

Introduzione

“Design” è un termine che negli ultimi anni ha preso sempre più piede nella quotidianità di ognuno di noi. È stato sdoganato probabilmente senza una volontà precisa, una sorta di evoluzione dei tempi che lo ha portato a uscire da ambiti specifici approdando, in pratica, in qualunque contesto. Oggi tutto è design, una guerra atavica che contrappone la sostanza alla forma in cui trova pacifica conciliazione questo termine. Design è molto di più della sua traduzione italiana “disegno”. Un progetto di un’automobile che racchiude in un’estetica accattivante tutte le funzionalità progettuali è, per definizione, un’opera di design.

Anni fa il termine “design” presupponeva una produzione in serie, come avveniva in ambiti industriali. Lo scenario attuale è più ampio rispetto al passato. Una qualsiasi azienda che produce apparecchi elettronici di fascia medio-alta ha certamente investito in *project design*. Basti pensare a Apple, che del design ha fatto la sua forza e la sua fortuna. Oggi questo termine racchiude ancor di più il senso di un’estetica ricercata, poiché il concetto di design rivive anche nelle piccole produzioni o nei singoli lavori e, perché no, anche all’interno di un browser.

NOTA

L’esempio per eccellenza di design italiano, storicamente riconoscibile, è la Fiat 500. In quest’auto forma e sostanza coesistono armonicamente in un preciso momento storico: questo ha fatto la differenza. La Fiat 500 è stata, ed è ancora, un’auto che difficilmente potrà essere dimenticata.

Fortunatamente per me e presumo anche per voi, l’obiettivo non è quello di raccontare di design, non ne sarei probabilmente in grado, ma di parlare di Web. Il design si è diffuso a macchia d’olio in svariati altri campi, approdando, come già accennato, anche nel World Wide Web. Trovare un applicativo web che dia un’idea di quanto accaduto nell’industria è davvero cosa assai difficile. Questo probabilmente perché all’interno del browser il design rimane un elemento creativo ma fortemente sottoposto agli standard che ne regolano il funzionamento. In questi anni l’esplosione di questo nuovo approccio alla progettazione è stato incalzante ma, rispetto a quanto accade in altri ambiti, nel mondo del Web si è proceduti con un passo meno spedito.

Un punto di inizio è stato sicuramente il sito internet di Audi (Figura I.1), caratterizzato da un innovativo progetto di design che gli consentiva di adattarsi alle dimensioni del dispositivo che lo stava navigando. Qualche anno dopo questo approccio sarebbe diventato standard *de facto*.

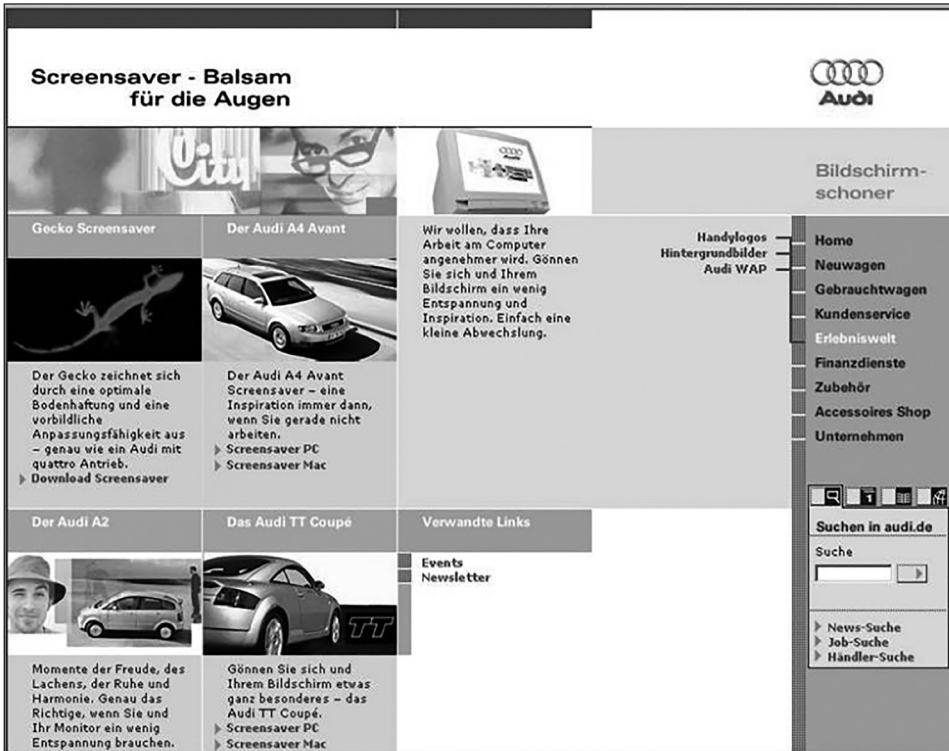


Figura I.1 Home page di Audi.de nell'anno 2002.

Siti internet, applicazioni web e desktop, applicazioni mobile e qualsiasi altra cosa che richieda un'interazione con l'utente necessita di una progettualità più complessa. Il design che vive all'interno delle applicazioni è in forte relazione alla *user experience*, altra espressione inglese che necessita di una breve spiegazione.

La user experience (UX), in italiano esperienza d'uso, non è altro che il processo che compie l'utente utilizzando uno specifico strumento, in questo caso un'applicazione web. La progettualità legata alla user experience è chiamata user experience design (UXD). Detto questo, l'esempio della Fiat 500, per quanto piuttosto azzardato, può accostarsi al tema appena introdotto. Come la Fiat 500 sposava alla perfezione l'esperienza di guida a una forma accattivante, creando così un design incredibilmente innovativo, l'esperienza utente scaturita dalla fruizione di un determinato applicativo deve essere "progettata" proprio con il fine ultimo di unire funzionalità d'uso a un'interfaccia grafica che ne faciliti l'utilizzo e allo stesso tempo lo renda gradevole e accattivante alla vista. Ancor di più, se lo user experience design è allo stesso tempo innovativo, il progetto a cui starete lavorando sarà certamente vincente.

A questo punto la traduzione letterale di “design” con “disegno” inizia ad apparire imprecisa. Per quanto letteralmente sia corretta, la traduzione più libera è “progetto”. Ecco dunque l’aforisma: il design non è un progetto ma piuttosto l’arte del progettare. Proseguendo la lettura di questo libro vi renderete conto di quanto il concetto di progetto si sposi perfettamente con quanto appena discusso.

A chi si rivolge questo libro

Un libro di questo tipo si rivolge a un target piuttosto ampio. Prenderemo in considerazione gli aspetti che ruotano intorno al web design, scorrendo ciò che si cela dietro l’utilizzo di un framework come Bootstrap.

NOTA

La versione di riferimento di questo volume è la 4. Lo studio degli argomenti introdotti in questo volume permette comunque di approcciare e comprendere dinamiche che riguardano anche le versioni precedenti.

Approfondiremo argomenti non necessariamente collegati al diretto utilizzo del framework, ma che sono fin troppo importanti al fine di fare un buon lavoro. L’obiettivo è essere tecnici senza rendere troppo pesante la lettura. Questo libro dovrebbe essere letto con la finalità di studiare come strutturare e realizzare applicazioni web e mobile, focalizzandosi particolarmente sulla fase di progettazione, di design!

Inizieremo a strutturare dei layout, dai più semplici ai più complessi (Figura I.2), analizzando e introducendo passo per passo i componenti web che implementano le interazioni con l’utente.

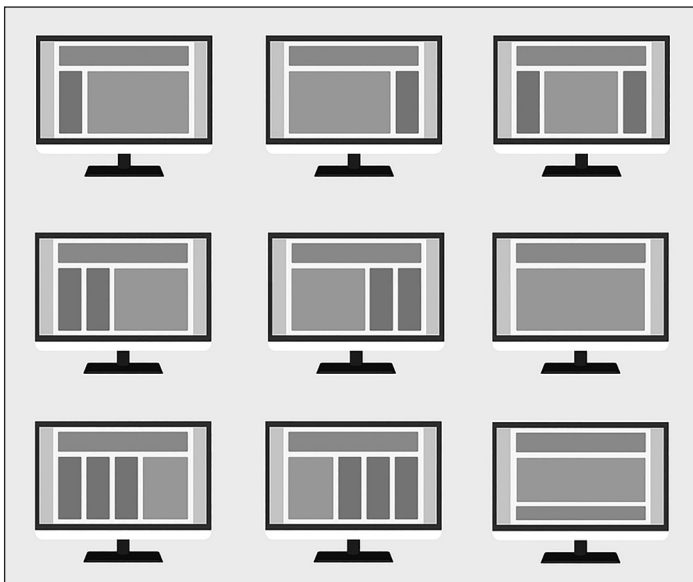


Figura I.2 Esempi di layout di siti web.

NOTA

Layout è un termine piuttosto usato nell'ambito del Web. La sua traduzione letterale è "disposizione", proprio perché con questo termine si identifica la disposizione degli elementi che strutturano la pagina.

Layout e design possono sembrare sinonimi, ma il layout deve essere immaginato come una piccola parte di un contesto molto più ampio, come appunto quello del web design. Un layout che manca di un'interazione con l'utente, per quanto possa essere complesso e strutturato, ha certamente una progettualità piuttosto ridotta e con tutta probabilità vivrà esclusivamente in piccoli ambiti legati al marketing (*landing page*) piuttosto che allo sviluppo di applicativi. Un layout che si sposa perfettamente con un'interfaccia con cui l'utente interagisce è invece un'opera di ingegno che, senza alcun dubbio, è corretto definire opera di design.

È doveroso procedere spediti con una breve introduzione ai concetti base che vivono all'interno del web design, affinché vengano comprese le differenti figure professionali che ruotano attorno a questo ambito. Una fetta importante del target a cui spero di fare riferimento è, senza alcun dubbio, chi si affaccia al web design con l'intento di farne il proprio lavoro. Questo è il motivo principale che mi spinge a essere più pragmatico che "accademico", nonostante, come già detto, non ci limiteremo solo a quello.

Una premessa

Le macro categorie che compongono il web design sono così ampie che è più importante specificare che cosa questo libro non tratta piuttosto che il contrario. Il web design appartiene alla categoria del *visual design*, che fa da cappello a tutto ciò che è comunicazione.

NOTA

Il visual design è una branca del design industriale e può essere pensato come la categoria madre della comunicazione visiva. Il web design ne è chiaramente parte costituente.

La figura professionale del web designer si occupa di approcciare a 360 gradi il design dell'applicativo, realizzando il suo layout e predisponendo le funzionalità legate all'interfaccia. Il web developer si occupa della realizzazione di applicativi, focalizzandosi sulla programmazione. La confusione potrebbe sorgere poiché entrambe le figure condividono determinate conoscenze. Faccio una considerazione banale. Il web designer non ha difficoltà nel creare un layout complesso per una landing page con le suite HTML5 e CSS3, mentre il web developer sarà certamente più a suo agio nello sviluppo di una mobile app in JavaScript.

NOTA

Una landing page, letteralmente "pagina di atterraggio", è una pagina web che viene mostrata all'utente dopo una precisa azione: un clic su un banner pubblicitario, dopo un pagamento, al termine di una registrazione ecc. Comunemente questa pagina è priva di interazione, ma si limita a mostrare un contenuto specifico con una forma accattivante.

A onor del vero, in ambiti aziendali le figure maggiormente ricercate sono sempre un mashup delle due e nonostante questo è molto importante che non vengano confuse!

Design vs development

Un web designer deve comprendere gli aspetti base che si celano dietro lo sviluppo di ciò che probabilmente ha progettato. Un web developer, invece, deve sempre essere in grado di creare un layout responsive. Per quanto banale, questo concetto racchiude la sostanziale differenza tra queste due figure. Etichettare con questa dovizia di particolari le figure professionali legate al Web è un gioco pericoloso. Questo perché attualmente è innegabile che ci sia una forte sovrapposizione di skill.

NOTA

Il responsive web design è una tecnica che consente di creare layout web che si adattano al dispositivo usato per la navigazione. La visualizzazione del sito web responsive cambierà in base al dispositivo utilizzato, desktop o mobile.

Approfondiamo ulteriormente il panorama delle figure professionali legate a questo mondo. Una figura che appartiene al web development è il *front-end developer*. Quest'ultimo si occupa di realizzare e creare applicativi web, focalizzandosi però maggiormente sulla programmazione che sul design.

La figura professionale del *back-end developer* completa, insieme al front-end, la categoria del web development. Il back-end developer si occupa maggiormente degli aspetti legati alla parte server, mentre il front-end developer si occupa principalmente di quella client.

NOTA

La categoria del back-end development comprende, di fatto, tutto ciò che viene posto dopo la "raccolta" dei dati effettuata dal client. Esistono ambienti intermedi, chiamati middleware, che nonostante abbiano anche in questo caso specifiche figure tecniche, vengono comunque identificati come ambienti back-end.

Il web designer e il front-end developer condividono le stesse conoscenze tecniche di base: HTML5, CSS3, JavaScript e tutto ciò che ne consegue. La differenza nasce proprio in quest'ultima skill che, se evoluta, sposta il peso della bilancia dal design allo sviluppo. Un web designer dovrebbe conoscere JavaScript, ma probabilmente le sue competenze saranno basilari. Un front-end developer è in grado di strutturare un applicativo web complesso con l'ausilio di design pattern e utilizzando a fondo la programmazione JavaScript.

Ok, ammetto che ho il dubbio di avervi confuso ulteriormente, ma sono convinto che la responsabilità sia intrinseca in questo ambito, che non ha mai conosciuto involuzioni ma solo una crescita esponenziale. Una delle conseguenze di questa crescita è proprio la costante nascita di nuove figure professionali in ambiti specifici che, in molti casi, vivono solo all'interno di una definizione.

Concordo con ciò che state pensando! Per questo motivo reputo che non sia corretto ragionare per definizioni, teorizzando ruoli e casistiche che non hanno nessuna

parvenza di realtà. Quello che faremo sarà dare un taglio il più pragmatico possibile a ciò che stiamo trattando. Una considerazione del tutto personale è che questa sovrapposizione di ruoli sarà sempre più labile, poiché le skill tecniche saranno sempre meno necessarie con l'aumentare di software, librerie, framework ecc. L'attenzione si è spostata dal comprendere il linguaggio al prendere dimestichezza con uno strumento. Il fine sarà lo stesso in entrambi i casi, ma lo strumento faciliterà di molto il lavoro. Un esempio concreto è Bootstrap, che consente, tramite l'utilizzo di griglie, di creare un layout responsive senza doversi preoccupare di scrivere righe e righe di CSS. È indubbio che sia più semplice studiare Bootstrap piuttosto che ciò che si cela dietro le quinte del browser. Questo non deve però limitare lo studio del web designer, poiché gli strumenti che facilitano il lavoro verranno usati molto meglio se si comprende effettivamente ciò che lo strumento fa, piuttosto che scorgerlo appena. Per quanto sia vero che in ambito lavorativo, specialmente in quello informatico, una figura polivalente risulta essere vincente, è ancora più vero che avere un ambito specifico e una conoscenza approfondita lo è ancora di più.

Per corroborare quanto appena detto, nonostante l'argomento trattato sia il design e non la scrittura del codice, affronteremo comunque argomenti legati al mondo dello sviluppo, come appunto il linguaggio JavaScript.

Gli strumenti del mestiere

Gli strumenti principi del web designer sono HTML5 e CSS3. Io credo che il web designer dovrebbe approcciare le basi di JavaScript come farebbe un web developer. Lo stesso dovrebbe valere anche per il graphic designer. Potremmo, idealmente, porre il web design a metà strada tra il web development e il graphic design.

Per deformazione professionale approfondirò di più l'ambito legato allo sviluppo rispetto a quello prettamente grafico. Di norma, la realizzazione di un layout grafico spetta al web designer, ma una bozza progettuale è di solito realizzata dal graphic designer, altra figura professionale che vive all'interno di un *project design*. Il web designer, in progetti di piccole o medie dimensioni, si occupa spesso anche dell'aspetto progettuale. In progetti complessi dove vengono coinvolte più figure professionali, il graphic designer realizzerà un mockup e il web designer si baserà su quello per realizzare un layout definitivo.

NOTA

Il mockup è una rappresentazione sotto forma di bozza, tipicamente realizzata dal graphic designer, che dà una chiara idea progettuale di ciò che si vuole realizzare.

A breve analizzeremo nello specifico quali sono i processi, le aree e le definizioni che permettono di comprendere meglio un argomento così ampio come quello del web design. Gli aspetti teorici lasceranno spazio a quelli pratici che senza alcun dubbio chiariranno quanto sia effettivamente più semplice realizzare un sito web con una buona user experience piuttosto che addentrarsi nei meandri di definizioni di termini che sono molto simili, ma che, ahivoi, dovrete comunque imparare. A oggi gli scenari lavorativi legati al mondo del Web, tra sviluppo e design, aprono una categoria molto ampia di posizioni tecniche. Procederemo sì focalizzati sul design, ma per approfondire al meglio questa tematica, sarà altresì importante comprendere le nozioni di base legate

allo sviluppo web. Ogni tecnico che lavora nel Web, per quanto sappia muoversi bene tra design e sviluppo, ha comunque fatto la sua scelta: ha deciso se essere sviluppatore o designer, e alla fine di questo libro potrebbe toccare anche a voi.

Scenario

Per comprendere ciò che andremo a trattare è bene innanzitutto chiarire da un punto di vista teorico ciò che ruota intorno al mondo del web design. Esso racchiude al suo interno due macro categorie: user interface (UI) e user experience (UX).

La differenza tra questi due ambiti è che un UI designer progetta e realizza a 360 gradi quella che è l'interfaccia del progetto, nella fattispecie quindi ciò che i nostri occhi percepiscono guardando una pagina web, un applicativo web, un'applicazione mobile. Un UI designer rende invece fruibili le informazioni che stanno alla base di qualsiasi progetto di comunicazione e si occupa di realizzare quanto progettato dall'UX designer.

L'UX design è un lavoro che rientra nell'ambito della progettualità piuttosto che in quello pratico. Lo user experience designer approccia il progetto con la finalità di rendere migliore l'esperienza d'uso del cliente. L'interazione tra l'utente e l'azienda che vive all'interno dell'applicativo è il focus principale della progettazione effettuata dall'UX designer. Semplificando ulteriormente, pensiamo all'UX designer come alla mente e all'UI designer come al braccio.

Argomenti tecnici

I modelli utilizzati per approcciare scenari di design più o meno complessi vengono definiti *layout pattern*. I layout pattern più comuni verranno mostrati e realizzati tecnicamente, passo dopo passo. Ma prima introdurremo i concetti base di HTML5 e CSS3, che stanno alla base del framework che utilizzeremo. Dopo i capitoli iniziali, più teorici, approderemo al cuore pulsante di questo volume: Bootstrap. I concetti di pattern e framework non sono legati esclusivamente allo sviluppo web, ma si sposano anche con il web design.

Bootstrap è il web framework (WF) per eccellenza. Il termine "framework" ha diverse definizioni, ma senza alcun dubbio si riferisce a uno strumento atto a semplificare la creazione di una web application o di una banale landing page. Le capacità tecniche legate alla conoscenza del funzionamento del browser vengono così dirottate verso la conoscenza specifica del framework. Quello che dovrebbe fare un buon framework è astrarre: "preoccupati di studiarvi che al resto penso io" direbbe se potesse parlare. È innegabile che sia molto più semplice studiare un framework piuttosto che comprendere nel dettaglio il lato nascosto di una pagina web.

Il passo tecnologico successivo è stato imposto dalla naturale espansione del Web e dalle conseguenti necessità di incrementare le possibilità di interazione con l'utente. L'obiettivo principale è sempre quello di migliorare costantemente ciò che viene definito "user experience", l'esperienza d'uso dell'utente. Le applicazioni sono diventate più complesse e ricordano sempre di più vere e proprie applicazioni desktop. Lo sviluppo del Web in questi anni ha fatto sì che anche la struttura di base si evollesse, e la conseguenza è proprio la nascita di strumenti che consentono con estrema facilità la

creazione e la manutenzione di applicativi che senza l'uso di framework sarebbero molto più complessi da realizzare e ancor di più da mantenere.

Bootstrap

Bootstrap è senza possibilità di smentita il primo framework front-end ad aver catturato l'attenzione dell'intera comunità legata al web design. A onor del vero la sua definizione come “framework” non è del tutto condivisa. È sì un framework, ma i suoi creatori lo definiscono *toolkit*: una libreria che al suo interno contiene una serie di componenti che permettono la creazione di applicativi *responsive* e *mobile first*. La tecnica mobile first vede un approccio orientato fin da subito allo sviluppo di un applicativo mobile, che si adatta a dispositivi desktop, non il contrario. Bootstrap supporta nativamente il design responsive, facendo sì che un'applicazione risulti fruibile da qualsiasi piattaforma e dispositivo.

Il detto “anche l'occhio vuole la sua parte” si sposa perfettamente con l'uso di Bootstrap, perché spesso e volentieri la user interface risulta essere più apprezzata dell'usabilità dell'applicativo stesso. Il progetto Bootstrap, in questo campo, è il principale e incontrastato protagonista. Nasce nel 2011 in casa Twitter, con lo scopo di sviluppare una libreria interna che gestisse interamente e in maniera uniforme tutto il layout dell'intero progetto Twitter. Gli sviluppatori si resero conto dopo poco tempo che il progetto aveva preso una piega piuttosto importante e decisero di condividerlo con la comunità rilasciandolo opensource. Bootstrap a oggi rappresenta la risorsa principale nel disegnare lo stile di un sito, contiene un'enorme quantità di risorse HTML5 e CSS3 che possono essere integrate in pagine web con estrema facilità, rendendo quel dato componente, che sia un bottone, un form o altro, esteticamente accattivante e uniforme in tutti i browser. Una delle caratteristiche di questo front-end framework è proprio il responsive design. Bootstrap aiuta lo sviluppatore a realizzare applicativi che supportino automaticamente dispositivi di risoluzioni differenti. Utilizzando specifiche classi CSS3 fornite da Bootstrap, non ci si dovrà preoccupare di quale dispositivo l'utente utilizzi. Il sito ufficiale di Bootstrap è <http://getbootstrap.com/> (Figura I.3).

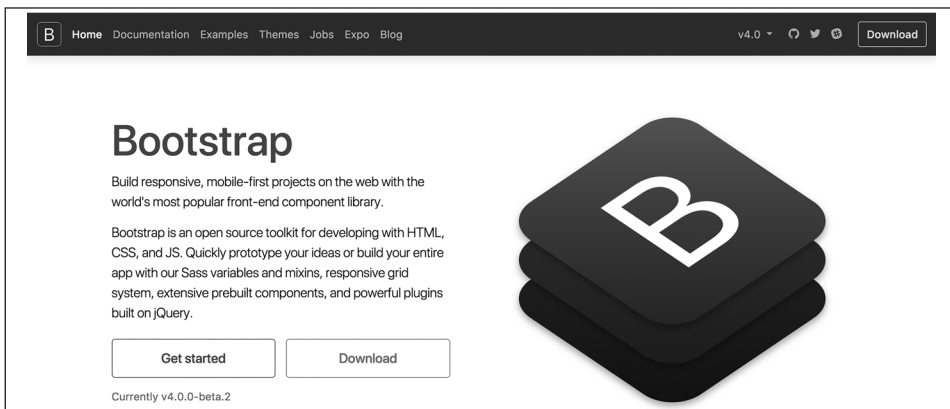


Figura I.3 Homepage del progetto Bootstrap.

I contenuti

Prima di buttarci a capofitto nella parte più pratica, i primi capitoli cercheranno di aiutarvi il più possibile nell'apprendimento di tutto ciò che è propedeutico all'uso di Bootstrap.

Le basi necessarie alla realizzazione di un'applicazione web o mobile sono la conoscenza di HTML5, CSS3 e JavaScript. Per quanto il web design non rientri propriamente nella sfera dello sviluppo, creare una UI accattivante richiede l'uso di JavaScript, per esempio i componenti Bootstrap che vedremo successivamente. Appunto per questi motivi, affronteremo nella fase iniziale lo studio della suite HTML5 e CSS3.

- Capitolo 1, *Le basi del Web: HTML5 e CSS3* – Iniziamo comprendendo il funzionamento del browser e introducendo al lettore l'accoppiata HTML5 e CSS3, fondamentali per la costruzione e lo styling di una pagina web. Vengono forniti numerosi esempi utilizzando i tag più comuni.
- Capitolo 2, *L'evoluzione del CSS: Sass* – Proseguiamo illustrando Sass, un'estensione del linguaggio CSS, fondamentale per una migliore creazione di stili.
- Capitolo 3, *JavaScript* – Linguaggio di riferimento del Web, JavaScript è indispensabile per la creazione di siti web accattivanti. Muoviamo i primi passi nel suo uso, propedeutico alla comprensione di jQuery.
- Capitolo 4, *Alla base del progetto: il wireframe* – Una volta messe in luce le basi del linguaggio, non si può realizzare una pagina web se non si effettua un'analisi di come questa debba apparire. Realizziamo degli esempi base.
- Capitolo 5, *Responsive web design* – Il responsive web design è fondamentale per la creazione di applicazioni web che siano fruibili attraverso tutte le tipologie di device. Analizziamo i principali pattern responsive.
- Capitolo 6, *Primi passi con Bootstrap* – Nato dal team di Twitter come libreria a uso interno, Bootstrap è diventato in pochi anni il framework front-end per eccellenza. Analizziamo le caratteristiche della versione 4 e creiamo i primi template.
- Capitolo 7, *Grid design* – Passiamo poi a comprendere il funzionamento delle griglie, indispensabile per utilizzare Bootstrap.
- Capitolo 8, *Elementi di layout* – Ora approfondiamo l'uso di Bootstrap. Creiamo alcuni esempi con gli elementi fondamentali che determinano la struttura di una pagina web.
- Capitolo 9, *Tipografia e immagini* – Introduciamo i componenti per gestire i principali contenuti di una pagina e i concetti base dietro all'applicazione dei relativi stili.
- Capitolo 10, *Componenti statici* – Scopriamo i componenti che consentono di creare le più comuni interfacce e rendono il design più accattivante.
- Capitolo 11, *Componenti dinamici* – Esaminiamo quindi i componenti offerti dal framework per le più frequenti interazioni di user experience e applichiamo ogni componente Bootstrap a una pagina web.
- Capitolo 12, *Utility* – Infine prendiamo in esame le utility, una serie di classi CSS che facilitano di molto le operazioni più comuni.
- Capitolo 13, *Consigli conclusivi* – L'ultimo capitolo, alla luce di quanto detto e fatto, ci lascia con preziosi consigli su come proseguire nel percorso che è stato tracciato.

Riepilogo

La struttura del libro, come introdotto dal paragrafo precedente, segue le necessità del lettore aiutandolo a comprendere quanto è necessario per poter utilizzare Bootstrap, progettare e realizzare una vera applicazione.

Innanzitutto capiremo il funzionamento del browser e studieremo HTML5 e CSS3 focalizzandoci sul box model. Dopo aver preso dimestichezza con CSS3 saremo in grado di evolvere quanto studiato e di migliorare la preparazione dei nostri fogli di stile integrando un'estensione CSS, di fondamentale importanza, chiamata Sass. Fatta la conoscenza dei compagni di lavoro HTML5 e CSS3, è importante approfondire i fondamentali di JavaScript. Un web designer che si muove bene anche con jQuery e EcmaScript 6 è sicuramente un valore aggiunto rispetto a chi si limita al markup e allo styling. Ricordiamo nuovamente che JavaScript è comunque propedeutico all'uso dei componenti dinamici di Bootstrap.

Dopo aver terminato lo studio di ciò che consente di poter affrontare in tutta sicurezza le principali attività legate al web design procediamo con la realizzazione di diversi wireframe. Il passo successivo e ultimo prima di lanciarsi a capofitto in Bootstrap sarà analizzare l'approccio più diffuso al web design, definito come responsive web design (RWD), che è davvero importante comprendere a fondo. I capitoli successivi saranno incentrati su Bootstrap e piuttosto tecnici, correlati da numerosi esempi. Nella fattispecie analizzeremo le griglie, che sono alla base del framework, comprendendone il funzionamento e l'utilizzo negli scenari più comuni.

La dimestichezza con le griglie ci consentirà di affrontare e capire meglio gli elementi di layout offerti da Bootstrap. Il Capitolo 9 si focalizzerà poi sui diversi stili di visualizzazione, che sono soltanto la "normalizzazione" degli elementi più comuni. Per esempio le immagini, che grazie all'ausilio di classi adatte allo scopo subiranno migliorie legate alla visualizzazione.

Il cuore del volume sono i due capitoli successivi, il 10 e l'11. Questi trattano nello specifico i componenti CSS3 e quelli dinamici di Bootstrap che lavorano con l'ausilio di JavaScript. I numerosi esempi e i template di prova che verranno realizzati completeranno la conoscenza del framework. Il lettore a questo punto sarà in grado di disegnare i primi layout e procedere all'interazione con il browser attraverso l'uso di piccoli script JavaScript.

Il Capitolo 12 introdurrà aspetti non necessariamente secondari, come le utility, e spiegherà al lettore come scrivere un codice che sia il meno ridondante possibile. Nel capitolo conclusivo, quando ormai potremo considerarci amici, daremo spazio ai sentimenti. Ci saluteremo con qualche consiglio e un grosso abbraccio.

Codice degli esempi

A questo libro sono affiancate alcune risorse reperibili online.

Tutto il codice degli esempi è liberamente disponibile agli indirizzi:

- <http://www.angular-italia.it/src/bootstrapbook.zip>
- <http://bit.ly/apo-wdb>

Buona lettura!