

Lacerba.io: Corso di jQuery

NECESSARIO PER RENDERE DINAMICO UN SITO

ovvero se vogliamo inserire interattività per L'User Interface

esempio :

<http://lookbook.quechua.com/spring-summer-2016/it/escursione>

con effetti parallax, menu a scomparsa

elementi dinamici

i **lightbox** ovvero **GALLERIA IMMAGINI**

i **jQuery Carousel**

Sticky Navigation => scorrendo il NAV cambia

Effetti ottenibili anche con JScript

jQuery è una LIBRERIA DI JS OVVERO UNA COLLEZIONE DI BLOCCHI DI CODICE

Il cui fine è semplificare la programmazione front-end

è compatibile con tutti i browser in circolazione

ci permette di fare 2 COSE FONDAMENTALI:

1. SELEZIONARE ELEMENTI DELLA PAGINA, COI SELETTORI CSS
2. CI PERMETTE DI COMPIERE AZIONI CON GLI ELEMENTI SELEZIONATI, manipolandoli, creando animazioni, modificandoli in seguito ad un click del mouse tutto questo senza bisogno di scrivere un **codice alternativo il cosiddetto FULLBACK CODE dei browser + vecchi**

slogan jQuery

WRITE LESS, DO MORE

è un file JS da includere in una pagina web

ACCEDERE AD UN ELEMENTO PAGINA

`_Sintassi`

`jQuery(selettore)`

oppure

`$(selettore)`

si crea così l'oggetto jQuery che ci permette di accedere all'elemento

Es:

`$('#li .icon')`

FAR COMPIERE AZIONI ALL'ELEMENTO CON UN METODO JQUERY

Metodi Reference

[Api Doc](#)

trovi tutti i metodi jQuery

- NASCONDERE UN OGGETTO NELLA PAGINA

`$('#li .icon').hide()`

mediante il metodo [hide](#) (trovi anche esempi del metodo)

molto utile scrivere commenti al codice per stabilire gli step che dobbiamo fare

restituisce l'oggetto jQuery (che è una collezione di oggetti del DOM)

puoi o meno inserirgli parametri

- FAR RIAPPARIRE UN ELEMENTO

`$('#li .icon').show()`

col metodo [show](#) (trovi anche esempi su esso e con uso parametri)

per avere un effetto graduale nel comparire gli passo dei parametri

`$('#li .icon').show('slow')`

IN JQUERY PUOI ANCHE CONCATENARE I METODI

```
$(‘li .icon’).hide().show(‘slow’);
```

Come scaricare sublime text: <http://lacerba.io/courses/html-css/lectures/615795>

DOM (Document Object Model)

rappresentazione in nodi di una pagina web, contenuto della pagina viene quindi memorizzato nel dom a cui si può accedere e che si può manipolare con js e jQuery

TIPI DI METODI IN jQuery

3 categorie:

- **DOM MANIPULATION METHODS:** per cambiare il Dom e dunque:
 - **Aggiungere, rimuovere o trasformare elementi HTML**
 - **Aggiungere, rimuovere o trasformare attributi degli elementi HTML**
 - **Trasformare gli elementi HTML con opportuni effetti**
- **DOM TRAVERSING METHODS:** per muoverci dentro al Dom e passare da un elemento all'altro
 - **Passare da un elemento “parent” ad un elemento “child”**
 - **Passare da un elemento “child” ad un elemento “parent”**
 - **Passare da un elemento ad un altro elemento adiacente**
- **EVENT METHODS:** ci permettono di creare effetti in seguito ad un evento che sia:
 - **CLICK DEL MOUSE**
 - **DIGITAZIONE DI CARATTERI SU TASTIERA**
 - **SCROLL DEL MOUSE**

[Qui](#) trovi la documentazione completa di tutti i metodi jQuery

sono suddivisi in:

- Manipulation

[Class Attributes](#)

agiscono su classi elementi

[Attributi Generici](#)

.attr()

ti permette o di ottenere valore o inserire valori ad un attributo

[CSS](#)

.css()

ottiene lo stile dell'elemento su cui la invochi

TRAVERSING METHODS

permettono di muoverci dentro al DOM

.children

restituisce tutti i figli dell'elemento selezionato

last()

restituisce solo ultimo elemento selettore

EVENTS METHODS

ci sono una serie di eventi

Mouse Events

.click

manipolo un elemento solo se utente clicca su esso

Effetti base

.hide

.show

.fadeIn per sfumare ele etc..

06_Dove inserire jQuery

due modi per includerlo

o [scaricarlo](#) in locale (il file compresso o non compresso) ed inserirlo nella cartella del progetto e dentro all'html, posso metterlo poi nell'head prima del **file js locale**

oppure usare un [CDN](#) (Content Delivery Network)

lui usa CDN di [GOOGLE](#)

vantaggi in termini di velocità perché jQuery SPESSO E NELLA CACHE DEL NOSTRO BROWSER GRAZIE A NAVIGAZIONI PRECEDENTI

**LO SCRIPT VA MESSO IN FONDO AL BODY (preferibili guadagni in tempo e nn devi usare metodo .ready)
OPPURE USARE IL METODO**

```
document.ready(function(){  
    istruzioni da svolgere a caricamento pagina  
})
```

effettuo azione slow col DOM caricato

Funzioni: <https://developer.mozilla.org/en-US/docs/Web/JavaS...>

Lezione funzioni: <http://lacerba.io/courses/corso-javascript/lecture...>

APPROCCIO 4P PROBLEM SOLVING

PER PROGRAMMARE IN JQUERY O QUALSIASI PROGETTO DI DESIGN

metodo di lavoro con 4 **fasi**:

1. **PREPARAZIONE**: Ove capire il problema e pensare a eventuale sol, in pratica riassumi il problema e ne scrivi una sommaria sol
2. **PLAN** (PIANIFICAZIONE): in cui prepariamo la sol al nostro problema, scrivi in punti, spesso commenti, le varie fasi della sol
3. **PERFORM**: **fase in cui mettiamo in pratica la sol che abbiamo preparato**
4. **PERFECT**: fase in cui correggiamo o perfezioniamo il progetto

Es: Problema schiaccio pulsante e appare scritta al suo posto

aggiungere un elemento

MANIPULATION METHOD -> DOM INSERTION INSIDE

`.append(ele da inserire)`

AGGIUNGE UN ELE DENTRO A ELE SELEZIONATO, IN FONDO AL SUO CONTENUTO

`.prepend(ele da inserire)`

AGGIUNGE UN ELE DENTRO A ELE SELEZIONATO, ALL'inizio AL SUO CONTENUTO

`$(ele da inserire).appendTo('selettore contenitore')`

CLICK BOTTONE

EVENTS METHOD -> MOUSE EVENTS -> CLICK

`.click(event handler)`

collega un event handler al click, passo come handler una funzione che si esegue ogni volta che si innesca l'evento

`$(sel).click(function(){`

```
}}
```

RIMUOVERE UN ELEMENTO

Manipulation -> DOM removal -> remove

```
$('sel ele da rimuovere').remove();
```

```
$(this)
```

PERMETTE DI **APPLICARE FUNZIONE SOLO AD ELEMENTO IN SCOPE QUELLO CHE HA INVOCATO LA FUNZIONE**

esempio se ho invocato un click sul bottone il remove interno all'handler verrà fatto solo al bottone su cui si è cliccato

per referenziare un elemento precedente ad un altro che è ns selettore uso un TRAVERSING METHOD

```
.prev()
```

ci fornisce elemento precedente al selezionato, buono usare il this

IMMAGINE MOSTRATA IN ZOOM SENZA LASCIARE LA PAGINA, creare un LIGHTBOX

clicchi su img e invece di aprirti scheda all'indirizzo immagine aprirla come un modal e quando clicchi fuori dall'img ritorna alla pagina

sol: sovrapporre immagine ad uno **sfondo nero**, tecnica molto comune a molti siti web e si chiama **lightbox**

overlay => sfondo scuro

Vediamo le varie fasi

PREPARATION

problema: quando clicco su un'immagine vengo

rimandato ad un'altra pagina

soluzione: sovrapporre l'immagine ad uno sfondo

nero - lightbox

PLAN

Catturare il click dell'utente sull'immagine

Mostrare lo sfondo scuro (overlay)

Mostrare sopra l'overlay l'immagine cliccata

Mostrare il testo sopra l'overlay

Aggiungere sulla pagina l'overlay

Aggiungere l'immagine

Aggiungere il testo

prevenire il comportamento di default della pagina, ad esempio l'apertura del link

preventDefault

For example, clicked anchors will not take the browser to a new URL

quindi lo copio dentro la funzione

```
event.preventDefault();
```

dobbiamo anche recuperare il link all'immagine che ci servirà dopo
devo catturare il contenuto di **href**

lo faccio con [attr](#) che ci da il valore di un attributo

.attr('attr di cui vogliamo il valore')

l'overlay lo mettiamo in fondo al body, prima del sui tag di chiusura (ma prima dello script)

col css gli diamo un position fixed, uno z-index sopra tutti gli elementi, dobbiamo aggiungerlo tramite jquery

```
$('#body').append(overlay);
```

convenzione NOMINARE LE VAR CHE CONTENGONO UN OGGETTO JQUERY COL SIMBOLO \$ DAVANTI

es:

```
var $overlay = $('<div id="overlay"></div>');
```

css overlay

```
#overlay {  
    background: rgba(0,0,0,.7);  
    width: 100%;  
    height: 100%;  
    position: fixed;  
    top: 0;  
    left: 0;  
    display: none;  
}
```


MOSTRARE L'IMG SULL'OVERLAY

creo una var pure per l'img

```
var $immagine = $('<img>');
```

aggiungo immagine all'overlay

```
$overlay.append($immagine);
```

aggiungo con **attr()** l'src all'immagine

```
$($immagine).attr('src', href);
```

SELEZIONARE UN ELEMENTO CHILDREN

usiamo il traversing Method [children](#)

seleziona gli elementi children dell'ele che lo invoca

gli puoi passare anche un **parametro con un attributo per individuare una corrispondenza esatta**

INSERIRE DEL TESTO DENTRO AD UN ELEMENTO

```
.text()
```

ti permette o di leggere o di inserire del testo da un elemento

quindi ricapitolando per generere effetto lightbox

```
// PREPARATION
```

```
// problema: quando clicco su un'immagine vengo
```

```
// rimandato ad un'altra pagina
```

```
// soluzione: sovrapporre l'immagine ad uno sfondo
```

// nero - lightbox

// PLAN

// Catturare il click dell'utente sull'immagine

```
$('#gallery a').click(function(){
```

```
    // evito che i link vengano aperti alla vecchia
```

```
    event.preventDefault();
```

```
    // recupero il valore dell'attributo dell'ele che clicchiamo
```

```
    var href = $(this).attr('href');
```

```
    // Mostrare lo sfondo scuro (overlay)
```

```
    $($overlay).show();
```

```
    // Mostrare sopra l'overlay l'immagine cliccata
```

```
    $($immagine).attr('src', href);
```

```
    $($immagine).show();
```

```
    // Mostrare il testo sopra l'overlay
```

```
    // seleziono tutte le img del tag a che clicco
```

```
    var testoDescrizione = $(this).children('img').attr('alt');
```

```
    $descrizione.text(testoDescrizione);
```

```
});
```

// dato che mi servirà spesso catturo l'ele in una var

// uso \$() per avere l'oggetto jquery

```
var $overlay = $('<div id="overlay"></div>');
```

```
// creo la var anche per l'img che andremo ad mettere sopra  
l'overlay
```

```
var $immagine = $('<img>');
```

```
// creo var che conterrà il valore dell'attributo alt dell'img
```

```
var $descrizione = $('<p></p>');
```

```
// Aggiungere sulla pagina l'overlay
```

```
$('body').append($overlay);
```

```
    // Aggiungere l'immagine
```

```
$overlay.append($immagine);
```

```
    // Aggiungere il testo, la descrizione dell'immagine
```

```
    // quello che ci sta dentro ad alt
```

```
$overlay.append($descrizione);
```

```
// Quando clicco sull'overlay
```

```
$overlay.click(function(){
```

```
    //nascondere l'overlay
```

```
    $(this).hide();
```

```
})
```

CREARE UN DROPDOWN MENU'

utile soprattutto in vista mobile (schermi piccoli) se hai una navbar con + voci, una select per accedere alle varie parti del sito

un menù dropdown si inserisce in html col tag [select](#) che include i tag option, le varie opzioni del dropdown menu

Quindi, la prima cosa da fare è inserire il tag select dentro al nav (dopo la **ul** con le varie voci, dentro il nav, poi provvederemo per gli schermi piccoli a far sparire la lista ul) tutte queste operazioni dobbiamo farle con jQuery

19_1 loop in jQuery

Per creare un loop in jQuery si usa il metodo [each\(\)](#) che esegue **1 o + funzioni per ogni elemento selezionato**

Esempio

```
$('#nav a').each(function(){  
    operazioni  
});
```

Per recuperare il testo o contenuto di un elemento

```
$('#selettore').text()
```

Per inserire il **value** di un elementon uso il metodo [val\(\)](#)

```
$option.val($(this).attr());
```

jQuery reindirizzare ad un link

[window.location](#)

per indirizzare l'utente su una parte della pagina, ad esempio all'option selezionato da select

window.location =\$select.val());

Loop: <http://lacerba.io/courses/corso-javascript/lecture...>

PERFEZIONIAMO DROPDOWN MENU TOGLIENDO IL BOTTONE MA SEMPLICEMENTE SELEZIONANDO LA SEZIONE DAL SITO VENIRE INDIRIZZATO

in pratica lanciare il link una volta selezionata una voce della **select**

vediamo i Form Event e trovo il **change()** metodo che si occupa solo di elementi input, textarea e select

```
$('#select').change(function(){  
    istr  
});
```

21_Modificare il CSS

Media queries; <https://developer.mozilla.org/en-US/docs/Web/CSS/M...>

COSA SONO E DOVE TROVARE I PLUGIN

jQuery semplifica ancora di **+ la creazione di elementi dinamici di user-interface grazie ai plugin**

ci sono un botto di plugin in giro per i + svariati fini:

- creare menu di navigazione dinamici
- lightbox
- background dinamici
- slider di immagini

ovvero **applicazioni che con jQuery o JS richiederebbero capacità abbastanza elevate**

coi plugin ti semplifichi vita

Vediamo come:

- **trovare e usare plugin**

ne applichiamo 3:

- **plugin per effettuare effetti di transizione fra una pagina e l'altra**

- **sticky navigation: una barra di nav che rimane sul bordo sup pagina quando si scrolla**

- **uno slider di immagini**

vedremo struttura base di un plugin

COME CERCARE PLUGIN?

Googola ad esempio jQuery plugin Sticky nav

<http://stickyjs.com/>

oppure esistono tanti siti che ci danno elenco dei vari plugin

<http://www.unheap.com/>

ci da plugin divisi per sezioni, quasi tutti con link a doc e demo ad esempio per effetti in scroll

<https://scrollrevealjs.org/>

crea animazioni quando scendiamo nella pagina e saliamo nella pagina, ho il link a github dove ci sta la [doc](#)

COME SCEGLIERE IL PLUGIN:

- **Scegliere solo quelli dei quali si ha veramente bisogno**
se ne mettiamo troppi **rallenta l'esperienza dell'utente NON ABUSARNE**

- **deve avere una buona documentazione**

- **deve essere abbastanza recente (non anni)**

- **DEVE ESSERE RESPONSIVE (LO VEDI RIDUCENDO VIEWPORT)**

un altro sito è

<https://www.sitepoint.com/jquery-popular-plugins-list/>

molti plugin

anche sul sito jquery ci sono molti plugin divisi per categorie

<https://plugins.jquery.com/>

anche se unheap è + aggiornato

STRUTTURA DI UN PLUGIN

sono molto diversi tra loro in complessità ma **hanno tutti una struttura simile, quindi una volta capita è facile passare da un plugin all'altro**

Questa la loro struttura:

- **hanno un file CSS**: descrive il design degli elementi che JS inserisce nella pagina (**opzionale può non esserci il CSS**)

esempio il plugin lightbox

<http://lokeshdhakar.com/projects/lightbox2/>

uno dei + famosi per creare effetti lightbox, il design degli elementi dell'effetto ovvero il bordo dell'img in lightbox, la freccia oppure la x per chiudere sono **nel css del plugin**

- **hanno un file JS** dove viene **inserito il codice jQuery** che fa eseguire il plugin (**OBBLIGATORIO, c'è sempre**)
- **molti hanno pure una cartella con le immagini** (che si usano nel plugin se vengono usate) (opzionale può non esserci)

ANIMATION

plugin che crea animazioni nelle transizioni fra una pagina e l'altra (sfumare, roteare ...)

una volta scelto il plugin si scaricano i suoi file

lo scarico da [github](#)

il README è molto importante per capire come usare il tutto, installarlo e usarlo

la cartella dist è la + importante perché è la distribuzione del plugin ove ci sono i file che dobbiamo inserire nella cartella del nostro sito (nel nostro caso js e css)

fai il download dello zip

i file min sono compressi, occupano meno spazio e ti fanno ottenere un caricamento + veloce

creo una cartella dentro il js e il css per inserire la versione minimizzata del file js e css

facciamo in sto modo al fine di ottenere un po' di ordine fra i vari plugin

AGGIUNGERE IL PLUGIN AL SITO

apri le pagine html del sito, anzitutto inseriamo il css

INSERIRLO PRIMA DEI FILE main.css e responsive.css

perché il plugin potrebbe essere personalizzato e quindi è buona cosa sovrascrivere il js o css del plugin dentro a **main.css**

inserire il js **prima del nostro file js script.js** ma dopo quello **jquery**

COME MODIFICARE L'HTML PER FAR FUNZIONARE IL PLUGIN

STEP 2: HTML MARKUP

```
<body>
  <div class="animsition">
    <a href="./page1" class="animsition-link">animsition link 1</a>
    <a href="./page2" class="animsition-link">animsition link 2</a>
  </div></body>
```

ci spiega di inserire un div con la classe **animsition** per tutto il contenuto per il quale vogliamo applicare il plugin

e all'interno dei tag a per andare da una pagina all'altra occorre inserire una class **.animsition-link**

se voglio che tutta la pagina sia influenzata dalla animazione in transizione metto un div che va da apertura body a dopo a footer con class **.animsition**

poi dentro al menu di navigazione aggiungere la classe **.animsition-link** agli ele a che si collegano ad un'altra pagina
questo deve essere fatto a tutte le pagine html

una volta fatta questa operazione dobbiamo andare ad inserire dentro **script.js** il codice per avviare l'effetto

STEP 3: CALL THE ANIMSITION

```
$(document).ready(function() {
  $(".ansition").ansition({
    inClass: 'fade-in',
    outClass: 'fade-out',
    inDuration: 1500,
    outDuration: 800,
    linkElement: '.ansition-link',
    // e.g. linkElement: 'a:not([target="_blank"]):not([href^="#"])'
    loading: true,
    loadingParentElement: 'body', //ansition wrapper element
    loadingClass: 'ansition-loading',
    loadingInner: '', // e.g. ''
    timeout: false,
    timeoutCountdown: 5000,
    onLoadEvent: true,
    browser: [ 'animation-duration', '-webkit-animation-duration'],

    // "browser" option allows you to disable the "ansition" in case the css property

    // The default setting is to disable the "ansition" in a browser that does not sup
    duration".
    overlay : false,
    overlayClass : 'ansition-overlay-slide',
    overlayParentElement : 'body',
    transition: function(url){ window.location.href = url; }
  });
});
```

quello che ci interessa è

```
$(".ansition").ansition({
```

ed il metodo ansition

il document.ready nn interessa perché ho messo il link al file alla fine del body

tutte le opzioni sotto sono già in essere con il plugin **e servono per personalizzarlo**

quindi inserire

```
$(".animsition").animsition();
```

posso mettere l'animazione anche in blocchi della pagina, basta spostare il `<div class="animsition">`

MODIFICARE LE OPZIONI DEL PLUGIN

per personalizzarlo, ad esempio questo ha vari effetti
nella doc dice

STEP 3: CALL THE ANIMSITION

```
$(document).ready(function() {
  $(".animsition").animsition({
    inClass: 'fade-in',
    outClass: 'fade-out',
    inDuration: 1500,
    outDuration: 800,
    linkElement: '.animsition-link',
    // e.g. linkElement: 'a:not([target="_blank"]):not([href^="#"])'
    loading: true,
    loadingParentElement: 'body', //animsition wrapper element
    loadingClass: 'animsition-loading',
    loadingInner: '', // e.g. ''
    timeout: false,
    timeoutCountdown: 5000,
    onLoadEvent: true,
    browser: [ 'animation-duration', '-webkit-animation-duration'],
    // "browser" option allows you to disable the "animsition" in case the cs
    s property in the array is not supported by your browser.
    // The default setting is to disable the "animsition" in a browser that d
    oes not support "animation-duration".
    overlay: false,
    overlayClass: 'animsition-overlay-slide',
    overlayParentElement: 'body',
    transition: function(url) { window.location.href = url; }
  });
});
```

riporta numerose voci che sono tanti elementi, ognuno è **un oggetto JavaScript (un OGGETTO LETTERALE)**

Con key value

usati nei plugin per modificare il comportamento di default del plugin

```
inClass: 'fade-in',  
outClass: 'fade-out',
```

determinano il modo in cui inizia e finisce transizione

Es:

```
$(".animsition").animsition({  
  inClass: 'rotate-in',  
  outClass: 'rotate-out'  
});
```

possiamo anche modificare la durata della transizione con

```
inDuration: 1500,  
outDuration: 800,
```

+ alto è il valore + la transizione sarà lenta

Es:

```
$(".animsition").animsition({  
  inClass: 'rotate-in',
```

```
    outClass: 'rotate-out',  
    inDuration: 1500,  
    outDuration: 1500  
});
```

un'altra ptty importante è il **linkElement**

```
linkElement: '.animsition-link',
```

che ci permette di scegliere a quali **link applicare il plugin**
se non lo scegliamo applica il plugin a tutti gli elementi con la
classe **.animation-link**

ad esempio usando il semplice selettore 'a', mi evito di definire le
classi dentro l'html

Es:

```
$(".animsition").animsition({  
    inDuration: 500,  
    outDuration: 500,  
    linkElement: 'a',  
});
```

STICKY NAVIGATION

scorrendo l'header rimane fisso e **cambia il background e
possiamo anche nascondere degli elementi scorrendo**

tutto questo lo si può fare installando il plugin e facendo opportune modifiche al codice, introdurremo il concetto di **EVENTO NEL PLUGIN** ovvero modificare il comportamento del programma a seguito di eventi

ES: lo **scroll**

useremo <http://stickyjs.com/>

che è **stickyjs.com**

per ottenere una sticky nav

vediamo la documentazione su [github](#)

è solo un file **js** non ci sono ne css ne immagini

la documentazione dice di includere il link a **jquery** ed al **plugin** ed attivare il plugin in javascript

scarichiamo lo zip

prendiamo solo il js ovvero **jquery.sticky.js**

e lo mettiamo dentro alla cartella js creando una cartella **sticky**

una volta inserito il plugin nella cartella lo inserisco nel file html

prima dello script personale

```
<script src="js/sticky/jquery.sticky.js"></script>
```

poi nel nostro file personale **script.js** chiameremo il plugin

basta selezionare l'elemento che vogliamo rendere sticky e

```
$("#ele").sticky({topSpacing:0});
```

topSpacing:0 è il valore default quindi possiamo nn inserirlo

Es:

```
$(".main-header").sticky();
```

Bisogna dare un effetto grafico quando si inizia a scrollare, questo lo si fa grazie alla struttura che il js crea ovvero un div con id=sticky-wrapper che va a contenere l'elemento che abbiamo indicato come sticky, e class **sticky-wrapper** e quando si **scrolla** si crea un'ulteriore classe chiamata **is-sticky** dobbiamo quindi usare questa classe come selettore per dare delle ptty css quando scrollo voglio dare un background bianco e scritte nere

```
.is-sticky .main-header {  
    background-color: #fff;  
}
```

```
.is-sticky a, .is-sticky h1, .is-sticky h2 {  
    color: #000;  
}
```

inoltre gli diamo una **transition** di modo che l'effetto di cambio colore in scroll sia + **dolce e lento**

```
.main-header {  
    transition: 0.5s;  
}
```

Quando scrolliamo vogliamo che il sottotitolo dell'header scompaia e il titolo principale si sposti un po' + sotto

Questo lo facciamo grazie agli **eventi sui plugin jQuery**

Il plugin sticky ha degli **eventi già preimpostati** che ci permettono, vengono riportati nella documentazione nella sezione [Events](#)

ci permettono questi eventi preimpostati di eseguire delle funzioni in risposta ad un evento

- sticky-start è l'evento che si scatena quando iniziamo a scorrere giù

Es:

```
$('#sticker').on('sticky-start', function() { console.log("Started"); });
```

lo copio nel mio script.js

l'on è simile al click o meglio, è una struttura diversa per invocare eventi, ma fa la stessa cosa

nel selettore metto quello identificativo dell'elemento che deve rimanere sticky ovvero il **.main-header**

Nel corpo dell'handler vogliamo

- far sparire h2

- aggiungere del margin all'h1

```
$('.main-header').on('sticky-start', function() {
```

```
    // nascondo h2
```

```
    $('.main-header h2').hide();
```

```
    // aggiungo margin top
```

```
    $('.main-header h1').css('margin-top', '15px');
```

```
});
```


Quando però torno in posizione non torna tutto al suo posto, ovvero riappare h2 e si toglie il margine

quindi devo creare un handler per un altro evento ovvero

lo **sticky-end**

```
$('.main-header').on('sticky-end', function() {  
    // mostro h2  
    $('.main-header h2').show();  
    // rimetto il margin-top ad 1px  
    $('.main-header h1').css('margin-top', '1px');  
});
```

TRASCINARE UN ELEMENTO CON HEADER FISSO QUANDO VIENE INCONTRATO

Es: Check My Work vogliamo venga trascinato con l'header e magari anche cambiare la scritta

1. Rendiamo sticky l'epe che vogliamo

```
$('.banner p').sticky();
```

questo viene coperto da header, uso le [pty o opzioni](#) del plugin esempio

topSpacing distanza fra bordo alto e l'ele

quindi lo setto come la distanza dell'header, sua altezza, + una decina di pixel (**non usare px**)

```
$('.banner p').sticky({  
    topSpacing: 65  
});
```

non si vede quindi dobbiamo dare un colore diverso al div e ridurre lo spazio

```
$('.banner p').on('sticky-start', function() {  
    // aggiungo margin top  
    $('.banner p').css('color', '#000');  
});
```

```
$('.banner p').on('sticky-end', function() {  
    // aggiungo margin top  
    $('.banner p').css('color', '#fff');  
});
```

PLUGIN PER CREARE SLIDER CAROUSEL

Creiamo degli **slider-carousel** che sono degli elementi di **UI** e inserire delle immagini dentro la pagina e farle scorrere in vari modi, farle sfumare, usare delle frecce o dei punti per saltare e anche **renderlo automatico mettendoci un intervallo che li esegue da solo**

ci son mille modi per creare lo slider

Vediamo un plugin buono per questo che si chiama [slick](https://github.com/kenwheeler/slick)

<https://github.com/kenwheeler/slick>

vediamo esempi di come viene usato plugin

- singola immagine
- + foto
- diversi modi per farle scorrere

Su Usage

ti dice come usarlo

- come settare HTML
- cosa inserire come link dentro l'head
- i file js
- come inserire lo script nel file js
- ci da dei **setting ovvero delle opzioni per personalizzare lo slider**

Vediamo come inserirlo e come applicarlo

1. scarica Slick da “get it now” e “Download Now”

recupero i file nella cartella slick e la metto dentro cartella js

2. HTML ci dice di **creare un div con una classe con dentro altri div che contengano le varie slide**

```
<div class="your-class">  
  <div>your content</div>  
  <div>your content</div>  
  <div>your content</div>  
</div>
```

ad esempio noi dentro al div con your content metteremo

- un'immagine
- delle scritte
- un pulsante

-) inseriamo le classi d'esempio sotto al div con l'h2

```
<h1>jQuery</h1>
```

```
<div class="benvenuti">
```

```
  <h2>Impara ad applicare i plugin di jQuery!</h2>
```

```
</div>
```

```
<div class="carousel-slick">
```

```
  <div>your content</div>
```

```
  <div>your content</div>
```

```
  <div>your content</div>
```

```
</div>
```

```
</div>
```

-) inseriamo il CSS dentro l'head, prima di main.css, per poter modificare con il css locale quello del plugin

```
<link rel="stylesheet" type="text/css"
href="js/slick/slick.css"/>
```

```
<link rel="stylesheet" type="text/css" href="js/slick/slick-
theme.css"/>
```

mettiamo anche il theme.css che in pratica è lo stile di default

-) inseriamo il JS sotto prima del body e prima del nostro script.js

```
<script type="text/javascript" src="js/slick/slick.min.js">  
</script>
```

-) inserisco ora il codice JS per avere l'effetto Single Item

```
$('.carousel-slick').slick();
```

ovvero ti mostra il contenuto in singola slide e due bottoni previous e next per scorrere i contenuti

nei div al posto della stringa "your content" ci metto del contenuto ovvero le img con scritte e bottoni

```
<div class="carousel-slick">  
  <!-- slide1 -->  
  <div class="banner slide1">  
    <div class="corso">  
      <h3>  
        Segui il corso su Lacerba:  
      </h3>  
      <h2>  
        Logiche di bilancio  
      </h2>  
      <button>  
        Vai
```

```
</button>

</div>

</div>

<!-- slide2 -->

<div class="banner slide2">

  <div class="corso">

    <h3>

      Segui il corso su Lacerba:

    </h3>

    <h2>

      Ottimizza il tuo profilo LinkedIn

    </h2>

    <button>

      Vai

    </button>

  </div>

</div>

<!-- slide3 -->

<div class="banner slide3">

  <div class="corso">

    <h3>

      Segui il corso su Lacerba:

    </h3>
```

<h2>

HTML e CSS: Basi

</h2>

<button>

Vai

</button>

</div>

</div>

</div>

Dove l'immagine viene inserita direttamente nel css

ecco la definizione di slide1

```
.slide1 {  
    background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),  
    url(..img/bilancio.jpg) no-repeat center / cover;  
}
```

OPZIONI DEL PLUGIN

Personalizziamo e abbelliamo un po il carousel grazie alle opzioni del plugin e alle modifiche sul css

Vediamo opzioni plugin, su Settings:

- autoplay, per far andare slider automaticamente
- autoplaySpeed, impostare velocità dell'autoplay in ms
- arrows , sono i pulsanti prev e next, per toglierli metto false

- dots, per mettere dei punti per cambiare slide

ma i dots fan cagare

son un elenco, per risolvere il problema è andare ad usare il **tema default ovvero**

```
<link rel="stylesheet" type="text/css" href="js/slick/slick-theme.css"/>
```

con il js che è diventato

```
$('.carousel-slick').slick({
```

```
  // autoplay
```

```
  autoplay: true,
```

```
  // velocità
```

```
  autoplaySpeed: 2500,
```

```
  // tolgo prev e next
```

```
  arrows: false,
```

```
  // metto i punti
```

```
  dots: true
```

```
});
```

MODIFICHIAMO IL CSS DEGLI ELEMENTI DEL PLUGIN

per migliorare il design del carousel

Alcuni plugin **forniscono linee guida per modificare il CSS, questo nn ci da ele per modificare CSS**

vado nel main.css a modificare e sovrascrivere alcune cose del plugin (mai farlo su quello del plugin)

come capire cosa modificare???

Esempio voglio modificare i punti ad esempio li voglio ingrandire e portare nell'img

uso il DevTools per trovare la classe dei punti

.slick.dots

che ha -45px bottom

se clicco sul **::before** nel dom in devTool trovi tutti gli stili del punto, ad esempio possiamo aumentare dimensione e colore del punto

slick-active è la classe del li del punto attivo

Ad esempio spostato nell'img i puntini

```
.slick-dots {  
    bottom: 10px;  
}
```

* do colore bianco ai puntini, selettore recuperato grazie a Dev Tool*/

```
.slick-dots li button:before {  
    font-size: 15px;  
    color: white;  
}
```

```
/* do colore anche a quello attivo */
```

```
.slick-dots li.slick-active button:before {  
    color: white;  
}
```

metto l'opzione **fade true**

che da effetto svanimento al cambiare

```
    dots: true,  
    fade: true  
});
```

rimetto frecce togliendo arrows: true

e ne modifico il css per mostrarle e metterle dentro l'immagine e bianche

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

SLICK ARROWS

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

```
.slick-prev:before, .slick-next:before {  
    font-size: 30px;  
}
```

```
/* le sposto dentro l'immagine */
```

```
.slick-prev {  
    left: 25px;
```

```
z-index: 1000;  
}
```

```
.slick-next {  
  right: 25px;  
}
```

puoi anche provare a fare la stessa cosa con un'immagine di background che occupa tutta finestra, provare varie opzioni, cambiare tipo di slider