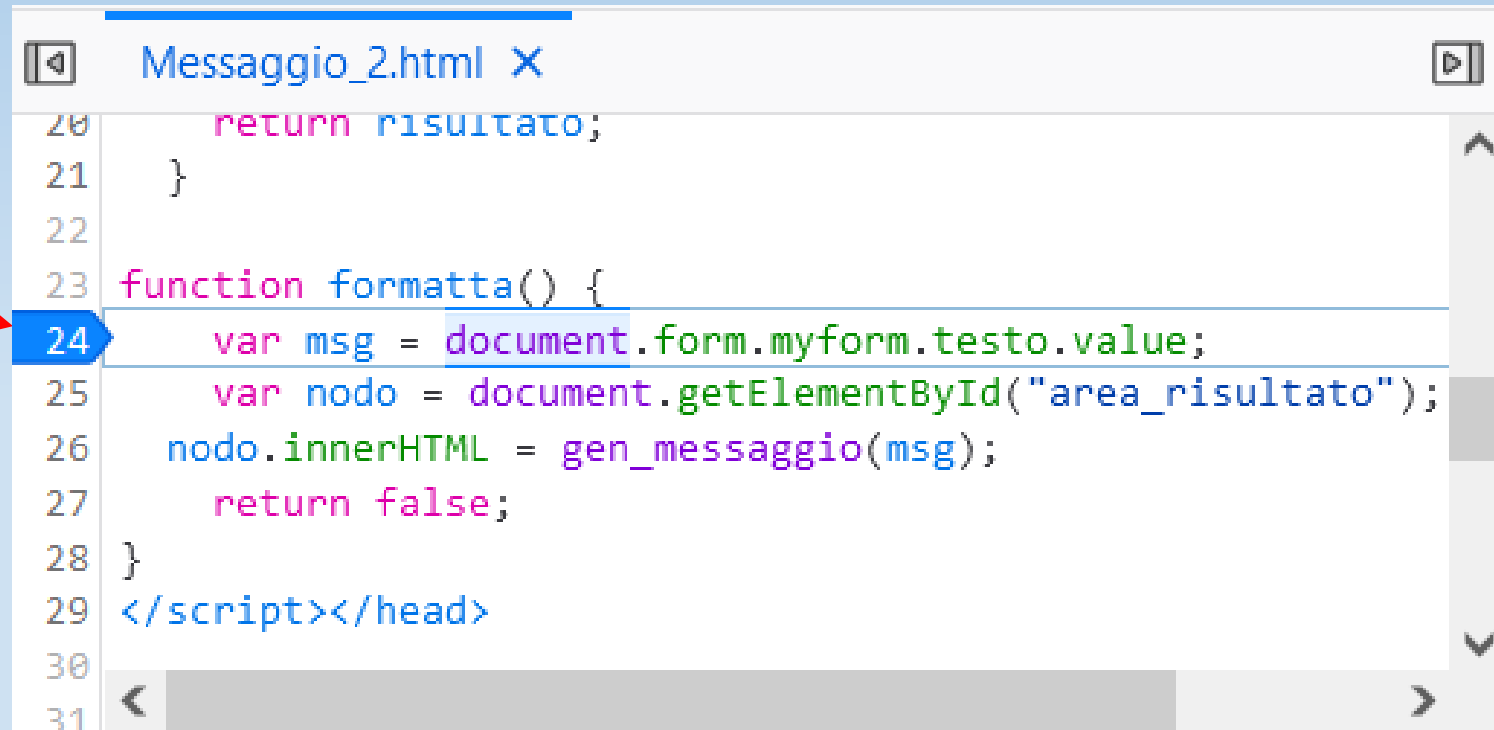
The screenshot shows the Firefox DevTools interface with a file named 'Messaggio_2.html' open. The code is as follows:

```
20     return risultato;  
21 }  
22  
23 function formatta() {  
24     var msg = document.form.myform.testo.value;  
25     var nodo = document.getElementById("area_risultato");  
26     nodo.innerHTML = gen_messaggio(msg);  
27     return false;  
28 }  
29 </script></head>  
30  
31
```

A blue highlight is on line 24, and a red arrow points to it from the 'Break Point' label.

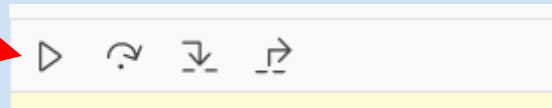
Firefox

Break Point



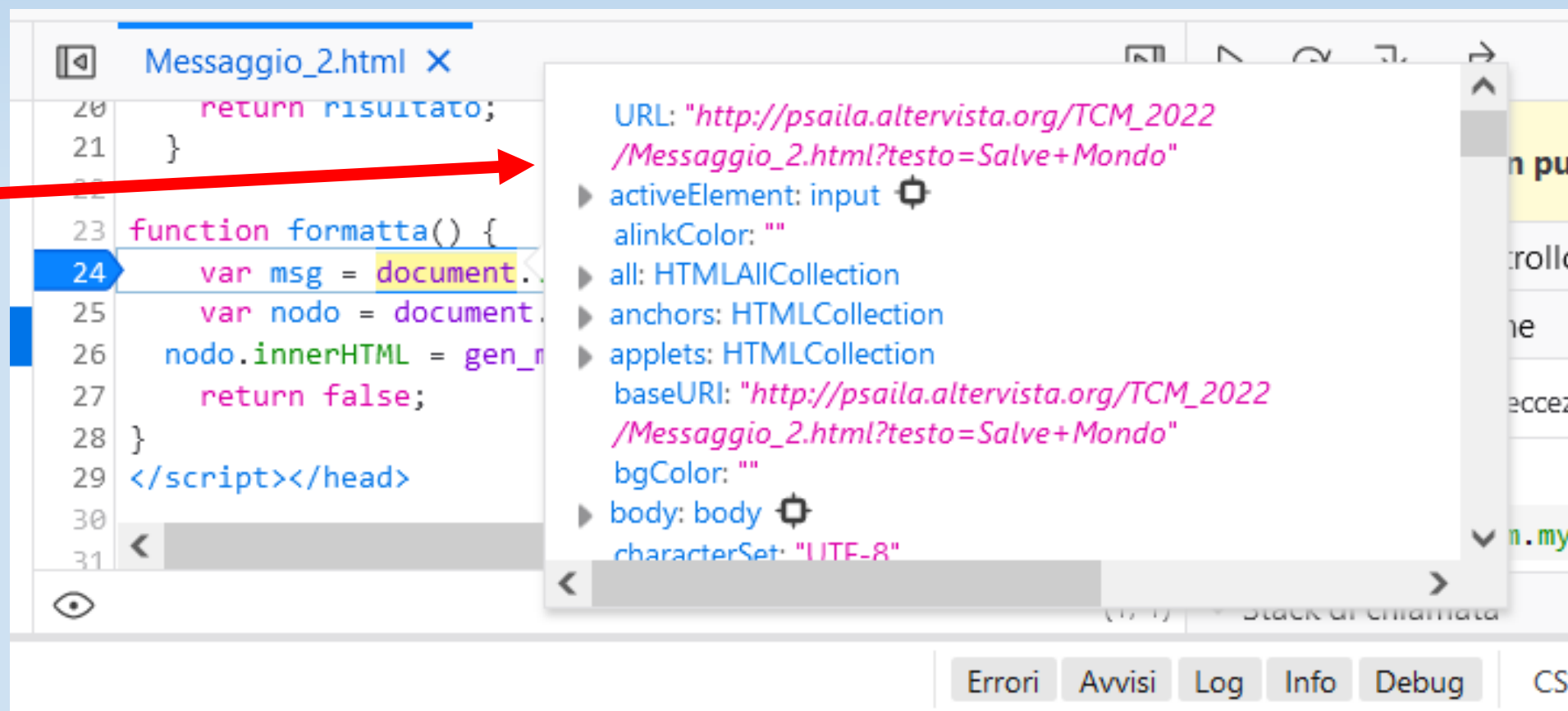
```
20     return risultato;
21 }
22
23 function formatta() {
24     var msg = document.form.myform.testo.value;
25     var nodo = document.getElementById("area_risultato");
26     nodo.innerHTML = gen_messaggio(msg);
27     return false;
28 }
29 </script></head>
30
31
```

Pulsanti di controllo della esecuzione

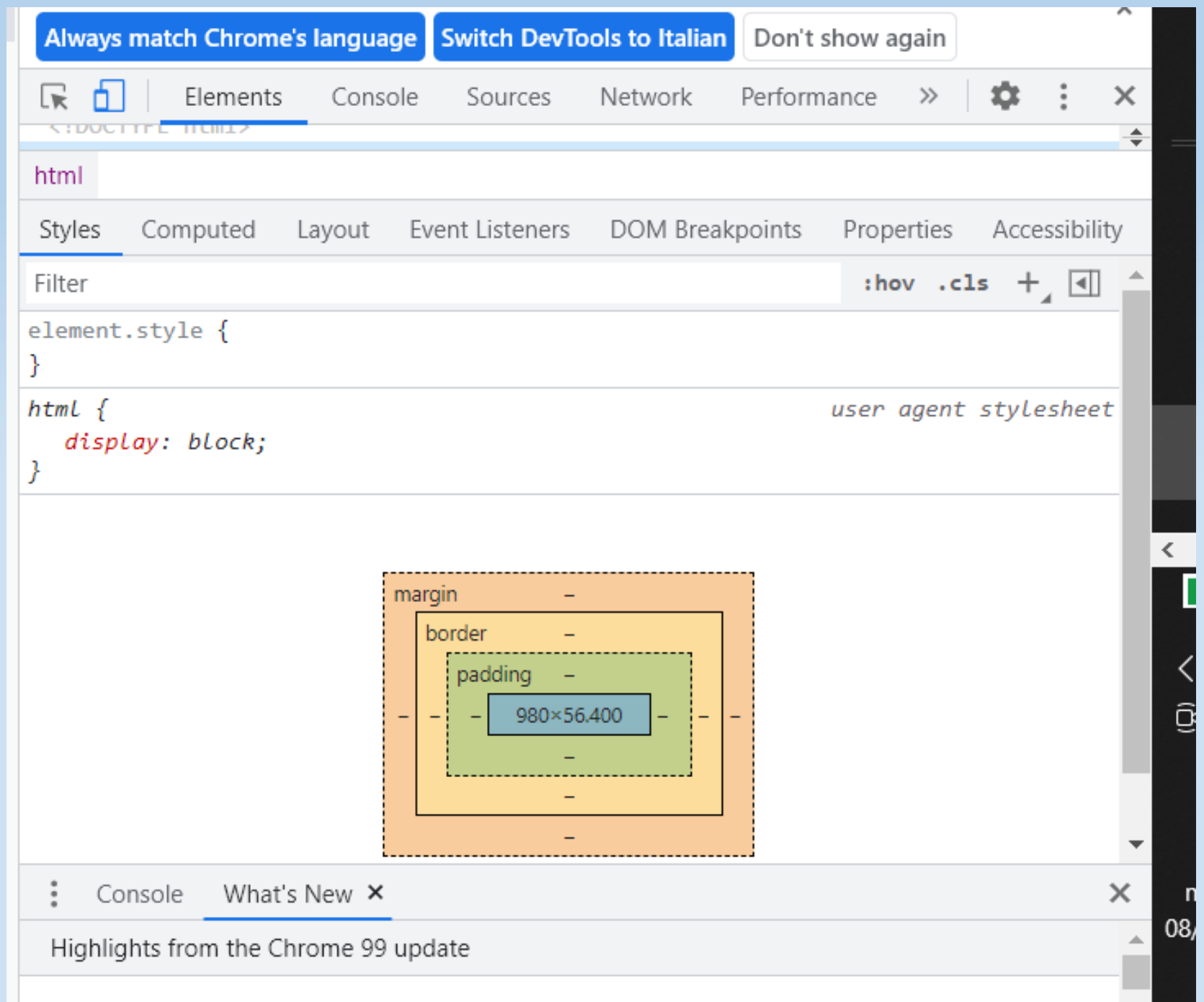


Firefox

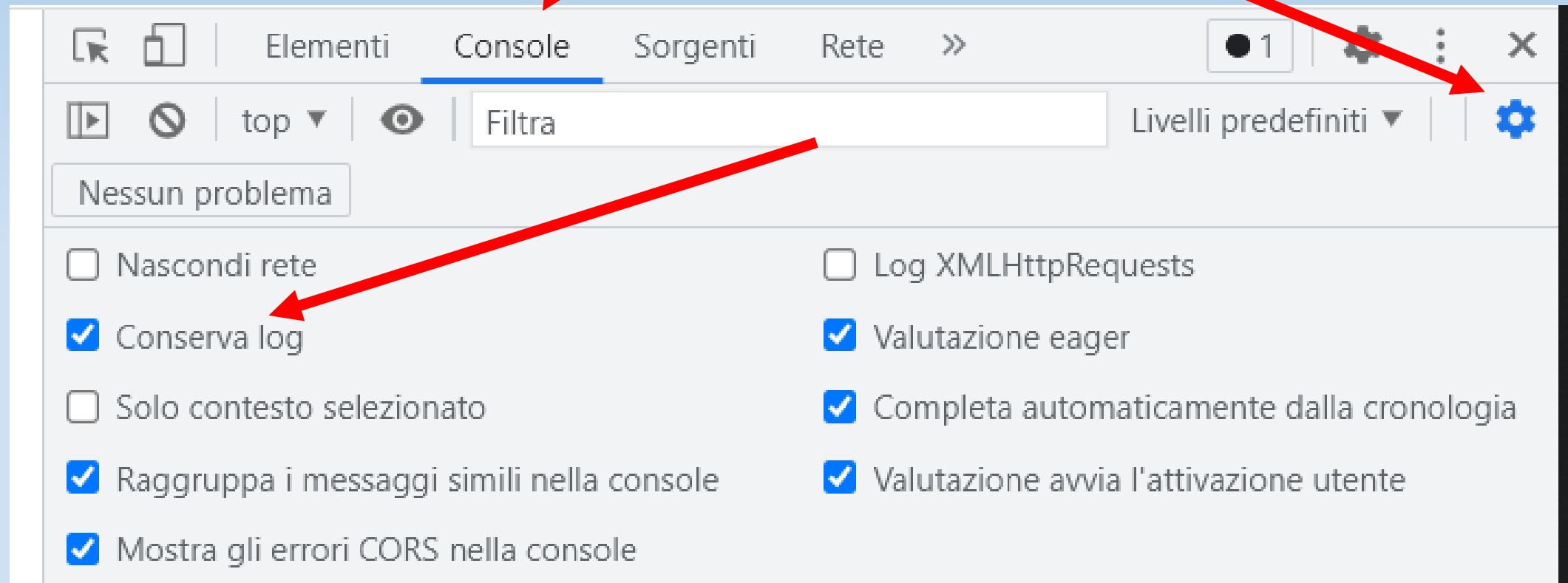
Ispezione delle
variabili



Chrome



Chrome



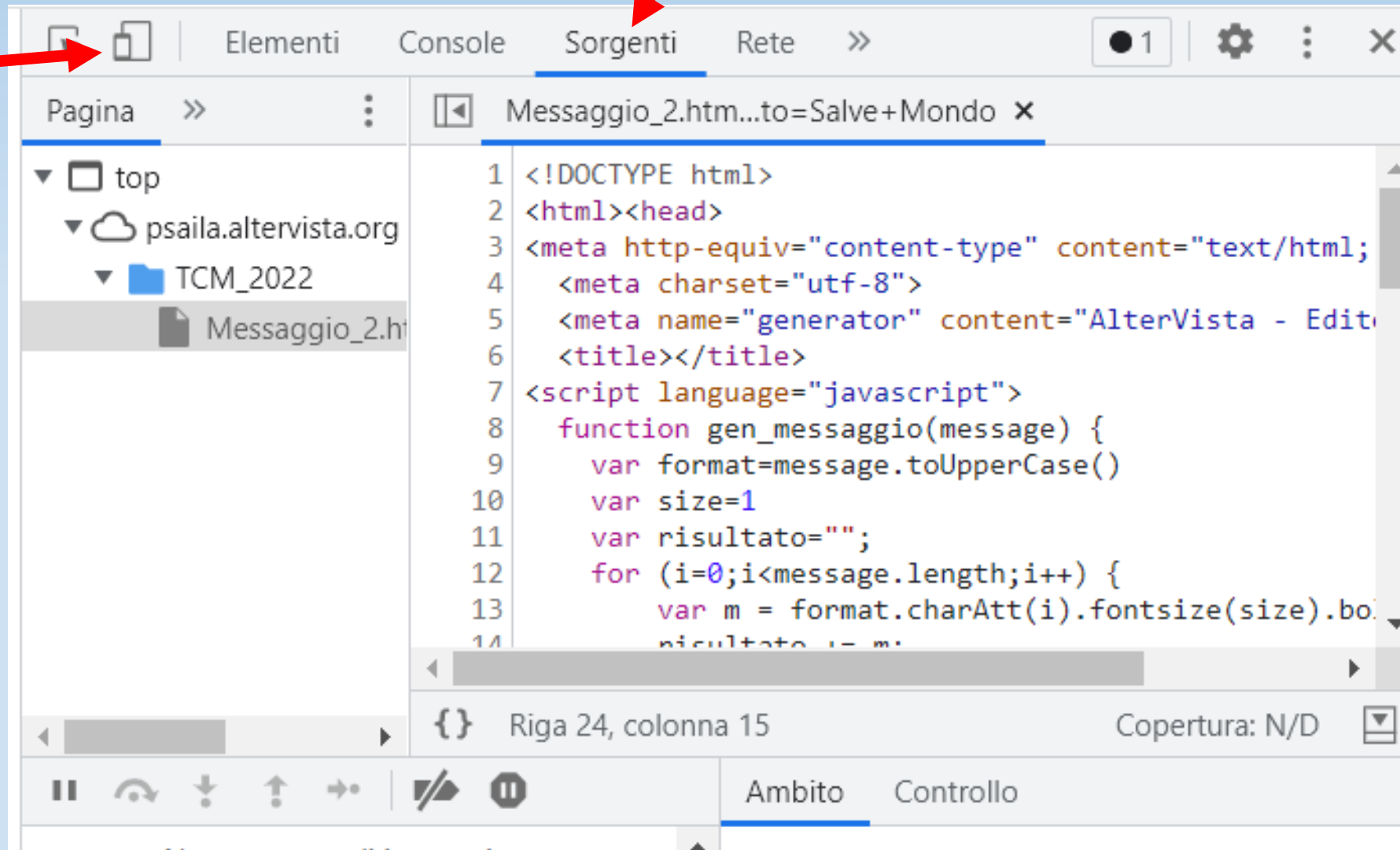
Chrome

```
✖ ▶ Uncaught TypeError: format.charAtt Messaggio_2.html?testo=Salve+Mondo:13  
is not a function  
    at gen_messaggio (Messaggio_2.html?testo=Salve+Mondo:13:19)  
    at formatta (Messaggio_2.html?testo=Salve+Mondo:26:19)  
    at HTMLFormElement.onSubmit (Messaggio_2.html?testo=Salve+Mondo:34:60)
```

Destinazione navigazione: http://psaila.altervista.org/TCM_2022/Messaggio_2.html?testo=Salve+Mondo

>

Chrome



Calcolatrice_1.html

Predisponiamo una calcolatrice

- Inseriamo una form
- Con due campi denominati op1 e op2
- E quattro pulsanti, per le 4 operazioni

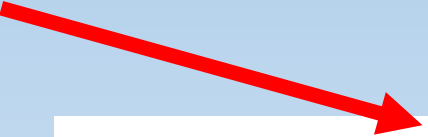
Calcolatrice_1.html

Qui il risultato

1+2=3

1-2=-1

Calcolatrice_1.html



```
<form name="myForm" onSubmit="return false;">
```

```
<input type="text" name="op1"/>
```

```
<input type="text" name="op2"/>
```

```
<input type="submit" value="+" onClick="Calcola('+');" />
```

```
<input type="submit" value="-" onClick="Calcola('-');" />
```

```
<input type="submit" value="*" onClick="Calcola('*');" />
```

```
<input type="submit" value="/" onClick="Calcola('/');" />
```

```
</form>
```



Calcolatrice_1.html

```
function Calcola(operazione) {  
    var v1=parseInt(document.forms.myForm.op1.value);  
    var v2=parseInt(document.forms.myForm.op2.value);  
    if (isNaN(v1))  
        v1 = 0;  
    if (isNaN(v2))  
        v2 = 0;  
  
    var testo = "";
```

Calcolatrice_1.html

```
if(operazione == "+")
    testo = "" + v1 + "+" + v2 + "=" + (v1 + v2);
if(operazione == "-")
    testo = "" + v1 + "-" + v2 + "=" + (v1 - v2);
if(operazione == "*")
    testo = "" + v1 + "*" + v2 + "=" + (v1 * v2);
```

Calcolatrice_1.html

```
if(operazione == "/")
{
    if( v2 == 0)
        testo = "" + v1 + "/" + v2 + "Impossibile";
    else
        testo = "" + v1 + "/" + v2 + "=" + (v1 / v2) ;
}
```


Calcolatrice_1.html

```
var nodo = document.getElementById("risultato");  
nodo.innerHTML = "<p>" + testo + "</p>";
```

Calcolatrice_2.html

Predisponiamo una calcolatrice, ma «ragioniamo alla JavaScript

- Ci ricordiamo che possiamo passare le funzioni come parametri
- Quindi predisponiamo 4 funzioni, una per ogni operazione

Calcolatrice_2.html

```
var fsomma=function (v1, v2)
{
    var r = v1 + v2;
    var testo = "" + v1 + "+" + v2 + "=" + r;
    return testo;
}

var fdifferenza=function (v1, v2)
{
    var r = v1 - v2;
    var testo = "" + v1 + "-" + v2 + "=" + r;
    return testo;
}
```

Calcolatrice_2.html


```
var fprodotto=function (v1, v2)
{
    var r = v1 * v2;
    var testo = "" + v1 + "*" + v2 + "=" + r;
    return testo;
}
```

Calcolatrice_2.html

```
var fdivisione=function (v1, v2)
{
    var r = v1 / v2;
    var testo = "" + v1 + "/" + v2 + "=";
    if (v2 != 0) {
        r = v1 / v2;
        testo += r;
    } else {
        testo += " impossibile";
    }
    return testo;
}
```


Calcolatrice_2.html

```
function Calcola(operazione) {  
    var v1=parseInt(document.forms.myForm.op1.value);  
    var v2=parseInt(document.forms.myForm.op2.value);  
    if (isNaN(v1))  
        v1 = 0;  
    if (isNaN(v2))  
        v2 = 0;  
  
    var testo = operazione(v1, v2);  
  
    var nodo = document.getElementById("risultato");  
    nodo.innerHTML = "<p>" + testo + "</p>";  
}
```



Calcolatrice_2.html

```
<form name="myForm" onSubmit="return false;">
  <input type="text" name="op1"/>
  <input type="text" name="op2"/>
  <input type="submit" value="+" onClick="Calcola(fsomma);"/>
  <input type="submit" value="-" onClick="Calcola(fdifferenza);"/>
  <input type="submit" value="*" onClick="Calcola(fprodotto);"/>
  <input type="submit" value="/" onClick="Calcola(fdivisione);"/>
</form>
```



Calcolatrice_3.html

Estendiamo la calcolatrice

- Per avere la lista delle operazioni svolte, invece di una singola operazione
- Con un pulsante che svuota la lista
- Ma ragioniamo alla JavaScript: creiamo una «classe» denominata MioElenco, che gestisce l'elenco dei risultati

Calcolatrice_3.html

Qui il risultato:

$$1+2=3$$

$$1-2=-1$$

$$1*2=2$$

$$1/2=0.5$$

Calcolatrice_3.html

```
function MioElenco( id_result)
{
    this.id_result = id_result;
    this.contatore = 0;
    this.Elenco = new Array(1);
```

Calcolatrice_3.html

```
this.addToElenco = function( elemento ) {  
    this.Elenco[this.contatore] = elemento;  
    this.contatore ++;  
}
```

Calcolatrice_3.html

```
this.showElenco = function() {  
    var nodo = document.getElementById(this.id_result);  
    var i;  
    for (i=0; i < this.contatore; i++) {  
        if( i==0)  
            nodo.innerHTML = this.Elenco[i];  
        else  
            nodo.innerHTML += "<br/>" + this.Elenco[i];  
    }  
}
```

Calcolatrice_3.html

```
this.Svuota = function() {  
    var nodo = document.getElementById(this.id_result);  
    nodo.innerHTML = "Qui il risultato:";  
    this.contatore = 0;  
}  
}
```

Calcolatrice_3.html

```
var Elenco;  
  
function Inizializza()  
{  
    Elenco = new MioElenco("risultato");  
    Elenco.Svuota();  
}
```

Calcolatrice_3.html

```
function Calcola(operazione) {  
    var v1=parseInt(document.forms.myForm.op1.value);  
    var v2=parseInt(document.forms.myForm.op2.value);  
    if (isNaN(v1))  
        v1 = 0;  
    if (isNaN(v2))  
        v2 = 0;  
    var testo = operazione(v1,v2);  
    Elenco.addToElenco( testo );  
    Elenco.showElenco();  
}
```

```
<body onLoad="Inizializza();">
```

```
<form name="myForm" onSubmit="return false;">
```

```
<input type="text" name="op1"/>
```

```
<input type="text" name="op2"/>
```

```
<input type="submit" value="+" onClick="Calcola(fsomma);"/>
```

```
<input type="submit" value="-" onClick="Calcola(fdifferenza);"/>
```

```
<input type="submit" value="*" onClick="Calcola(fprodotto);"/>
```

```
<input type="submit" value="/" onClick="Calcola(fdivisione);"/>
```

```
<input type="submit" value="Svuota" onClick="Elenco.Svuota();"/>
```

```
</form>
```

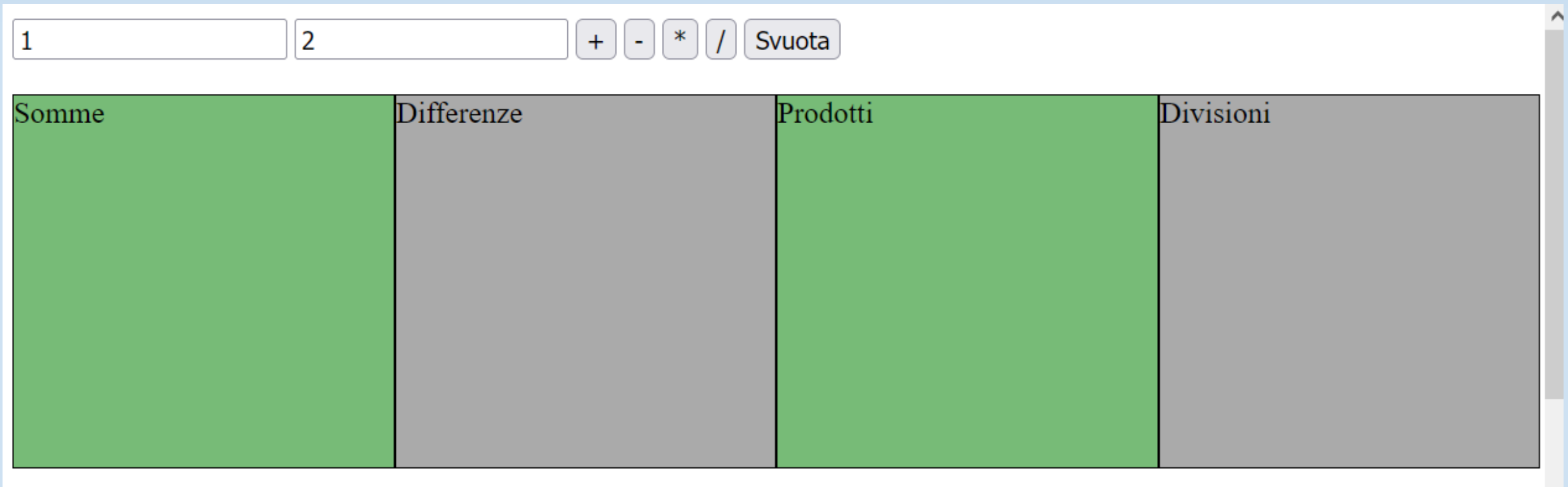
```
<div id="risultato">
```

```
</div>
```


Calcolatrice_4.html

SFIDA:

- Creare quattro aree di risultato, una per ogni operazione



The image shows a web-based calculator interface. At the top, there are two input fields containing the numbers '1' and '2'. To the right of these fields are five buttons: a plus sign (+), a minus sign (-), a multiplication sign (*), a division sign (/), and a button labeled 'Svuota' (Clear). Below the input fields and buttons, there are four large rectangular areas for displaying results. The first area, labeled 'Somme' (Sum), is green. The second area, labeled 'Differenze' (Differences), is gray. The third area, labeled 'Prodotti' (Products), is green. The fourth area, labeled 'Divisioni' (Divisions), is gray. The interface is simple and functional, designed to demonstrate the challenge of creating four distinct result areas for different operations.