

Laboratorio 3 – Mercoledì 21/10/2020

- Per modificare il documento utilizziamo il cosiddetto DOM, un albero che viene creato al momento del caricamento del browser.
- Javascript permette la modifica di elementi della pagina: prima cosa che dobbiamo saper fare è recuperare questi elementi all'interno dell'albero DOM.
- **Metodi per ottenere gli elementi DOM:** per modificare il documento utilizzeremo l'oggetto document

Metodo	Descrizione
<code>document.getElementById(<i>id</i>)</code>	Ottiene un elemento dal suo ID
<code>document.getElementsByTagName(<i>name</i>)</code>	Ottiene elementi tramite il nome del tag
<code>document.getElementsByClassName(<i>name</i>)</code>	Ottiene elementi tramite la classe

- Proprietà e metodi per cambiare gli elementi DOM:

Proprietà	Descrizione
<code>element.innerHTML = <i>new html content</i></code>	Cambia il contenuto interno di un elemento
<code>element.attribute = <i>new value</i></code>	Cambia il valore di un attributo di un elemento
<code>element.style.property = <i>new style</i></code>	Cambia lo stile (proprietà CSS) di un elemento
Metodo	Descrizione
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Cambia il valore di un attributo di un elemento

Attenzione: come già detto con Marcelloni i nomi degli stili sono diversi (si rimuovono gli scores e si pongono lettere maiuscole per distinguere le varie parole)

```
document.getElementsByTagName("H1")[0].setAttribute("class", "democlass");
document.getElementById("demo").style.fontSize = "x-large";
```

Nell'esempio vediamo font-size -> fontSize

- Proprietà e metodi per aggiungere o cancellare elementi:

Metodo	Descrizione
<code>document.createElement(<i>element</i>)</code>	Crea un elemento HTML
<code>document.removeChild(<i>element</i>)</code>	Cancella un elemento HTML
<code>document.appendChild(<i>element</i>)</code>	Aggiunge un elemento HTML
<code>document.replaceChild(<i>new</i>, <i>old</i>)</code>	Rimpiazza un elemento HTML

Nel seguente esempio abbiamo creato un elemento button, abbiamo impostato il suo contenuto e infine lo abbiamo inserito in fondo all'elemento body

```
var btn = document.createElement("BUTTON"); // Create a <button> element
btn.innerHTML = "CLICK ME"; // Insert text
document.body.appendChild(btn); // Append <button> to <body>
```

Eventi

- Gli eventi sono importantissimi all'interno di Javascript.
- Abbiamo visto con Marcelloni che esistono vari modi per creare *listener*.
- Il metodo suggerito dal prof. Tesconi è l'`addEventListener`.

```
document.getElementById("myBtn").addEventListener("click", function() {
    document.getElementById("demo").innerHTML = "Hello World";
});
```

- Gli eventi più utilizzati sono i seguenti
 - o `click`, che può essere utilizzato non solo su bottoni ma anche su anchors, paragrafi, su un qualunque elemento HTML...
 - o `change`, si intercetta il cambiamento di una casella di input
 - o `scroll`, quando si fa scrolling della pagina (si usa l'ascensore per scendere o salire)
 - o `drag`, relativo allo spostamento di elementi
 - o `DOMContentLoaded`, evento che si scatena con il caricamento completato del DOM. Può essere utile quando la pagina da caricare è estremamente complessa: il DOM potrebbe richiedere più tempo per essere caricato e Javascript, nel frattempo, potrebbe provare a modificare un albero non ancora completato. La cosa può provocare errori.
 - o `resize`, ridimensionamento della pagina. Utile se vogliamo fare schermate che si adattano alla dimensione della finestra del browser.
- Lista completa degli eventi: https://www.w3schools.com/jsref/dom_obj_event.asp

Riscaldamento

```
<!DOCTYPE html>
<html lang="it">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>JS Bin</title>
  </head>

  <body>
    <input id="nome">
    <button id="mybtn">Click me</button>
    <div id="output"></div>

    <script type="text/javascript">
      var mybtn = document.getElementById("mybtn");
      var output = document.getElementById("output");

      // Quanto segue può essere scritto in due modi: il modo non commentato è quello
      // consigliato dal docente
      /*
      function hello() {
        var nome = document.getElementById("nome").value;
        output.innerHTML = "Hello " + nome;
      }

      mybtn.addEventListener("click", hello);
      */

      mybtn.addEventListener("click", function() {
        var nome = document.getElementById("nome").value;
        output.innerHTML = "Hello " + nome;
      });
    </script>
  </body>
</html>
```

- Il codice pone un input di testo in cui noi andiamo a indicare un nome
- L'`addEventListener` associa all'elemento button con id `mybtn` un *listener*: quando qualcuno clicca l'elemento si modifica il contenuto dell'elemento avente id `output`.
- Questo contenuto include il nome che abbiamo indicato nell'input

- Output:

lettore Click me
Hello lettore

setInterval() e setTimeout()

- Le seguenti funzioni sono importantissime per realizzare animazioni:
 - o `setInterval()`
chiama una funzione ogni tot millisecondi e continua ad invocarla fino a quando non viene fermata con `clearInterval()` o viene chiusa la finestra
 - o `setTimeout()`
chiama una funzione dopo tot millisecondi, salvo chiusura della finestra o chiamata della funzione `clearTimeout()`.

- **Esempio:**

```
var myVar;

function myFunction() {
    myVar = setTimeout(function(){ alert("Hello") }, 3000);
}

function myStopFunction() {
    clearTimeout(myVar);
}
```

Progetto libretto universitario

- Tesconi ha introdotto il progetto Libretto universitario creato dal prof. Tanganelli (docente che ha curato i laboratori degli anni passati)

- **File index.html**

```
<!DOCTYPE hmtl>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/x-icon" href="./css/img/favicon.ico" />
    <title>Libretto universitario</title>
    <link rel="stylesheet" href="./css/libretto.css" type="text/css" media="screen">
    <script type="text/javascript" src="./js/libretto.js"></script> <!-- js -->
  </head>
  <body onLoad="main()">
    <h2> Analisi statistica dei voti riportati negli esami </h2>
  </body>
</html>
```

L'index è molto semplice: è presente il riferimento al CSS e al Javascript, mentre nel body è presente solo il titolo della pagina. Il contenuto vero e proprio verrà inserito dalla funzione `main()`, triggerata dall'evento `onLoad`. Tesconi è insofferente sull'inserimento di tutto questo codice HTML attraverso Javascript: avrebbe preferito avere larga parte del codice direttamente nella index.

- **File js/libretto.js**

Andiamo ad analizzare i vari aspetti del progetto:

- o **Gestione di messaggi rivolti all'utente e agli errori:** possiamo raccogliere messaggi stampati all'interno del documento in un array. Questo permetterà di recuperarli in modo veloce. Tra questi messaggi possiamo porre anche quelli di errore: segue un esempio di funzione.

// Array di messaggi utente, istanziati prima che la pagina venga visualizzata

```
MESSAGGI_UTENTE = [
    "Inserisci una sequenza di voti tra 18 e 33",
    " non e` un numero",
    " e` un numero minore di 18 o maggiore di 33"
];
```

// gestore dei messaggi di errore

```
function stampaErrore(dato, codMess) {
    window.alert("Errore: '" + dato + "' " + MESSAGGI_UTENTE[codMess]);
}
```

- **root:**

- index.html
- js:
 - libretto.js
- CSS:
 - libretto.css
 - img:
 - contiene le immagini utilizzate dal CSS

- **Programmazione ad oggetti:** Tanganelli ha creato un oggetto di nome `Statistico`. Questo oggetto presenta attributi con i dati richiesti (quelli che stamperemo: minimo, massimo, media, variabilità) e funzioni utilizzate per calcolare i vari dati. L'associazione delle funzioni avviene con l'attributo `prototype`.

```
function Statistico(dati) {  
    this.voti    = this.analizzaDati(dati);  
    this.min     = 0;  
    this.mas     = 0;  
    this.med     = null;  
    this.variab  = null;  
}  
[... per le funzioni vedere più avanti]
```

- `analizzaDati()`

La funzione viene eseguita quando si inizializza un'istanza dell'oggetto. Data una stringa in ingresso restituisce

- un array con i voti se tutto va bene
- `null` nel caso siano stati trovati problemi (caratteri non numerici o voti non validi)

```
Statistico.prototype.analizzaDati = function(datiInput) {  
    // Array contenente i voti degli esami inseriti dall'utente (gli elementi sono delle stringhe)  
    var dati = datiInput.split(";");  
  
    // Array da restituire contenenti i voti degli esami (gli elementi sono degli interi)  
    var voti = new Array();  
  
    // Per ogni voto inserito dall'utente  
    for (var i = 0; i < dati.length; i++) {  
        var voto = Number(dati[i]);  
        if (isNaN(voto)) { // controllo se è un numero valido  
            stampaErrore(dati[i], 1);  
            return null;  
        }  
        // controllo se è compreso tra 18 e 33  
        else if (voto < 18 || voto > 33) {  
            stampaErrore(voto, 2);  
            return null;  
        }  
  
        voti[i] = voto;  
    }  
  
    return voti;  
}
```

- Con la `split` genero un array (poniamo come separatore il punto e virgola)
- Creo un `Array` `voti`
- Col `for` controllo gli elementi dell'array generato precedentemente
 - Se il singolo voto non è un numero stampo l'errore 1 e restituisco `null`.
 - Se il singolo voto non rispetta le convenzioni universitarie stampo l'errore 2 e restituisco `null`.
 - Se non si hanno errori il voto è valido e può essere incluso nell'array `voti`.
- Restituisco l'array `voti`.

- **Output:** riprendiamo il codice della funzione `main()`

```
function main() {
    var voti = window.prompt(MESSAGGI_UTENTE[0]);
    if (voti == null)
        return;

    var stat = new Statistico(voti);
    if (!stat.datOk())
        return;

    stat.calcolaMinimo();
    stat.calcolaMassimo();
    stat.calcolaMedia();
    stat.calcolaVariabilita();
    stat.stampa();
}
```

- Si richiede l'inserimento di una lista di voti con la funzione `prompt()` (Il testo è preso dall'array citato all'inizio).
- Si verifica che sia stato inserito del contenuto: in caso contrario ci si ferma subito.
- Si crea un'istanza di `Statistico` ponendo in ingresso la stringa `voti`.
- Si verifica con la funzione `datOk()` che tutti i dati siano corretti. In caso contrario mi fermo.
- Eseguo una serie di funzioni per calcolare minimo, massimo, media e variabilità.
- Eseguo un'ultima funzione per stampare i risultati e il contenuto vero e proprio della pagina HTML.

- `datOk()`
semplice funzione che verifica il valore in uscita della `analizzaDati()`. Dobbiamo solo verificare che l'uscita non sia *null*.

```
Statistico.prototype.datOk = function() {
    return this.voti != null;
}
```

- `calcolaMinimo()`
Si scorrono gli elementi dell'array membro `this.voti`, dall'ultimo al primo elemento. Ogni volta si aggiorna il minimo utilizzando la funzione `Math.min()`. Alla fine si aggiorna la variabile membro `this.min` ponendo il minimo trovato.

```
Statistico.prototype.calcolaMinimo = function() {
    var minimo = this.voti[0];
    for (var i = this.voti.length-1; i > 0 ; i--)
        minimo = Math.min(minimo, this.voti[i]);

    this.min = minimo;
}
```

- `calcolaMassimo()`
Stesso meccanismo di prima per trovare il massimo. Si utilizza la funzione `Math.max()`

```
Statistico.prototype.calcolaMassimo = function() {
    var massimo = this.voti[0];
    for (var i = this.voti.length-1; i > 0 ; i--)
        massimo = Math.max(massimo, this.voti[i]);

    this.mas = massimo;
}
```

o `calcolaMedia()`

funzione che restituisce un array contenente due valori:

- Media quantitativa (valore numerico)
- Media qualitativa (stringa contenente un giudizio deciso sulla base

```
Statistico.prototype.calcolaMedia = function() {  
    /*****  
        CALCOLO MEDIA QUANTITATIVA  
        *****/  
    var i = 0, media = 0;  
    while (i < this.voti.length) {  
        media += this.voti[i];  
        i++;  
    }  
    media /= this.voti.length;  
    media = Math.round(media*100)/100;  
  
    /*****  
        CALCOLO MEDIA QUALITATIVA  
        *****/  
    var mediaQual = null;  
    switch(Math.floor((media-18)/3)) {  
        case 0:  
            mediaQual = "sufficiente";  
            break;  
        case 1:  
            mediaQual = "discreta";  
            break;  
        case 2:  
            mediaQual = "buona";  
            break;  
        case 3:  
            mediaQual = "distinta";  
            break;  
        case 4:  
            mediaQual = "ottima";  
            break;  
        default:  
            mediaQual = "eccellente";  
    }  
  
    this.med = { numerica: media, qualitativa: mediaQual };  
}
```

- Per il calcolo della media quantitativa:
 - Sommo tutti i voti presenti nell'array `this.voti`
 - Divido per il numero di voti (lunghezza dell'array, `this.voti.length`)
 - Con la `Math.round()` arrotondo a due cifre decimali.
 - Moltiplico per 100, spostando la virgola di due posizioni avanti.
 - Arrotondo con la `round` (cambia eventualmente l'ultima cifra intera)
 - Divido per 100, spostando la virgola di due posizioni indietro.
 - Ottengo un numero che ha solo due cifre decimali.
- Per la media qualitativa:
 - Prendo la media quantitativa, sottraggo 18, divido per 3 e arrotondo per difetto con la `floor`.
 - Ottengo un numero compreso tra 0 e 5. Vediamo i casi limite:
 - 0: $(18-18)/3=0$
 - 5: $(33-18)/3=5$
 - Con la `switch` individuo quale valore qualitativo si addice alla mia media.

- o `calcolaVariabilita()`
funzione che restituisce un array contenente due valori
 - Variabilità quantitativa (valore numerico)
 - Variabilità qualitativa (stringa)

Con variabilità intendiamo la seguente formula

$$\frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

Dove x_i consiste nell' i -esimo elemento, \bar{x} nella media ed n nel numero di elementi coinvolti. Si considera una somma di valori assoluti per evitare che alcuni termini si annullino.

```

Statistico.prototype.calcolaVariabilita = function() {
  /*****
    CALCOLO VARIABILITA' QUANTITATIVA
    *****/
  var i = 0, varia = 0;
  do {
    varia += Math.abs(this.voti[i]-this.med.numerica);
    i++;
  } while (i < this.voti.length);

  varia /= this.voti.length;
  varia = Math.round(varia*100)/100;

  /*****
    CALCOLO VARIABILITA' QUALITATIVA
    *****/
  var variabQual = null;
  switch(Math.ceil(varia/7.5*3)) { // 7.5 massima variabilita
    case 1:
      variabQual = "bassa";
      break;
    case 2:
      variabQual = "normale";
      break;
    case 3:
      variabQual = "alta";
      default:
      variabQual = "nessuna";
      break;
  }
  this.variab = { numerica: varia, qualitativa: variabQual };
}

```

- Per la varianza quantitativa:
 - Sommo i valori assoluti (utilizzando la `Math.abs`) delle differenze viste prima.
 - Divido per il numero di elementi dell'array.
 - Arrotondo alla seconda cifra esattamente come abbiamo fatto prima.
- Per la varianza qualitativa:
 - Divido la varianza quantitativa per 7.5, moltiplico per 3 e arrotondo per eccesso con la funzione `ceil`.
 - Con la switch individuo un valore qualitativo che si addice alla mia varianza quantitativa.

- o stampa()
ultima funzione eseguita nella main(), permette di stampare il contenuto della pagina includendo i dati inseriti e i risultati calcolati poco prima. All'interno sono chiamate due ulteriori funzioni: stampaTabellaVoti() e stampaDatiStatistici()

```
Statistico.prototype.stampa = function() {
    // Stampa intestazione pagina HTML (doctype e head)
    document.writeln("<!DOCTYPE html>");
    document.writeln("<html><head><meta charset=\"utf-8\">");
    document.writeln("<link rel=\"shortcut icon\" type=\"image/x-icon\"");
    document.writeln("href=\"./css/img/favicon.ico\"/>");
    document.writeln("<title>Libretto universitario</title>");
    document.writeln("<link rel=\"stylesheet\" href=\"./css/libretto.css\"");
    document.writeln("type=\"text/css\" media=\"screen\"> <!-- css --></head>");

    // Qua inizia il <body>
    document.writeln("<body>");
    document.writeln("<div id=\"wrapper\">");
    document.writeln("<div id=\"topnav\"><img");
    document.writeln("src=\"./css/img/unipi_logo.png\" alt=\"Logo\"></div>");
    document.writeln("<p>Libretto Universitario</p>");

    this.stampaTabellaVoti();
    this.stampaDatiStatistici();

    document.writeln("</div>");
    document.writeln("</body>");
    document.writeln("</html>");
}
```

- o stampaDatiStatistici()
stampa dei risultati calcolati con le funzioni spiegate prima della stampa()

```
Statistico.prototype.stampaDatiStatistici = function() {
    document.writeln("<div id=\"datiStatistici\">");
    document.writeln("Minimo: " + this.min + "<br>");
    document.writeln("Massimo: " + this.mas + "<br>");
    document.writeln("Media: " + this.med.numerica + " (" +
        this.med.qualitativa + ")<br>");
    document.writeln("Variabilit&agrave;; " + this.variab.numerica + " (" +
        this.variab.qualitativa + ")");
    document.writeln("</div>");
}
```

- o stampaTabellaVoti()
stampa dei valori inseriti con la prompt() sottoforma di tabella

```
Statistico.prototype.stampaTabellaVoti = function() {
    document.writeln("<div id=\"tabellaVoti\">");
    document.writeln("<table>");
    document.writeln("<caption>Elenco Esami</caption>");
    document.writeln("<tr><th>Voti");

    for (var i = 0; i < this.voti.length; i++)
        document.writeln("<tr><td>" + this.voti[i]);

    document.writeln("</table>");
    document.writeln("</div>");
}
```


Esercizio: bottoni che aumentano/riducono la dimensione del font

```
<!DOCTYPE html>
<!--
Created using JS Bin
http://jsbin.com

Copyright (c) 2020 by mtesconi (http://jsbin.com/fefumam/2/edit)

Released under the MIT license: http://jsbin.mit-license.org
-->
<html>
  <head>
    <meta name="description" content="cambia dimensione font">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>cambia dimensione font</title>
    <style id="jsbin-css">
      #sentence {
        padding: 20px;
        font-size: 20px;
      }
    </style>
  </head>
  <body>
    <button id="plus">+</button>
    <button id="minus">-</button>
    <div id="sentence">HELLO WORLD!!!</div>

    <script id="jsbin-javascript">
      var fontSize = 20;

      document.getElementById("plus").onclick = function(){
        fontSize++;
        document.getElementById("sentence").style.fontSize = fontSize + "px";
      };

      document.getElementById("minus").onclick = function(){
        fontSize--;
        document.getElementById("sentence").style.fontSize = fontSize + "px";
      };
    </script>
  </body>
</html>
```

- Abbiamo la frase Hello world!!! che inizialmente ha dimensione di 20px (impostata da CSS)
- Abbiamo inserito due bottoni: uno con id plus e uno con id minus.
- Abbiamo associato delle funzioni a due eventi:
 - o click sul bottone con id plus: funzione che incrementa la dimensione del font;
 - o click sul bottone con id minus: funzione che decrementa la dimensione del font;
- Entrambe le funzioni agiscono tenendo conto di una variabile `fontSize` inizializzata a 20 (la stessa dimensione del carattere impostata attraverso CSS. Dopo aver incrementato o decrementato la variabile aggiornano la proprietà del CSS col nuovo valore (si concatena la variabile aggiornata a px)

- Esempio di output:



HELLO WORLD!!!

HELLO WORLD!!!

HELLO WORLD!!!

Esercizio: bar chart

Il seguente esercizio mira a creare un bar chart, cioè un diagramma a barre il cui contenuto è stabilito da un input. Precisamente: si pone una lista di numeri, separati da virgola, in ingresso; si crea una barra per ciascun numero; ciascuna barra avrà una lunghezza in px pari al numero indicato.

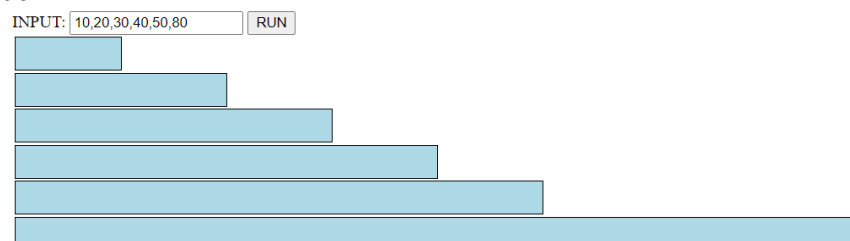
```
<!DOCTYPE html>
<html>
  <head>
    <meta name="description" content="Esercitazioni con Javascript + CSS">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>Esercitazioni con Javascript + CSS</title>
    <style id="jsbin-css">
      .bar {
        margin: 2px;
        border: 1px solid black;
        width: 100px;
        height: 30px;
        background-color: lightblue;
      }
    </style>
  </head>
  <body>
    INPUT: <input id="input"></input>
    <button id="button">RUN</button>
    <div id="output"></div>

    <script id="jsbin-javascript">
      var button = document.getElementById("button");
      var output = document.getElementById("output");

      button.onclick = function() {
        //output.innerHTML = "HELLO WORLD!!!";

        var numbers = input.value.split(",");
        numbers.forEach(function(value) {
          div = document.createElement("div");
          div.setAttribute("class", "bar");
          div.style.width = parseInt(value) * 10 + "px";
          output.appendChild(div);
          //console.log(parseInt(value) * 100 + "px");
        });
      };
    </script>
  </body>
</html>
```

- Associo al click del pulsante con id `button` una funzione:
 - o Genero un array dividendo la stringa posta in ingresso con l'input (la virgola è il separatore)
 - o Per ogni numero presente nell'array generato:
 - Creo un nuovo div
 - Associo al div appena creato la classe `bar` (che presenta certe proprietà grafiche - CSS)
 - Imposto la lunghezza della barra alterando il `width`.
 - Aggiungo la barra nel contenuto del div `output`.
- **Effetto collaterale:** con la funzione `append` concateniamo nuovo codice con quello già presente all'interno del div class `output`. Questo significa che nuovi dati comporteranno l'aggiunta di nuove barre senza eliminare le precedenti (a meno che non si ricarichi la pagina).
- **Esempio di output:**



Esercizio: indovina il numero

Proviamo a realizzare un gioco molto semplice dove l'utente deve indovinare un numero. Il computer, al caricamento della pagina, genera un numero random. L'utente prova a indovinarlo attraverso un input: la pagina risponde dicendo se ha indovinato (mostrando il num. di tentativi) o se ha indicato un numero maggiore o minore.

```
<html>
  <head>
    <meta name="description" content="indovina il numero">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>JS Bin</title>
  </head>
  <body>
    <h3>INDOVINA IL NUMERO</h3>
    <input id="numero" type="text">
    <button id="button">TEST</button>
    <div id="output"></div>

    <script>
      var num = Math.floor(Math.random() * 100); // returns a random integer from
      0 to 9
      var output = document.getElementById("output");
      var button = document.getElementById("button");

      var num_tentativi = 0;

      // La console è molto utile per fare debugging: col comando console.log(num);
      possiamo stampare il numero random generato

      button.onclick = function() {
        num_tentativi++;
        console.log(num_tentativi);

        var tentativo = document.getElementById("numero").value;
        if (tentativo == num)
          output.innerHTML = "HAI INDOVINATO CON "+num_tentativi+" TENTATIVI";
        else if (tentativo > num)
          output.innerHTML = tentativo + " è più grande";
        else
          output.innerHTML = tentativo + " è più piccolo";
      };
    </script>
  </body>
</html>
```

- Con le funzioni `Math.floor` e `Math.random()` generiamo il numero random. Ricordiamoci quanto detto durante la lezione di Marcelloni relativamente al generare numeri random.
- Si utilizza la `getElementById` per ottenere gli oggetti del DOM da manipolare (il div con id `output` dove stamperemo le risposte del gioco e il bottone con id `button` che premiamo per inviare gli input)
- Si inizializza con valore 0 una variabile contatore `num_tentativi` per ricordarsi il numero di tentativi usati (sarà stampata in caso di vittoria)
- Si associa una funzione all'evento `onclick`:
 - o Incremento i tentativi
 - o Stampo il numero di tentativi nella console (questo è per comodità nostra, per debuggare)
 - o Con `getElementById("numero").value` estraiamo il valore posto nell'input
 - o Con una serie di *if-statement* confronto il numero inserito con quello generato da Javascript e agisco di conseguenza modificando il contenuto del div `output`.
- Esempio di output:

INDOVINA IL NUMERO

5

5 è più piccolo

INDOVINA IL NUMERO

50

50 è più grande

INDOVINA IL NUMERO

33

HAI INDOVINATO CON 11 TENTATIVI