

Lezione 10

HTML and CSS: interazione utente

Interaction design è lo studio del comportamento degli elementi di una pagina web (e.g. link, bottoni, immagini) e l'analisi del comportamento dell'utente mentre interagisce con tali elementi.

CSS3 permette di applicare stili a elementi della pagina per modificare il **comportamento** che questi hanno durante l'interazione con l'utente.

In particolare, CSS3 offre **Pseudo-classi** e **Pseudo-elementi** per:

- interagire con elementi prima/durante/dopo un evento (e.g. click del mouse, hover su un link)
- intervenire su aspetti particolari di elementi che non possono essere strutturati in altri elementi HTML (e.g. la prima lettera di una parola, righe pari/diparsi di una tabella, etc.).

Interazione per l'orientamento dell'utente

La *pseudo-classe* più utilizzata nel web è `:hover`, è permette di modificare il comportamento di un elemento quando il mouse vi passa sopra (senza cliccare). Un esempio comune è il comportamento dei link.

`hover`

Un utente che passa il cursore su un elemento del menu di navigazione si aspetta un comportamento diverso rispetto al testo che lo circonda, tale da permettergli di capire che quello è un elemento interattivo.

Sui link è possibile intervenire in 4 momenti:

- `a:link` - un link mai visitato
- `a:visited` - un link già visitato dall'utente
- `a:hover` - un link quando il mouse vi passa sopra
- `a:active` - un link nel momento in cui viene cliccato

In un foglio di stile css, è fondamentale mantenere l'ordine di queste istruzioni.

Gli aspetti su cui intervenire nel comportamento di un link sono, generalmente i seguenti (uno o più contemporaneamente):

- colore del testo `~~~~ a:link {color: blue;} a:visited {color: violet;} ~~~~`
- stile del font `~~~~ a:hover {font-style: bold;} ~~~~`
- decorazione `~~~~ a:hover {text-decoration: underline;} ~~~~`
- background `~~~~ a:hover {background-color: white;} ~~~~`

NB. su device medio-piccoli (tablet, mobile), dove il mouse non viene utilizzato, i comportamenti relativi all'over del mouse scompaiono o vengono sostituiti con comportamenti diversi (e.g. il solo click). Pertanto, non si implementano mai comportamenti legati **solo** all'over, ma si studiano comportamenti che possono essere applicati anche al click.

In ultimo, per orientare l'utente è sempre bene **evidenziare** nel menu di navigazione la pagina in cui l'utente si trova in quel momento, applicando stili diversi all'elemento del menu corrispondente.

Ad esempio:

```
<!-- index.html -->
<nav>
  <ul>
    <li><a href="#" class="active">HOME</a></li>
    <li><a href="clients.html">OUR CLIENTS</a></li>
    <li><a href="about.html">ABOUT US</a></li>
  </ul>
</nav>
```

Applichiamo stili alla classe `.active` che abbiamo attribuito al link della pagina in cui ci troviamo.

```
a.active {
  background-color: white;
  color: blue;
}
```

Al cambiare della pagina cambieremo l'elemento a cui attribuiamo la classe `.active`.

```
<!-- about.html -->
<nav>
  <ul>
    <li><a href="index.html">HOME</a></li>
    <li><a href="#" class="active">OUR CLIENTS</a></li>
    <li><a href="about.html">ABOUT US</a></li>
  </ul>
</nav>
```

Un caso speciale: Breadcrumbs

In siti complessi (con molte pagine o diversi raggruppamenti/categorizzazioni delle stesse pagine, e.g. magazine online) la navigazione del sito necessita ulteriori mezzi per l'orientamento dell'utente. In tali casi, i menu di navigazione primaria vengono affiancati da ulteriori forme di navigazione dei contenuti, chiamate *briciole di pane*.

```
<!-- about.html -->
<nav>...</nav>
...
<section class="breadcrumb">
  About / <a href="#gallery">Gallery</a> / <a href="#pictures_of_us">pictures of us</a>
</section>
```

Che in un browser apparirà come segue:

About / [Gallery](#) / [pictures of us](#)

Sempre ai fini dell'orientamento dell'utente all'interno della pagina, è possibile applicare degli stili ad alcuni **punti specifici di un elemento** (e.g. di un paragrafo, di una parola, di una tabella) per migliorare la leggibilità e indirizzare l'occhio dell'utente. In questo caso si utilizzano gli *pseudo-elementi*.

Di seguito alcuni esempi utili:

```
p::first-line {
  font-style: bold; /* la prima linea di ogni paragrafo */
}

p::first-letter {
  font-size: 5em; /* la prima lettera di ogni paragrafo */
}

section.search::before {
  content: "search: "; /* aggiunge un testo all'interno dell'elemento section, prima del contenuto HTML */
}

section.intro::after {
  content: "read more... "; /* aggiunge un testo all'interno dell'elemento section, alla fine del contenuto HTML */
}

tr:nth-child(even) {
  background-color: #f2f2f2; /* applica un colore diverso a tutte le righe pari di una tabella */
}
```

NB. è possibile inserire anche delle **icone** utilizzando gli pseudo-elementi `::before` e `::after`. Le icone sono infatti oggetti vettoriali che possono essere identificati da codici.

```
section.search::before {
  content: "\f002" ; /* aggiunge un'icona con la lente d'ingrandimento */
}
```

Vedi [una lista di codici per le icone Fontawesome](#).

Esplorazione

Elementi interattivi possono agevolare non solo l'orientamento ma favorire l'esplorazione dei contenuti in maniera progressiva (e.g. *progressive disclosure*).

Per esempio, all'interno di una galleria di **immagini**, per evitare che l'utente sia visivamente disorientato dalle troppe immagini o dai colori intensi, si possono applicare **filtri**, opacizzando le immagini.

Ad esempio:

```
img {
  opacity: 0.5;
}

img:hover {
  opacity: 1.0;
}
```

Viceversa, è possibile opacizzare maggiormente un'immagine al passaggio del mouse e visualizzare una didascalia.

Vedi [alcuni esempi](#).

Altri casi di esplorazione dei contenuti sfruttano lo *scrolling* (scorrimento verso il basso/alto) della pagina.

N.B. i link non rimandano per forza ad un'altra pagina, ma possono rimandare a punti specifici della pagina web in cui ci si trova. Per fare ciò si utilizzano gli attributi `@id` per identificare il punto di arrivo (e.g. un titolo all'inizio di una sezione), e l'attributo `@href` per linkare uno o più punti di partenza al punto di arrivo.

Ad esempio:

```
<!-- about.html -->
<p>Vedi la <a href="#approfondimento">sezione successiva</a> per approfondimenti</p>

...

<section>
  <h2 id="approfondimento">Dicevamo...</h2>
  <p>...</p>
</section>
```

Transizioni e animazioni

Negli esempi visti, il passaggio da un livello di opacità ad un altro (nel caso delle immagini), il cambio di colore del background (nel caso dei link), o il cambio di colore del testo avviene repentinamente. Si possono effettuare delle **trasformazioni** (e.g. rotazioni), applicare effetti alle **transizioni** (stabilendo la durata e il tipo di effetto), andando a creare vere e proprie **animazioni**.

```
img {
  opacity: 0.5;
  transition: opacity .25s ease-in-out;
}

img:hover {
  opacity: 1.0;
}

p {
  transform: rotate(20deg); /* ruota un paragrafo di 20 gradi, non dipende da alcun evento */
}

section {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s; /* transizione sulla proprietà width di durata 2s */
}

section:hover {
  width: 300px; /* risultato finale della transizione, che avviene solo in caso di mouse over */
}
```

Vedi [un esempio](#)

Nel caso dello **scrolling** il movimento da un elemento ad un altro è repentino. Si può renderlo più elegante e *smooth* utilizzando una proprietà apposita

```
body {
  scroll-behavior: smooth;
}
```

Azioni

Oltre ai link, i **bottoni** mutano comportamento durante l'interazione con l'utente. Questi, rispetto ai semplici link che rimandano l'utente in un'altra pagina o punto della pagina, permettono agli utenti di compiere delle azioni (e.g. effettuare ricerche, visualizzare risultati).

```
span.button1 {
  background-color: white;
  color: black;
  border: 2px solid black;
  transition-duration: 0.4s;
  cursor: pointer;
}

span.button1:hover {
  background-color: black; /* il bottone cambia background e colore testo con una transizione di 0.4 secondi */
  color: white;
}
```

Vedi [alcuni esempi](#)

Altri esempi

Esistono altre tipologie di comportamenti (più sofisticate) che favoriscono l'interazione con l'utente, per esempio:

- dropdown
- tooltip
- sliders
- mappe interattive
- carousel di immagini
- autosuggest, as-you-type results
- paginazione dei risultati

Molte di queste tipologie richiedono istruzioni CSS più complesse o l'utilizzo di Javascript per modificare il comportamento di default degli elementi della pagina.

Nelle prossime lezioni vedremo:

- l'utilizzo di **librerie CSS** ([Bootstrap](#)) per ottenere effetti anche molto complessi utilizzando solo poche righe di codice
- l'utilizzo di un linguaggio di programmazione **Javascript** per modificare il comportamento degli elementi della pagina HTML

CSS: interazione utente

Riprendiamo il template grafico:

screenshot1 screenshot2

1. Apriamo in Sublime text il file HTML e CSS creati nella scorsa lezione
2. Modifichiamo il comportamento dei link nel menu (colore, background, hover, link pagina attiva)
3. Modifichiamo l'opacità dell'immagine e modifichiamo l'effetto di transizione
4. Creiamo dei link all'interno della stessa pagina e modifichiamo l'effetto dello scrolling
5. Inseriamo i bottoni e applichiamo degli stili per definirne il comportamento

Link utili

Per una lista completa di pseudo-classi e pseudo-elementi con esempi interattivi, vedi la pagina dedicata del [w3school](#)