Appunti di Web Design

Giacomo Zanatta

April 16, 2018

Contents

1 Introduzione

1.1 Evoluzione dei linguaggi

- Layout solo HTML
- Layout ibridi (tabelle + regole css per la presentazione)
- Layout CSS (div, span + css evoluto per caratteristiche posizionali di presentazione)
- Layout CSS + media queries (responsive)

Separare contenuto e presentazione significa poter cambiare la visualizzazione del sito senza doverne cambiare il contenuto

1.2 Siti Responsive

- Permettono una presentazione flessibile, che si adatta alla tipologia di dispositivo utilizzato.
- Senza un sito responsive, possono sorgersi problemi se l'accesso ad un sito avviene attraverso l'uso di una rete sociale (esempio, se metto un link di un sito su facebook, se il sito non è responsive posso aver problemi a visualizzarlo correttamente se sono da mobile)
- Per ottenere un sito responsive posso utilizzare diverse tecniche: posso usare un framework, un CMS, oppure usare direttamente HTML, CSS e JavaScript. Inoltre posso usare tecniche di riconoscimento del browser per indirizzare l'utente verso la versione del sito adatta per il suo dispositivo o fornire una versione unica per i contenuti e forme di presentazione differenziate, opportunamente selezionate dai browser.

2 Graphics VS Web

2.1 World Wide Web

Il web nasce nel 1993, per permettere di condividere rapidamente in modo centralizzato i risultati scientifici di gruppi di lavoro. Il web permette l'ipertestualità e l'ipermedialità. Proprio su queste caratteristiche sono nati applicativi commerciali (come Hypercard e Toolbox) e sono stati fatti studi teorici (Nelson e Engelbart).

2.1.1 Caratteristiche di successo

- Paradigma intuitivo per l'utente
- Condivisione dell'informazione

- Multipiattaforma
- Linguaggio di facile utilizzo e basato su marcatori (tag)

2.1.2 Evoluzione del Web

Inizialmente era una rete locale, poi è diventata globale.
Ora nel web possono accederci non solo utenti che lavorano in ambito scientifico ma qualsiasi persona, da qualsiasi parte del mondo.
Il web inoltre presenta diversi ambiti applicativi: può essere utilizzato come frontend di sistemi informativi oppure come contenitore di contenuti.

2.1.3 Epoche del web

1. 1993: Far Web
2. 1994-1995: Monolito
3. 1996 - ...: Approccio interdisciplinare

2.2 Desing pre-web (editoria)

- Output cartaceo fisso, con metodologie non informatizzate (stampe d'autore...)
- Output cartaceo fisso, con metodologie informatizzate (giornali, riviste, libri...)
- Output elettronico fisso, con metodologie informatizzate (cd-rom, pubblicazioni elettroniche...)

2.2.1 Dogmi del graphic design

- Carattere tipografico inalterabile (font PostScript)
- Colore inalterabile
- Inalterabilità della composizione visuale

2.3 Progettare per l'ignoto

Per progettare per il web è necessario tenere conto di molti fattori:

- Versione del browser
- Piattaforme HW
- Preferenze utente
- Velocità connessione
- Caratteri tipografici

- Colori
- Dimensione della finestra di visualizzazione e layout

Per sopravvivere all'ignoto è necessario abbandonare la pretesa del controllo assoluto. Disegniamo quindi strutture e imponiamo set di regole, ma non dobbiamo raggiungere il controllo assoluto al singolo pixel.

2.4 Il carattere digitale

La rappresentazione digitale del carattere è soggetta a molte limitazioni.
Viene rappresentato attraverso un insieme di pixel che fanno riferimento ad una griglia di base. Nei sistemi operativi più vecchi, i caratteri tipografici vengono rappresentati come mappe di punti (font bitmap) pre-disegnate a dimensioni specifiche.
Nei sistemi moderni vengono utilizzati font outline, ossia rappresentati come primitive matematiche.

2.4.1 Rasterizzazione

Il processo di rasterizzazione di un font consiste nel convertire il testo da una descrizione vettoriale (font outline) ad una descrizione bitmap o raster.
Spesso si usano tecniche di antialiasing sul testo che deve essere letto sullo schermo, per renderlo più gradevole e leggibile.
È possibile usare anche l'hinting, che rende il font più gradevole e leggibile per una particolare dimensione usando informazioni pre-calcolate. Recentemente viene usato il subpixel rendering, nel quale si utilizzano le tre componenti RGB per aumentare la risoluzione dell'immagine.

2.4.2 Unità di misura del carattere

Il carattere tipografico può essere specificato attraverso diversi sistemi.
In ogni caso la presentazione del carattere su schermo passa attraverso il processo di rasterizzazione, che si basa sulla specifica dell'unità di misura, ma anche su alcune assunnzioni. Le unità di misura utilizzate sono:

- Pixel (px): unità minima di colore visualizzabile su schermo.
- Point (pt): misura tipografica tradizionale (72 punti per pollice).
- Pica (pc): 1 pica sono 12 punti (1/6 di pollice).
- Em (em): unità relativa che corrsiponde alla larghezza della lettera 'M' nel carattere utilizzato.
- Ex (ex): si basa sull'altezza della lettera 'x' nel carattere utilizzato (circa metà em)

- Inches (in): unità di misura standard negli USA.
- Millimetri (mm)
- Centimetri (cm)

2.4.3 Se l'autore non specifica il font

In assenza di specifiche, la selezione del tipo e della dimensione dei font della pagina è a carico del browser. In questo caso, possono sorgere risultati diversi a seconda del browser o del S.O. utilizzato. Negli ultimi anni, comunque, si sta cercando di attuare un processo di standardizzazione.
Vengono utilizzati due font:

1. Font proporzionale: viene allocata una diversa quantità di spazio orizzontale per ogni carattere, sono più facili da leggere (Times, Helvetica, Arial).
2. Font a larghezza fissa: viene allocata la stessa quantità di spazio orizzontale per tutti i caratteri. È utile per incolonnare i caratteri, ad esempio per mostrare sulla pagina web frammenti di codice (Courier, Monaco).

Ovviamente (per quasi tutti i browser) è possibile cambiare il font di default dalle impostazioni.

2.4.4 Testo come immagine

È possibile codificare il testo come una immagine. Questo porta dei vantaggi: abbiamo il controllo assoluto (sappiamo che la visualizzazione sarà esattamente come la vogliamo, su qualsiasi browser) ma anche dei svantaggi (le immagini sono lente a caricarsi sul browser, su browser non grafici l'immagine non sarà visualizzata e il testo non può essere indicizzato).

2.4.5 Il futuro del font

Da un po' di tempo sono disponibili i web fonts, ossia font direttamente accessibili da un server online che ne permette l'utilizzo e il download.
In questo modo abbiamo la possibilità di scaricare un'ampia famiglia di font vettoriali attraverso un link da una pagina web. Al font possiamo applicare effetti, come ad esempio l'ombreggiatura.
Google fornisce font gratuiti, ma esistono online alcuni store che permettono di scaricare font a pagamento e su licenza.
Un problema da tenere in considerazione sono le perfomance: è necessario tenere conto della velocità di caricamento del font e inoltre è necessario implementare una soluzione di backup per i browser che non riescono a visualizzare questi font.

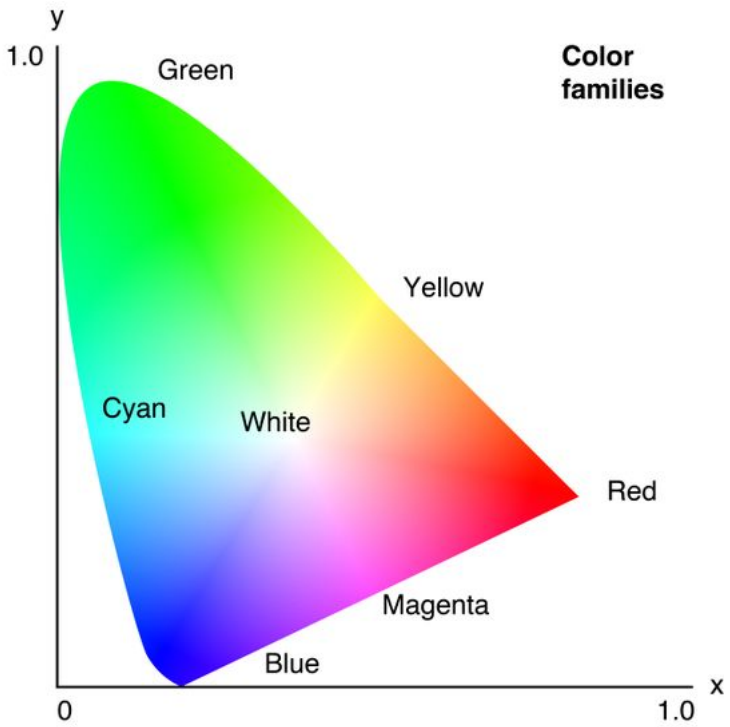
2.5 Colore

Il colore inoltre è in funzione di: risorse HW di sistema, sistema operativoe browser (il quale possiede risorse autonome per effettuare il rendering del colore su sistemi con limitate capacità grafiche.

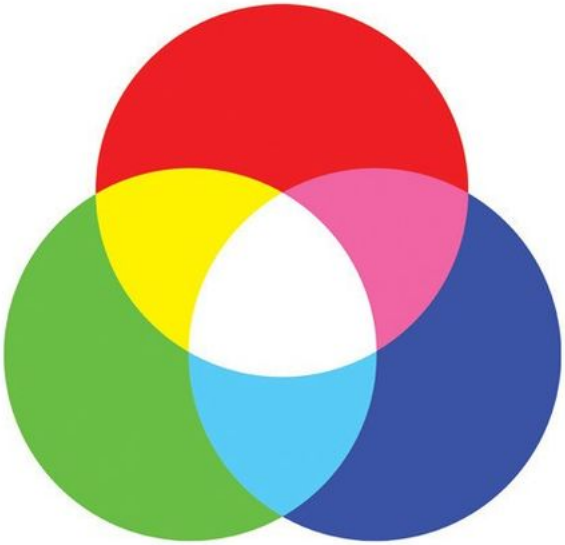
2.5.1 Modelli di colore

Un modello di colore è un modello matematico astratto che permette di rappresentare i colori in forma numerica. Esistono diversi modelli di colore, tra cui:

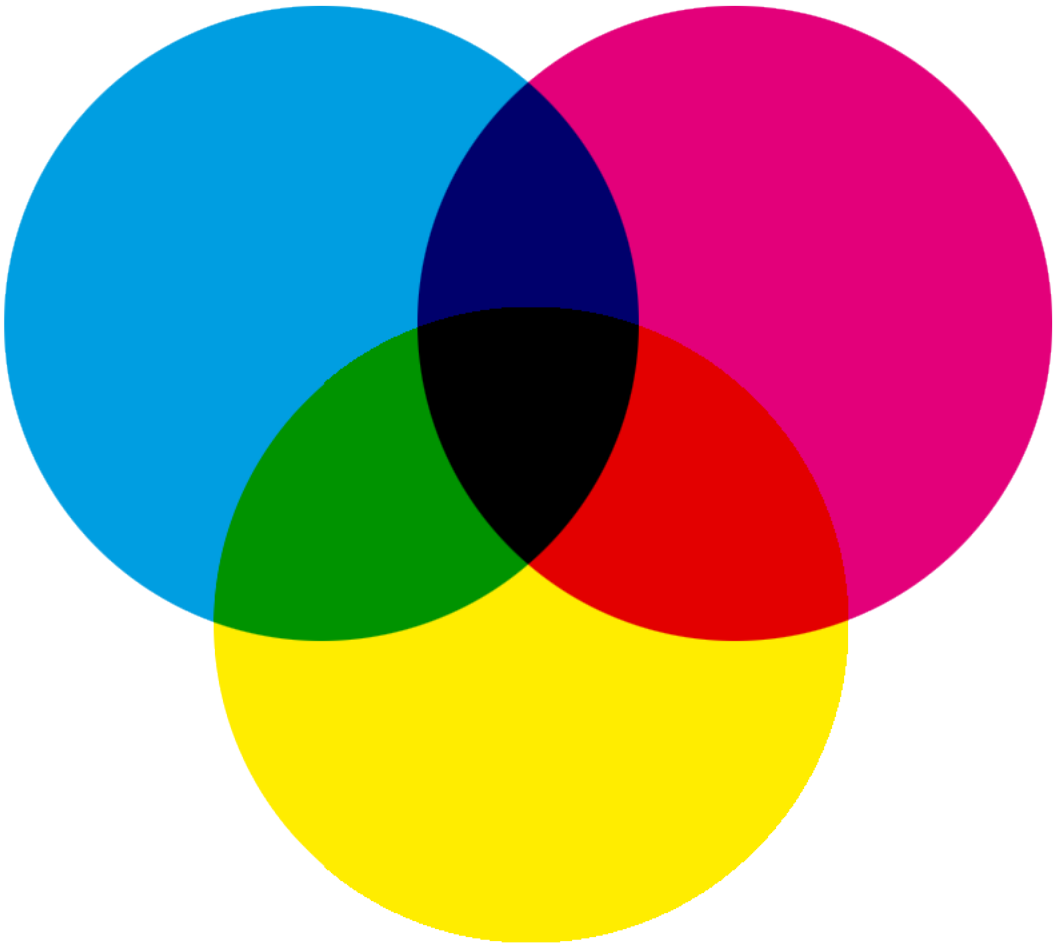
1. **CIE Yxy**: è un sistema che simula bene il processo visivo, e per definire un colore viene utilizzato un triangolo.
   Questo triangolo descrive lo spazio colore tramite due variabili cromatiche x e y.
   Sul piano cartesiano è situtata una curva a ferro di cavallo, sul cui bordo sono situati i colori puri identificati dalle lunghezze d'onda. Più ci si sposta verso il centro del grafico, più la saturazione si riduce e il colore diventa sempre più neutro.
   Il valore x indica l'importanza della componente rossa del colore nei confronti delle componenti verde e blu, ed è inferiore a 1. Il valore y invece indica l'importanza della componente verde nei confronti delle componenti rossa e blu, anche questa inferiore a 1.
   Y (maiuscolo) indica la lumosità (compresa tra 0 e 100).


y
1.0
Green
Color families
Yellow
Cyan
White
Red
Magenta
Blue
x
0
1.0

2. **RGB**: una vasta percentuale dello spettro visibile può essere rappresentata miscelando i 3 componenti della luce colorate Rosso, Verde e Blu in diverse proporzioni e intensità.
   Vengono chiamati colori addittivi perchè se sommiamo questi 3 colori generiamo il colore bianco. I colori addittivi vengono usati per l'illuminazione, i video e i monitor.



3. **CMYK**: questo modello si basa sulla capacità di assorbimento della luce dell'inchiostro sulla carta. Quando la luce bianca colpisce gli inchiostri traslucidi, una parte dello spettro viene assorbita e una parte viene riflessa all'occhio.
   I pigmenti puri di Cyan, Magenta e Giallo si sommano per assorbire tutto il colore e produrre il Nero: per questo sono chiamati colori sottrattivi.
   La combinazione di questi inchiostri per riprodurre il colore viene chiamata stampa in quadricomia.



2.5.2 Spazio colore

Lo spazio colore indica l'intervallo dei colori che possono essere visualizzati o stampati.
Lo spettro dei colori percepiti dall'occhio umano è maggiore di qualsiasi metodo di riproduzione del colore.

Fra i vari modelli del colore, CIE ha lo spazio colore più ampio, e comprende gli spazi di colore di RGB E CMYK.
Ci sono diversi spazi colore RGB: ad esempio, sRGB (disegnato in relazione alle possibilità di visualizzazione dei monitor CRT, gamma molto ristretta), Adobe RGB (che include la maggior parte dei colori ottenibili sulle stampanti CMYK ma utilizzando colori primari RGB, gamma molto più ampia di sRGB), Adobe WWide Gamut RGB (versione estesa di Adobe RGB).
È necessario tenere in considerazione che questi sono modelli teorici rispetto ai quali i singoli dispositivi riescono ad adeguarsi in modo diverso.

2.5.3 Pixel

Per controllare il colore di ogni pixel sullo schermo il sistema operativo deve dedicare una piccola quantità di memoria ad ognuno di essi.
La memoria dedicata allo schermo è una memoria distinta, che risiede nella scheda grafica e ci si riferisce con Video RAM (VRAM).

2.5.4 Colore RGB digitale

Per ogni pixel, abbiamo:

- HIGH COLOR: 16 bit, 5 bit per componente (il sedicesimo bit utilizzato per altri scopi).
- TRUE COLOR: 24 bit, 8 bit per ogni componente. Se si utilizzano 32 bit per pixel, i bit aggiuntivi vengono riservati per velocizzare le operazioni della scheda grafica o per informazioni di mascheratura/trasparenza.
- DEEP COLOR: 16/32/48/64 bit per componente.

Colore RGB a 24 bit
Con schede grafiche RGB a 24 bit, viene assegnato un valore di intensità a ogni pixel compreso tra 0 (nero) a 255 (bianco) per ognuna delle componenti RGB di un'immagine a colori. Le immagini RGB utilizzano 3 colori per riprodurre fino a 16,7 milioni di colori sullo shcermo.

2.5.5 Discretizzare il reale

La realtà percepita dall'occhio umano è a tono continuo. Catturare un'immagine con strumenti digitali vuol dire discrettizare l'informazione contenuta nella scena reale.

2.5.6 Colori per il Web

Per specificare i colori RGB è quello di utilizzare il valore numerico della tripletta di componenti, convertiti in notazione esadecimale.

Su sistema hardware dotati capacità grafiche inferiori (8-16 bit) i colori provenienti da uno spazio colore true space (ossia a 24 bit) vengono approssimati.
Per operazioni a livello di sistema i computer utilizzano un set specifico di 256 colori, chiamato palette di sistema.
Ogni sistema operativo ha la sua palette. Per questo motivo è stata introdotta una web-safe palette, che consiste di 216 colori comuni alle palette di sistema Win e Mac.
Tutti i colori della web palette sono combinazioni dei valori esadecimali 00,33,66,99,CC,FF.
L'utilizzo di colori che non appartengono alla web palette su computer con scarse risorse grafiche può portare ad una approssimazione del colore (dithering) e a risultati sgradevoli.

2.5.7 Gamma

La gamma denota la luminosità complessiva del display. Più è alto il valore, meno luminosa è l'immagine sul display. Notare che ogni piattaforma ha una gamma diversa, quindi immagini create su Mac appaiono più scure sui sistemi Win.

2.6 Progettare il layout

Nel design tradizionale, le dimensioni del supporto vengono stabilite a priori e costituiscono un vincolo immutabile durante la progettazione del layout.
Nel design tradizionale viene definita la gabbia tipografica, che partiziona lo spazio disponibile in aree omogenee le quali conterranno testo, grafica, o (nel caso di supporti elettronici) elementi multimediali. È un elemento caratterizzante del design complessivo, quindi da mantenere costante durante la pubblicazione.

2.6.1 Prodotto editoriale stampato

Nei giornali è presente un modulo verticale (principalmente su 8 colonne) che costiuisce la base del layout del quotidiano.
Tutti gli altri elementi sono costruiti basandosi su questo modulo di base, rispettando allineamenti e regole di simmetria.
Per convenzione, la pubblicità viene posizionata nei due riquadri superiori e nella base del layout. L'articolo di fondo è sempre posizionato a sinistra. La zona superiore viene riservata alle notizie generali rilevanti mentre quella inferiore alla cronaca locale.

2.6.2 Vantaggi della gabbia tipografica

- Aspetti funzionali: magiore facilità nel reperimento dell'informazione da parte del lettore e maggiore facilità nell'interazione con i prodotti editoriali interattivi da parte dell'utente.

- Aspetti comunicativi: è l'elemento fondamentale dell'identità di un progetto grafico.
- Creatività: è possibile coniugare rigore e creatività, introducendo eccezioni rispetto alla regola data.

2.6.3 Progettare layout web

- Supporto: problematico assicurare la corrispondenza tra la dimensione complessiva della pagina e l'area visibile all'utente.
- Controllo del layout: nelle prime versioni di HTML non era possibile alcuna forma di controllo del layout della pagina. Il CSS permette un controllo accurato del layout, permettendo anche di ottenere layout diversi per media diversi.

2.6.4 Area sicura

- Per la visualizzazione: numero di punti dello schermo (pixel) disponibili per visualizzare l'informazione di una pagina web.
- Per la stampa: numero di punti dello schermo stampabile su carta.

L'area sicura dipende da configurazione HW e SW, browser e preferenze dell'utente. Nell'area sicura è meglio mettere gli elementi informativi fondamentali, gli artefatti fondamentali per l'interazione con il sito, e gli elementi grafici caratterizzanti.

2.6.5 Layout flessibili e fissi

- Layout flessibile: dimensioni delle aree fissate usando misure relative (es percentuale) come anche le dimensioni del carattere, per permetterne uno zoom e un rimpicciolimento.
  Vantaggi: pagina si adatta al display e alle preferenze dell'utente.
  Svantaggi: in alcuni casi possono verificarsi righe troppo lunghe e non facilmente leggibili.
- Layout fissi: dimensioni fissate usando misure assolute (es pixel).
  Vantaggi: risultato unico, maggior controllo.
  Svantaggi: problemi su determinate configurazioni HW e SW (ad esempio IE fino alla versione 6 non permette all'utente di riscalare un testo le cui dimensioni siano state fissate in pixel. Non viene garantito il controllo sul carattere tipografico (come descritto nella sezione font).

La scelta della tipologia di layout dipende da diversi fattori, ad esempio gli utenti, l'ambiente e la tipologia di servizio offerto.
È preferibile l'adozione di layout flessibili, che permettono un migliore adattamento a dispositivi eterogenei.

3 Pianificare un sito web

Progettare e sviluppare siti web è un'attività complessa che richiede un team di sviluppo interdisciplinare.

- Project Manager: responsabile coordinamento team, si occupa della schedulazione dei task e del controllo del budget.
- Esperto in marketing: identifica obiettivi del sito e utenti.
- Information Designer: è il responsabile dei sistema di strutturazione, classificazione, ricerca e navigazione all'interno del sito.
- Informatico: amministra il server, sviluppa e tiene aggiornati servizi e app.
- Web Designer: progettista del design e del layout. Crea relazioni tra gli elementi del sito.
- Giornalista/Responsabile editoriale: prepara e adatta i testi da inserire nel sito.
- Esperto in usabilità: responsabile valutazione usabilità dei prototipi e del sito finale.

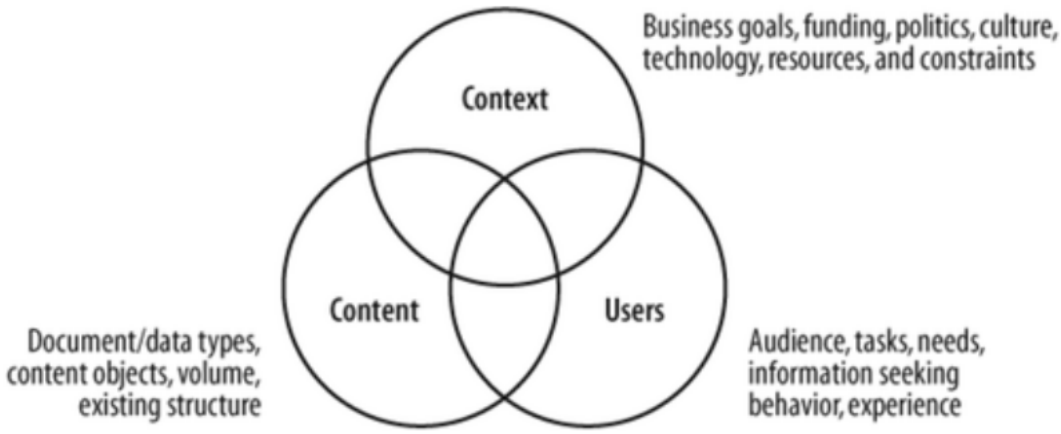
4 Architettura dell'informazione

Cos'è l'architettura dell'informazione?

- il design strutturale di ambienti informativi condivisi
- la combinazione di organizzazione, etichettatura, ricerca e sistemi di navigazione relativi a siti web e intranet
- l'arte e la scienza di dare forma a prodotti ed esperienze informative per supportare l'usabilità e la trovabilità.

Nell'architettura dell'informazione troviamo Sistemi (sistemi di ricerca, di navigazione, reti semantiche) e prodotti (strutture organizzative, vocabolari controllati, metadati, layout).

4.1 I 3 cerchi dell'architettura dell'informazione



- Contenuti: includono documenti, applicazioni, servizi, schemi e metadati. I parametri da considerare includono il produttore e il proprietario dei contenuti, il formato, la granularità, i metadati, il volume e la dinamicità dei contenuti.
- Contesto: i siti vengono definiti all'interno di un determinato contesto, con una missione specifica, obiettivi, strategia, staff, processi e procedure. L'architettura dell'informazione di un sito deve fornire un'immagine tangibile dell'organizzazione che lo promuove.
- Utenti: esistono diversità nelle preferenze degli utenti e nei comportamenti.

Contenuto, utenti e contesto sono componenti interdipendenti nell'ambito del processo di definizione dell'informazione.

4.2 Bisogni e comportamenti degli utenti

Modello informativo too-simple

- utente pone una domanda
- accade qualcosa
- utente riceve risposta

- fine ricerca

I problemi che possono sorgere sono che l'utente non sempre sa quello che vuole. Inoltre spesso la ricerca termina con un insuccesso, o un successo parziale. Il contesto viene ignorato.

4.2.1 Che cosa vogliono gli utenti?

Ci sono diversi tipi di bisogno, possiamo definirli con la metafora della pesca:

- **Il tiro perfetto**: quando gli utenti sanno quello che stanno cercando.
- **Trappola per aragoste**: gli utenti non hanno un'idea precisa di quello che stanno cercando, si aspettano di trovare qualcosa e di imparare qualcosa dal processo espolorativo, che possa guidarli verso una nuova ricerca.
- **Pesca con la rete**: utenti vogliono esaminare ogni elemento relativo ad un particolare argomento
- **Boa di segnalazione**: gli utenti vogliono ritrovare un elemento informativo utile

4.2.2 Comportamenti dell'utente

Gli utenti trovano l'informazione mediante:

1. searching (ricerca)
2. browsing (navigazione)
3. asking (facendo domande)

Searching, browsing e asking spesso sono integrati nella stessa sessione di lavoro, oppure utilizzati in iterazione.

4.2.3 Modello berry-picking

Gli utenti partono con un bisogno informativo, formulano una query (rihiesta informativa), e si muovono iterativamente attraverso percorsi potenzialmente complessi, raccogliendo progressivamente elementi informativi. Se il comportamento degli utenti rispecchia questo modello, è necessario che sia possibile spostarsi facilmente tra searching e browsing.

4.2.4 Modello pear-growing

Gli utenti partono con uno o pochi documenti che corrispondono esattamente a quello di cui hanno bisogno per cercare altri documenti con quelle caratteristiche (es. pagine simili di Google, oppure ritrovare documenti indicizzati con le stesse keyword).

4.3 Progettare la struttura dell'informazione

Internet dà la libertà di pubblicare l'organizzazione e la responsabilità di organizzarla (nel passato quest'ultimo compito era svolto da figure professionali come i bibliotecari).

4.3.1 Problemi nell'organizzazione dellinformazione

- Ambiguità: i sistemi di classificazione sono basati su un linguaggi (che può essere ambiguo), inoltre classificare oggetti e concetti astratti può essere difficoltoso.
- Eterogeneità: molti siti web sono eterogenei, forniscono cioè l'accesso a documenti a diversi livelli di granularità e formati multipli.
- Differenze di prospettiva: è indispendabile mettersi nei panni dell'utente.
- Diversità di politiche

4.4 Organizzare l'informazione

4.4.1 Schemi organizzativi

Permettono di suddividere gli oggettii informativi in raggruppamenti logici basandosi su proprietà caratterizzanti dei singoli oggetti. Sono suddivisi in schemi organizzativi esatti e ambigui.

- **Esatti**: l'informazione viene suddivisa in sezioni mutuamente esclusive. La progettazione e la manutenzione è facile, ma richiedono che l'utente conosca il nome specifico della risorsa che sta cercando (know-item searching). Degli esempi sono: schema alfabetico, schema cronologico, schema geografico.
- **Ambigui**: sono utili per realizzare e soddisfare uno stile di ricerca basato sulla serendipità. L'informazione è suddivisa in categorie nelle quali può essere difficile collocare l'oggetto. Sono più importanti e utili dei schemi organizzativi esatti, nel caso in cui non sappiamo che cosa stiamo cercando (exploratory searching).
  In questo caso la serendipità può essere supportata implementando una ricerca iterativa e interattiva, e coinvolgendo meccanismi di apprendimento associativo.
  Gli schemi organizzativi ambigui più comuni sono: schemi per argomento (topical, andando a definire l'universo degli argomenti che l'utente si aspetta di trovare), schemi orientati al compito (task oriented, dove il contenuto e le applicazioni sono organizzati come collezioni di processi, funzioni o compiti), schemi specifici per audience, schemi metaforici (utilizzati per far comprendere concetti nuovi collegandoli a concetti familiari, da usare con cautela ed è necessario evitare problemi di inconsistenza), schemi ibridi (mix di tutto).

Le facets (sfaccettature) permettono di accedere allo stesso informativo da angolature (schemi organizzativi) diversi. Definire più facets permette all'utente una maggiore interazione in quanto può navigare in diversi modi, è possibile inoltre associare una caratteristica dell'oggetto al facets (ad esempio un negozio di vestiti può essere navigato per marca, colore, taglia, ...).

4.4.2 Strutture organizzative

Definiscono le tipologie di relazione tra elementi o gruppi di oggetti dell'universo informativo. Esistono diverse strutture organizzative, ognuna con i sui punti di forza e debolezze.

- **Sequenze**: l'informazione è messa in sequenza. Le sequenze lineari sono adatte a siti didattici in cui l'utente deve procedere ordinatamente attraverso un insieme di materiali.

- **Gerarchia**: c'è un livello di parentela tra le informazioni, come una struttura ad albero. Gli utenti che utilizzano questa organizzazione sono in grado di sviluppare più facilmente un modello mentale della struttura del sito.
  Le categorie gerarchiche dovrebbero essere mutuamente esclusivo ma non è illegale posizionare un numero limitato di oggetti informativi in più di una categoria (cross listing) per creare una struttura poligerarchica. Posizionare un numero eccessivo di elementi in più cateagoria porta una perdita di valore di questa struttura.
  È necessario inoltre definire un equilibrio tra ampiezza (numero di opzioni ad ogni livello) e profondità (numero di livelli) della gerarchia (non oltre 10 opzioni, e 4 o 5 livelli).

- **Ipertesto**: innotiva modalità non lineare per strutturare l'informazione. Le unità informative possono essere collegate gerarchicamente, non gerarchicamente o in entrambe le modalità.
  Può essere un ostacolo per la formazione di un modello mentale del sito

- **Database**: è una collezione di record, dove ogni record ha un numero n di campi associati. I vantaggi sono: ricerca per campo, metadati associati ai dati, vocabolario controllato (imponibile un grado di consistenza che può risultare utile nella ricerca e navigazione), gestione più facile dei contenuti. Gli svantaggi sono che la strutturazione a record è rigida, e può essere costoso disporre ogni elemento del sito in un database.
  È utile usare un database per rappresentare una collezione di oggetti con le stesse proprietà. Molto spesso un datbase viene usato per modellare anche gli altri elementi strutturali del sito e non solo il contenuto principale.

4.4.3 Approccio top-down

Le strutture organizzative viste precedentemente vengono costruito usando un approccio top-down: l'information designer fornisce una soluzione relativa ad

un dominio informativo per un determinato sito web.
Spesso, però, non tutti gli utenti trovano una corrispondenza tra la struttura progettata dal designer e il proprio modello mentale relativo al sito. Per limitare questi problemi è possibile fornire soluzioni poligerarchiche, motori di ricerca interni, mappe del sito oppure permettere agli utenti di costruire la gerarchia informativa con tecniche quali il free listing e il card sorting.

- **Free listing**: permette di coinvolgere gli utenti nella definizione dei contenuti del dominio. Viene richiesto di formulare un elenco di elementi informativi a partire dalla descrizione di un tema fornita da chi gestisce il test.
- **Card sorting**: permette di coinvolgere gli utenti nella strutturazione dei contenuti. Viene richiesto di sudividere in gruppi una lista di schede etichettate. Si può usare l'open card sorting (agli utenti vengono fornite schede con etichette relative al contenuto del sito ma senza gruppi prestabiliti; l'utente deve quindi creare i gruppi e assegnare le etichette descrittive dei gruppi stessi), o il closed card sorting (agli utenti vengono fornite sia le schede con etichette dei contenuti del sito, sia i gruppi con le etichette già definite; l'utente deve riempire i gruppi con le schede). L'open card viene usato per creare un'architettura informativa, per ottenere feedback su quali contenuti vengono inseriti in uno stesso gruppo, e capire quali etichette vengono utilizzate dagli utenti per descrivere il contenuto. Il closed card sorting, invece, risulta utile per testare il design di un'architettura dell'informazione, e per ricevere feedback sull'efficacia delle etichette.

Il free listing e il card sorting mirano ad aumentare la trovabilità degli elementi di un sito web. Per validare la gerarchia di un sito viene usato il tree testing: all'utente viene proposto una serie di task consistenti nel trovare un determinato elemento informativo facente parte di una struttura gerarchica. L'utente una volta raggiunto l'elemento informativo che reputa corretto deve confermare la scelta. In un tree testing vengono considerati: tempo impiegato, precisione (percentuale di utenti che non sono tornati indietro) e percentuale di successo.

4.4.4 Approccio botttom-up

Si costruisce la struttura informativa dal basso, dando modo all'utente di marcare gli elementi informativi che sta navigando con un set di tag. Una tecnica è il free tagging, ossia categorizzazione collaborativa di elementi informativi del web che può far emergere una forma di organizzazione complementare (o alternativa) all'approccio top-down.
Si può usare una folksnonomia allargata (accettati tutti i possibili tag, anche ripetuti, usata da Delicious) o una folksonomia ristretta (per ogni risorsa il sistema non accetta che un utente inserisca tag già inseriti da altri utenti, usata da Flickr). I vantaggi di questo approccio sono: l'approccio è meglio di niente, in quanto molte volte non è sempre possibile trovare e applicare un vocabolario controllato a tutte le situazioni. Gli svantaggi invece sono: non c'è una

maggiore trovabilità degli elementi, molto spesso viene ignorata l'importanza del contesto, e la mancanza di un vocabolario controllato può creare diversi problemi legati alla semantica, all'ononimia (stesso tag per concetti diversi), polisemia, desinenze. Per risolvere alcuni limiti è possibile suggerire all'utente i tag da inserire da una lista di popular tag o da una lista di tag raccomandati, oppure visualizzando una lista di termini le cui lettere iniziali corrispondono ai caratteri inseriti dagli utenti.

4.4.5 Creare sistemi organizzativi coesi

Usare schemi organizzativi esatti se l'utente sa quello che sta cercando, altrimenti utilizzare schemi ambigui, che sono migliori per la navigazione e l'apprendimento associativo. Quando è possibile è opportuno utilizzare entrambi i tipi di schema.

5 Sistemi di navigazione

I sistemi di navigazione devono fornire all'utente il contesto, devono permettere all'utente di spostarsi in modo flessibile tra le unità informative. È necessario bilanciare la flessibilità e la fornitura di troppe opzioni.

5.1 Il design di sistemi di navigazione

L'utente si sposta attraverso le unità informative di un sito web in due modi:

- Browsing (selezione di elementi ipertestuali presenti nella presentazione di partenza, organizzati in gruppi e facenti riferimento alle strutture organizzative del sito)
- Searching (navigazione attraverso un'interrogazione del sistema, svolta attraverso la compilazione di un modulo e la selezione di una della unità informative proposte come risposta)

Integrare questi due meccanismi permette ad un utente di navigare nel sisto attraverso una modalità mista.
Molti browser inoltre forniscono un supporto alla navigazione (aprire un url per accesso diretto, bookmarks, uso di colore nei link ipertestuali, tasti avanti e indietro). Non bisogna danneggiare questo sistema di navigazione (modificare colore link, rimuovere sottolineatura dei link, nascondere l'url di destinazione)

5.2 Costruire il contesto

È necessario costruire quindi un contesto, ad esempio includendo in tutte le pagine il nome dell'ente/azienda oppure un logo e presentando in tutte le pagine la struttura dell'informazione gerarchica e la posizione dell'utente rispetto ad essa.

5.3 Navigation Stress Test

È utile per verificare la capacità di un sito di rappresentare il contesto. Si parte da una pagina a caso, e si rispondono alle domande:

- So dove sono?
- In che sezione principale del sito mi trovo?
- Qual'è la pagina padre di questa pagina?
- Dove mi condurrà la pagina in cui mi trovo?
- I link suggeriscono la destinazione?
- I link sono abbastanza differenziati per aiutarmi a scegliere un link piuttosto che un altro?

5.4 Tipologie di Sistemi di Browsing

Un sito complesso può includere diversi sistemi di navigazione.

5.4.1 Sistema Gerarchico

Gerarchia informativa, permette la navigazione da elementi padre ad elementi figlio, e viceversa.

5.4.2 Sistema Globale

Permette la navigazione verticale (da elementi padre a figlio, anche a passi maggiori di 1) e la navigazione orizzontale (tra elementi dello stesso padre).

5.4.3 Sistemi Locali

Sistema di navigazione utilizzato in un sottosito, cioè in una sezione del sito nella quale le unità informative sono legate da relazioni peculiari (es. catalogo online di un'azienda). È importante estendere il sistema Globale anche all'interno di sottositi.

5.4.4 Sistemi Ad hoc - Trasversali - Contestuali

Le relazioni tra contenuti informativi non riconducibili ad attraversamenti verticali o orizzontali della gerarchia, ad esempio i links contenuti nel corpo di un paragrafo (embedded links) oppure links evidenziati separatamente del layout della pagina.

5.5 Elementi di browsing integrati

5.5.1 Menu di navigazione

È una collezione di links ipertestuali, sottoforma di testo, imagemap o immagini separate.
La progettazione dell'architettura può influenzare la scelta dell'implementazione. Ad esempio, siti con aree in espansione o variabili scelgono soluzioni testuali. Per la navigazione globale è opportuno usare una barra grafica, per la navigazione locale una barra testuale.
I menu permettono di effettuare la navigazione orizzontale all'interno del sito.

5.5.2 Shortcuts

Sono elementi di navigazione che risiedono tipicamente nella home page di un sito: possono essere singoli oppure utilizzati in gruppi.
Permettono all'utente un accesso diretto a elementi che si trovano nei livelli più profondi della gerarchia informativa del sito (sono una navigazione verticale).
Sono associati all'esigenza di rendere noto all'utente l'esistenza di un elemento o di un gruppo di elementi informativi.

5.5.3 Menu pop-up e pull-down

Permettono di rappresentare molti link in forma compatta, nascondono opzioni. Vegono utilizzati per effettuare selezioni secondarie oppure in contesti informativi molto affollati. È sconsigliato utilizzarli per le opzioni informative principali nei sistemi desktop (nei layout per smartphone invece è possibile utilizzarli per ottimizzare lo spazio).
È opportuno che la riga di menù visibile indichi la tipologia dell'informazione.
È possibile anche specificare eventi che possono attivare l'apertura del menu (ad esempio uno swipe, la pressione di un tasto, al passaggio del mouse).

5.5.4 Drawer

I drawer permettono di rappresentare links in forma compatta: può essere utile utilizzarli nei casi in cui sia necessario un livello di strutturazione elevato dei menu pop-up e pull-down.
Vengono utilizzati nei dispositivi con uno schermo piccolo.

5.6 Sistemi di browsing

- Navigazione gerarchica (può essere molto limitante)
- Navigazione globale
- Navigazione locale

- Navigazione trasversale

  Embedded links (navigazione adhoc con link nel testo)

  Navigazione trasversale

  See also

5.7 Elementi di browsing remoti

Sono elementi che forniscono un punto di vista esterno alla gerarchia base del sito. Sono utilizzati come complemento agli elementi di navigazione integrati.

5.7.1 Indice dei contenuti

Rappresenta i livelli della gerarchia informativa, l'indice attivo (che impiega link ipertestuali) facilità l'accesso non lineare ai contenuti.
È utile utilizzarlo quando il sito ha un'organizzazione gerarchica.

5.7.2 Indice analitico

Alternativa per siti che non hanno ua forte organizzazione gerarchica.
È utile per gli utenti che sanno quello che stanno cercando; gli elementi dell'indice devono puntare a componenti del sito dove prevalgono i contenuti piuttosto che a componenti dove prevalgono le strutture di navigazione.

5.7.3 Mappa del sito

È una rappresentazione grafica dell'architettura di un sito.
La mappa evidenzia le relazioni tra le componenti informative.
Deve rappresentare gli elementi informativi importanti e le loro relazioni in modo chiaro e significativo.
Le mappe non si prestano ad una lettura TTSS, è preferibile quindi affiancare una mappa ad un indice dei contenuti.

5.7.4 Tour guidato

Utile per introdurre ai nuovi utenti i contenuti del sito. Dovrebbe avere una navigazione lineare ed essere eseguibile dalla home page del sito.

5.8 Sistemi di browsing personalizzati

Possiamo distinguerli in:

- Adattivi: le opportunità di navigazione vengono selezionati dal sistema in abse ad alcuni parametri, come il profilo dell'utente o lo storico di utilizzo.
- Adattabili: l'utente può scegliere diverse opzioni di navigazioni differenziate.

5.9 Sistemi di navigazione sociale

Permettono all'utente di navigare il sito utilizzando l'attività di tagging degli altri utenti, si costruisce quindi una folksnomia in modo progressivo.
Può essere l'unico sistema di navigazione oppure affiancarne un altro.

5.9.1 Tag clouds

Una folksonomia è rappresentata da una tagcloud, che esprime la frequenza di utilizzo dei tag attraverso la dimnesione del carattere tipografico. Può essere costruita a partire da una folksonomia allargata che ristretta.

5.9.2 Interfaccia per l'inserimento di tag

Oltre all'interfaccia di navigazione può essere presente l'interfaccia di inserimento dei tags (nei sistemi di tipo bottom-up l'utente può contribuire ad arricchire la classificazione).

5.9.3 Integrazione nel layout delle interfacce di navigazione sociale

Le interfacce dei sistemi di navigazione possono essere presenti nel layout del sito specifico, nel layout di un altro sito che funge da collettore di tagging degli utenti sul web, oppure secondo modalità miste.

5.9.4 Utenti, risorse e annotazioni

1. Annotazioni top-down: i singoli siti forniscono all'utente fornitore di contenuti le interfacce per inserire risorse e annotarle. L'utente che gestisce la singola risorsa coincide con l'utente autorizzato ad annotare la risorsa (solo lui può annotare la risorsa). Abbiamo una integrazione totale dell'interfaccia di annotazione nel layout.

2. Annotazioni bottom-up tipo 1: le risorse informative e le annotazioni risiedono in unico repository. Le interfacce e le risorse per annotarle sono fornite da un gestore unico.
   Abbiamo comunque un'integrazione totale delle interfacce di annotazione nel layout di presentazione, ma in questo caso è consentito ad utenti diversi l'annotazione di risorse gestite da altri.

3. Annotazioni bottom-up tipo 2: le risorse informative e le annotazione risiedono in siti diversi. È difficile quindi un'integrazione delle interfacce di annotazione nel layout di presentazione delle risorse informative.

5.9.5 Fake tagcloud

Esistono anche della tagcloud fake: in questo caso i contenuti della tagcloud, e la grandezza delle parole è una scelta redazionale, per mettere in risalto alcuni contenuti. In questo caso, non possiamo parlare di tagging sociale.

6 Sistemi di ricerca

6.1 Quando implementare un sistema di ricerca?

- Quando abbiamo molta informazione da percorrere attraverso sistemi di browsing
- Quando il sito è costituito da componenti frammentati
- Quando dobbiamo imparare dagli utenti (analisi search-logs)
- Quando gli utenti si aspettano che ci sia
- Quando il sito è dinamico

6.2 Zone di ricerca

Non è sempre opportuno indicizzare tutto il sito.
Creare search-zones riduce l'effetto apple-and-oranges, ossia la ricerca effettuata su aree eterogenee di contenuti che può essere un ostacolo al ritrovamento dell'informazione.
Le zone di ricerca sono parti di sito che sono state indicizzate separatamente.
Si possono creare suddividendo logicamente o fisicamente i documenti. È possibile crearla a partire da schemi e strutture organizzative.
Con le zone di ricerca però andiamo ad aggiungere un ulteriore complessità alla ricerca.

6.3 Pagine di destinazione e navigazione

Le pagine di destinazione contengono il contenuto vero e proprio.
Le pagine di navigazione hanno lo scopo di condurre gli utenti verso le pagine di destinazione. Non vengono indicizzate.
Spesso, le pagine appartengono ad entrambe le categorie.

6.4 Componenti di una pagina

La nascita dei sistemi CMS (Content Management System) rende più facile l'identificazione degi elementi della pagina (contenuto informativo, menu di navigazione, pubblicità, disclaimers...) e la loro inclusione o meno dall'indicizzazione.
Usare un CMS permette di selezionare gli elementi da indicizzare ad un livello di granularità più accurato rispetto a quello della pagina.

6.5 Presentare i risultati

6.5.1 Quali?

- mostrare come risultato all'utente che sa quello che sta cercando meno informazione rispetto a quella presentata all'utente che non sa quello che sta cercando.

- Agli utenti del primo caso possono bastare alcuni elementi rappresentativi del contenuto.
- La seconda categoria di utenti richiede contenuti più descrittivi.
- In alcuni casi, l'utente può scegliere cosa visualizzare.

6.5.2 Quanti?

Dipende da due fattori, dalla quantità di componenti informativi per componente e dalla caratteristica del dispositivo.
Si parte dalla presentazione di un gruppo limitato di documenti (10 per pagina) e si lascia la laibertà di configurare al sistema.
È opportuno visualizzare il numero complessivo di documenti ritrovati e fornire un sistema di browsing per permettere una rapida navigazione tra i risultati.

6.5.3 Elencare i risultati

Esistono due metodi per elencare i risultati.
Sorting: i risultati vengono ordinati secondo un criterio basato su una della componenti del documenti (utile quando l'utente deve comparare due o più documenti).
Ranking: risultati ordinati con criteri come la rilevanza del documento, dedotta analizzandone i contenuti (condiderando quanti termini della query sono presenti nel documento e quante volte, la vicinanza e la locazione di questi termini)o la sua popolarità. (ad esempio PageRank di Google). Oppure può essere realizzato un ranking attraverso il giudizio di esperti, i quali valutano il valore dell'informazione, o attraverso un modello pay-for-placement.

6.5.4 Ragruppare i risultati

Una tecnica alternativa (o complementare) al Sorting e al Ranking consiste nel raggruppare i risultati in base a qualche aspetto comune.
Come possiamo raggruppare i risultati? Attraverso l'utilizzo di metadati (tipo di documento, data creazione), attraverso l'uso di metadati applicati manualmente (audience, contenuto, tags). Alcuni tool cercano di derivare autonomamente alcuni cluster di contenuti considerando i contesti possibili a partire da una keyword inserita dall'utente.

6.5.5 Esportare i risultati

Un utile complemento può essere quello di esportare i risultati (stampare, inviare via mail, salvare).
Quando si vuole salvare un insieme di elementi interessanti è utile fornire una selezione di un sottoinsieme di risultati.
Il salvataggio di una ricerca, inoltre, è un'alternativa utile per i domini dinamici, per i quali la riesecuzione dell'interrogazione in un altro momento può portare nuovi risultati.

6.6 L'interfaccia di ricerca

È necessario mantenere l'interfaccia di ricerca il più semplice possibile. L'utente non deve utilizzare operatori logici, non deve preoccuparsi di utilizzare un vocabolario controllato e la ricerca viene fatta sull'intero sito.
Per educare l'utente è possibile creare una pagina di consigli, oppure inserire la possibilità di restringere la ricerca ad una search zone o inserendo un collegamento ad un modulo di ricerca avanzata.

6.7 L'interfaccia per i risultati

È necessario fornire un supporto alla revisione, ad esempio descrivendo gli eventuali filtri applicati, mostrando gli operatori booleani applicati implicitamente alla ricerca, mostrando le caratteristiche utilizzate per la presentazione (ad esempio il tipo di sorting) ed evidenziando il numero dei risultati ottenuti dalla ricerca.

6.8 Interfacce di ricerca basate sulla classificazione sociale

In alcune situazione sono disponibili sistemi di ricerca dei tags.
Succede per le situazioni in cui i tags costituiscono il sistema prevalente di classificazione dell'informazione o nei sistemi caratterizzati dalla presenza di molti tags associati alle risorse.
È importante rendere consapevole l'utenza che la ricerca delle informazioni avviene sull'insieme dei tags e non sull'insieme delle parole che compongono la pagina web.

6.9 Trovabilità

La trovabilità è definita come $Finding = Searching + Browsing$. Per aumentare la trovabilità degli elementi è necessario quindi integrare le due modalità di navigazione di un sito, rendendo possibile passare da una navigazione ad un'altra in modo semplice.

7 HTML

Formato non proprietario basato su SGML e può essere creato e manipolato da una vasta gamma di tool.
SGML è un sistema di regole per definire linguaggi basati su marcatori. Stabilisce il sistema di descrivere documenti in termini di struttura, indipendentemente dall'apparenza.
HTML utilizza un sottoinsieme delle caratteristiche di SGML. Gli elementi di una pagina web sono identificati da marcatori (tag) che danno istruzioni al browser sul ruolo dell'elemento e sulla modalità di visualizzazione del contenuto.

HTML si basa sul principio di mantenere separata l'informazione relativa allo stile dalla struttura del documento.

7.1 Timeline di HTML

- HTML
- HTML 2.0 (1995)
- HTML 3.2 (1997)
- HTML 4.0 (1997)
- HTML 4.01

  XHTML 1.0

  XHTML Basic (browser web per palmari)

  XHTML 1.1 (basato sul framework di modularizzazione)

  XHTML Mobile Profiler 1.0 (sviluppato da Nokia, per mobile phone)

  XHTML 2.0
- HTML 5

Mentre XHTML 2.0 (evoluzione di XHTML 1.1) è XML puro, HTML 5 rimane compatibile con gli standard precedenti e fornisce estensioni per lo sviluppo di applicazioni.

7.2 Tag HTML

Un tag X/HTML è racchiuso tra < e >. Il tag è formato dal nome dell'elemento seguito da una lista (opzionale) di attributi. Abbiamo due tipologie di tag:

- Elementi non vuoti: $< elemento > testo < /elememnto >$
- Elementi vuoti: $< elemento/ >$

Lo standard XHTML richiede sempre il tag finale, opzionale in HTML. Se il contenuto non è presente, va sempre utilizzata la forma con tag iniziale e tag finale per compatibilità con i browser HTML.

7.2.1 Elementi vuoti

Alcuni tag non richiedono il tag dinale. Va inserito i carattere / prima della fine del tag.
È opportuno inserire uno spazio prima del carattere / per compatibilità con i vecchi browser.

7.2.2 Nidificazione di Tag

Un tag può essere inserito all'interno di un altro tag. Non è possibile sovrapporre i tag.
Documenti XHTML sono ben formati se tutti gli elementi hanno il tag di chiusura e sono annidati correttamente.

7.2.3 Attributi di tag

Gli attributi vengono aggiunti per estendere o modificare le azioni di un tag.
Va inserito sempre nel tag iniziale, è possibile inserire molti attributi, separati da spazio. Non è importante l'ordine.
Tutti i tag e attributi devono essere scritti con lettere minuscolo e i valori degli attributi devono essere racchiusi tra doppi apici (richiesto in XHTML).
I valori degli attributi possono venire scritti sia maiuscoli che minuscoli.
In HTML 5, esiste l'attributo booleano, con una sintassi semplificata. La presenza dell'attributo indica un valore true, la sua assenza false.
Se dobbiamo utilizzare caratteri speciali ($<, >$ e " ",) come parte del valore di un attributo possiamo usare stringhe alternative dette CER (es. $\"$, $\<$).

7.3 Caratteri ignorati

In una pagina HTML vengono ignorati:

- Interruzioni di linea
- Taulatori e spazi multipli
- Tag $< p >$ nidificati
- Tag sconosciuti

7.4 Struttura di un documento XHTML 1

Un documento XHTML 1.0 è composto da:

- Prologo XML (opzionale)
- Dichiarazione del tipo di documento ($<!DOCTYPE>$)
- Sezione di testa ($< head >$)
- Corpo del documento ($< body >$)

L'head e il body devono comparire all'interno del tag $< html >$.

7.4.1 Prologo XML

Serve a specificare la versione XML e la codifica dei caratteri.
W3C ne raccomanda l'utilizzo, ma alcuni browser non lo gestiscono correttamente. Per inserire la codifica dei caratteri al di fuori del prologo è possibile inserire un elemento content-type nella sezione head del documento.

7.5 Specifica dei caratteri

È necessario stabilire una relazione tra i singoli caratteri utilizzati e il codice numerico che li rappresenta.
Diversi standard effettuano questa mappatura:

- Famiglia ISO-8859: set di caratteri multilingua a 8 bit per scrivere in lingue alfabetiche (ad esempio, Latin-1, Latin-2).
  Ha una scarsa portabilità del testo al di fuori di ambiti regionali, ma anche tra sistemi operativi diversi (nei primi anni del Web Windows e MacOs utilizzavano standard proprietari)

- Unicode: standard universale, codifica tutti i caratteri utilizzati nei linguaggi scritti utilizzati in tutto il mondo.

È inoltre possibile, per identificare un carattere, utilizzare una stringa CER (&#*nnnn*, dove nnnn sono i riferimenti numeri ad un elemento del set di caratteri espressi in forma decimale o esadecimale (in quest'ultimo caso, prima del riferimento, va inserita una x, xnnnn)). Oltre ad usare un codice numerico, è possible utilizzare un nome (&*name*).
Le nuovi versioni dei Sistemi Operativi supportano unicode, rendendo quindi inutile le CER. CER comunque permette una corretta compatibilità verso i sistemi meno recenti.

7.6 DTD (Document Type Definition)

Un documento X/HTML valido deve iniziare con una dichiarazione nella quale si dice quale versione di X/HTML viene utilizzata.
Ogni versione è definita da una DTD, che ne descrive in un liguaggio preciso e leggibile da computer la sintassi e la grammatica permesse nel documento.
I DTD sono disponibili online.
Quando il documento XHTML viene esaminato da un validatore, il contenuto viene validato rispoetto ad DTD indicato.
HTML 4 E XHML 1.0 fanno riferimento a 3 diverse DTD che includono 3 diversi sottoinsieme di tag consentiti nel documento (questo perchè questi formati sono formati di transizione):

- STRICT: codice XHTML privo di tag o attributi di presentazione

- TRANSITIONAL: documento con tag di presentazione

- FRAMESET: da utilizzare se il documento utilizza, oltre ai tag previsti dal TRANSITIONAL, anche le frames.

7.7 Il tag $< html >$

Contiene una dichiarazione di namespace.
Un namespace fornisce un metodo per qualificare elementi ed attributi utilizzati

da un documento XML, associandoli con i namespaces identificati da un riferimento URI.
Per documenti che fanno riferimento a vocabolari multipli, viene descritto un meccanismo che assegna nomi espansi ed attributi per ottenere un'identificazione unifica, ad esempio usando i attributi xml:lang e lang per specificare la versione xml e la lingua del documento.

7.8 Conformità

Un documento XHTML è conforme se:

- l'elemento radice è head.
- l'elemnto radice deve indicare lo spazio dei nomi di XHTML usando l'attributo xmlns.
- prima della radice deve esserci una dichiarazione di DOCTYPE, il quale conterrà un'identificatore che si deve riferire ad una delle 3 DTD (strict, transitional, frameset).
- deve essere validato rispetto ad una delle 3 dtd.

7.9 Il tag $< head >$

Non ha attributi, contiene tag utilizzati per definire i contenuti del documento:

- $< title >$: descrizione contenuti della pagina. Elemento importante per l'indicizzazione.Obbligatorio in XHTML 1.0. Solo caratteri ascii.
- $< base >$: localizzazione base sul server web, da utilizzare come riferimento per i links presenti nel documento.
- $< link >$: definisce la relazione con un altro documento (usato ad es. per collegare il documento ad un foglio di stile)
- $< script >$

7.9.1 Il tag $< meta >$

Il tag meta è presente in head, ed è utilizzato per molti scopi. I dati inclusi nel tag sono utili per server, browser, e sono invisibili al navigatore.
È necessario specificare l'attributo content, per fornire un valore all'informazione.
Ci sono due categorie di tag meta:

- http-equiv: viene processata come se provenisse da un header di risposta http. Fornisce di conseguenza informazioni che condizionano le modalità con le quali il browser manipola il docuemento.
  L'attributo http-equiv con valore content-type permette di specificare il set di caratteri utilizzati.

- name: permette di inserire informazioni riguardanti il documento (descriptions, keywords, author, robots (per prevenire l'indicizzazione), rating...).

7.10 Il tag $< body >$

Contiene le informazioni che verranno presentata all'utente sulla pagina.
Nel passato sono stati definiti un insieme di tag di presentazione, ora deprecati.

7.11 HTML 5

7.11.1 Struttura di un documento HTML 5

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Page Title</title>
    </head>
    <body>
    </body>
</html>
```

7.11.2 Doctype

La dichiarazione di tipo del documento viene semplificata: HTML 5 supporta il contenuto esistente di HTML 4.01 o di XHTML 1.0, e le versioni successive dovranno supportare il contenuto di HTML 5. Quindi non è necessario specificare la vesione di HTML.

7.11.3 Codifica dei tag e attributi

Puoi scrivere come vuoi, puoi scegliere di mettere le virgolette o meno, di scrivere i tag in maiuscolo...

7.11.4 Obsolescenza

Non ci sono attributi deprecati, ma solo obsoleti, perchè si vuole mantenere la compatibilità con il passato.
Un documento con elementi o attributi obsoleti sarà non conforme.
Tuttavia, alcuni elementi di presentazione sono stati ridefiniti per superare i limiti passati.

8 HTML - parte 2

8.1 Separare struttura e presentazione

È fondamentale separare struttura e presentazione. La separazione permette di definire unicamente i contenuti e di definire molteplici stili di presentazione a seconda dei dispositivi utilizzati e delle preferenze degli utenti.

8.2 Markup strutturale per il testo

Possiamo suddividere i tag definiti per marcare le caratteristiche di un testo in 2 gruppi:

8.2.1 Block-level elements

Elementi per definire blocchi strutturali. Abbiamo, come tag:

1. $< p >$: denotare paragrafi. Il contenuto di questo tag viene visualizzando partendo da una nuova linea di testo; sopra e sotto il paragrafo viene aggiunta una nuova riga vuota.

2. $< hn >$ (con n un numero da 1 a 6): rappresentare livelli di intestazione di un testo (titoli). Le intestazioni di livello inferiore devono essere precedute da quelle di livello suoeriore.
   Non è consigliabile saltare livelli.
   Le intestazioni vengono rappresentate in grassetto, h1 viene rappresentato con un carattere molto grande, i successivi con caratteri decrescenti. Nella visualizzazione vengono aggiunti automaticamente un ritorno a capo e una riga vuota sopra e sotto l'intestazione.

3. liste: viene aggiunta una riga vuota prima e dopo la lista. Abbiamo 3 tipologie di liste: liste ordinate, liste non ordinate, definizioni. È possibile nidificare liste ordinate e non.

   $< ul >$(liste non ordinate): davanti ad ogni elemento viene inserito un pallino (nb: un elemento è identificato dal tag $< li >$). Usando fogli di stile è possibile modificare l'identificatore.

   $< ol >$ (liste ordinate): davanti ad ogni eemento viene inserito un numero arabo.

$< dl >$ (definizione): serve per denotare termini ($< dt >$) e definizioni (dd). I termini vengono visualizzati sul margine sinistro della pagina e le definizioni vengono spostate a destra rispetto al margine.

8.2.2 Inline Styles

Tag che si riferiscono ad aspetti di struttura o di presentazione relativi a stringhe di caratteri inseriti nel flusso di un testo. Condizionano il testo contenuto all'interno di essi, senza agigungere righe vuote prima e dopo. Possiamo suddividerli in 2 categorie.
La prima sono gli stili logici (strutturali), che descrivono il significato, il contesto o l'uso dell'elemento racchiuso. L'aspetto di presentazione è a discrezione del browser.
Per definire partizioni logiche vengono introdotti questi due tag: $< div >$ e $< span >$. Div marcatore generico utilizzato per dividere un documento in sezioni. È un elemento a livello di blocco (testo marcato inizia su una nuova riga e una riga vuota inserita prima e dopo il tag). Il tag Span invece è un inline generico da applicare ad una stringa all'interno di un paragrafo o altri contenuti. Questi tag vengono utilizzati in associazione con gli attributi class e id, e delle regole di stile associate.
Gli stili fisici, invece, fornicono istruzioni sulle modalità di rappresentazione. Sono sconsigliati. Un esempio molto comune e utilizzato è il tag $< font >$, che permette di specificare il carattere tipografico, ha attributi quali face (specificare il font), size (dimensione, da 1 a 7).
È bene sottolineare che la specifica del carattere tipografico (anche se fatta in CSS) non garantisce la visualizzazione del testo secondo le caratteristiche definite. Ad esempio, può non essere disponibile il carattere nel sistema.
Il tag font inoltre può impedire ai motori di ricerca di identificare gli elementi più importanti della pagina da un punto di vista semantico (se ad esempio modifichiamo la size nel tag font per far apparire il testo più importante). L'indicizzazione del sito avviene attribuendo pesi sbagliati a diversi elementi, generando problemi nella ricerca da parte degli utenti dei motori di ricerca stessi.

8.3 Link Ipertestuali

Il tag $< a >$ viene utilizzato come contenitore di una stringa di testo (o immagine) che funge come collegamento ipertestuale. Si possono utilizzare url assoluti oppure relativi.
È possibile inoltre collegarsi, oltre che a pagine differenti, a sezioni o punti specifici di un documento come aiuto alla navigazione (usando # seguito dall'id dell'elemento).

8.4 Link non ipertestuali

Sono, ad esempio mail link o telnet link (viene aperta una sessione di terminale su un client), FTP link o altro.

8.5 Il tag $< link >$

Viene utilizzato per definire una relazione tra il documento corrente e un altro documento esterno. Possono esserci più tag link all'interno dello stesso documento, gli attributi più importanti sono: href, rel, rev e type.

9 HTML - Parte 3

9.1 Tabelle $< table >$

Originariamente, le tabelle vennero sviluppate per rappresentare i dati.
Possono però anche venire utilizzate come griglie strutturali della pagina e/o per l'allineamento di testo, oppure come contenitori di immagini multiple.
Le tabelle sono costituite da celle organizzate in righe. Le colonne sono definite implicitamente.

- $< table >$: definisce inizio e fine della tabella.
- $< tr >$: inizio e fine di una riga.
- $< td >$: inizio e fine di cella (dove va posto il contenuto).
- $< caption >$ (prima di ogni riga): titolo o descrizione della tabella
- $< th >$ (elemento di testa). descrivere il contenuto della colonna sottostante.

L'attributo summary, facoltativo, di table può essere usato per dare una descrizione estesa dei contenuti della tabella, per gli utenti di dispositivi braille e convertitori TTS, e anche per i motori di ricerca.

9.2 Colspan

L'attributo colspan permette l'espansione di colonne. Il valore attribuito a colspan determina il numero di colonne attraverso le quali la cella si espande, da destra a sinistra.

9.3 Rowspan

L'attributo rowspan permette l'espansione di righe. Il valore attribuito a rowspan determina il numero di righe attraverso le quali la cella si espande, dall'alto verso il basso.

9.4 Gruppi di righe e colonne

Le righe di una tabella possono essere raggruppate in:

- $< thead >$: sezione di testa

- $< tfoot >$: sezione conclusiva
- $< tbody >$: una o più sezioni per i contenuti.

Le prime due sezioni possono venire scritte di seguito nel codice, per poter avere una aprziale visualizzazione della tabella prima che tutto il contenuto di tbody venga caricato.
I gruppi di colonne vengono realizzati con $< colgroup >$, il quale delimita un gruppo di colonne dal punto di vista strutturale. Il numero di colonne è definito dall'attributo span o dal numero di tag col contenuti, ognuno moltiplicato per il proprio valore di span.

9.5 Attributi XHTML di presentazione della tabella (deprecati)

9.5.1 Dimensione

Le tabelle vengono rappresentate con la minima dimensone necessaria a contenere i dati.
Per specificare una dimensione della tabella, abbiamo gli attributi width e height nel tag table.
Per le dimensioni della cella, possiamo usare width nei tag td e th (larghezza valida per tutta la colonna) e height nei tag td e th (altezza valida per tutta la riga).

9.5.2 Bordi

Nel tag table possiamo inserire l'attributo border, che permette di visualizzare la tabella senza borsi.

9.5.3 Spaziatura tra e nelle celle

Cellspacing permette di definire la spaziatura tra le celle, cellpadding definisce la spaziatura tra il bordo interno della cella e i contenuti della cella.
Entrambi gli attributi vanno messi in table.

9.5.4 Altri attributi

bgcolor, align su celle (allineamento orizzontale del testo) valign su celle (allineamento verticale del testo), align su table (posizionamento della tabella nel layout). NB: font non può essere un contenitore della tabella (se vogliamo cambiare testo alla tabella, dobbiamo ripetere il tag font per ogni elemento).

9.6 Tips and Tricks

1. Caratteri indesiderati: eliminare ogni carattere non necessario nel tag td

2. Celle che collassano: le celle che non contengono nessun carattere possono collassare. È possibile inserire la stringa CER di codifica dello spazio per evitare questo (&*npsp*).

3. Caricamento di una tabella: prima di mostrare la tabella, il browser deve caricarla tutta. Per visualizzare progressivamente i contenuti è possibile utilizzare gruppi di righe e di colonne.

9.7 Tabelle come griglie strutturali di una pagina

Per definire la griglia strutturale della pagina, molti siti antichi utilizzavano una tabella costituita da 2 o più celle. Bisogna però stare attenti ai margini, perchè una tabella viene posizionata rispettando i margini del browser.

10 CSS

I fogli stile permettono di applicare regole di presentazione agli elementi che caratterizzano una pagina XHTML. Un foglio di stile è un CSS (Cascading Style Sheet), dove con cascading ci riferiamo al fatto che l'informazione di stile può essere definita a più livelli, generando alla fine un unico folgio di stile virtuale nella quale si compongono tutte le regole provenienti da origine diverse e con specificità diverse.
I vantaggi sono: stile e struttura separati, maggior controllo sui caratteri e sul layout della pagina, produzione di documenti più piccoli, manutenzione più facile del sito, facilità di apprendimento.
Lo svantaggi principalmente è uno solo: il supporto dei browser del passato.

10.1 Evoluzione dei CSS

1. CSS1: 1996

2. CSS2: 1998 (include metodi avanzati per il posizionamento degli elementi, in grado di sostituire tabelle e frames)

3. CSS 2.1: 2011

4. CSS 3: 2011 (selettori e colori)

10.2 Regole di sintassi

CSS contengono una o più regole per descrivere la presentazione di un elemento nella pagina.
La sintassi è composta da: selettore { proprietà: valore; }
Il selettore identifica l'elemento che verrà condizionato dalla regola. La dichiarazione (composta da coppie proprietà/valori) specifica lo stile da applicare al'elemento.
I valori sono dipendenti dalle proprietà.

10.3 Selettori

o 2 tipi di selettori (i selettori sono la parte della regola che identifica l'elemento a cui lo stile viene applicato):

1. selettori di tipo: selettori semplici, possono venire raggruppati in liste separati da virgole. (li { color: green } , tutti i li avranno il colore verde)

2. selettori contestuali: specificano attributi di stile basati sul contesto (li em { color: gree } , tutti gli em dentro i li avranno il colore verde).Possono venire raggruppati in liste separati da virgole. Hanno la precedenza sui selettori semplici.

10.4 Ereditarietà

La maggior parte (ma non tutte) le proprietà di stile vengono trasmesse o ereditate dai livelli più alti nella gerarchia dei tag XHTML a quelli più bassi.
Gli stili applicati a elementi specifici del documento hanno la meglio sugli stili applicati a livello più alto.

10.5 Class

Class è un selettore, identifica un insieme di elementi come parte di un gruppo concettuale. Gli elementi di una classe possono essere modificati con una singola regola di stile. Per specificare lo stile è necessario aggiungere il nome della classe al selettore di tipo, separato da . oppure omettere il nome della classe e scrivere solo il selettore di tipo, preceduto da '.'
È possibile associare più classi ad un elemento html, separando i valori con uno spazio.

10.6 Id

Un id viene utilizzato per identificare un elemento univoco nel file XHTML.
Per specificare lo stile è necessario usare #id . È possibile utilizzare classi e id per definire il contesto di applicazione di una regola.

10.7 Pseudo-selettori

Vengono interpretati dal browser in base al contesto. Sono indicati dal carattere : .
In CSS1 ci sono gli pseudo-elementi, sono first-line e first-letter e permettono di applicare stili alla prima riga o lettera di un elemento XHTML.
Possono essere combinati con l'attributo class.
CSS2 specifica 4 pseudo-classi che possono essere applicate al tag $< a >$: link, visited, active, hover.

10.8 Aggiungere stili ad un documento

In 3 modi:

- Inline Styles: utilizzati soprattutto per gestire le eccezioni.
- Embedded Style Sheets: blocco con informazioni di stile all'interno del tag style di head. Possiamo specificare il media (screen ,tty, print, all, ...).
- Foglio di stile esterno collegato (con funzione import o con tag link).

10.9 La cascata

Tutte le regole di stile fluiscono in un unico foglio di stile virtuale, dove possono generarsi dei conflitti. Per risolvere questi conflitti, le regole CSS hanno delle priorità.

1. Raccogli tutte le regole che riguardano un determinato elemento HTML.
2. In caso di conflitto, stabilisci la priorità in base all'origine delle regole (in ordine, dalla meno prioritaria alla più prioritaria: settaggi del browser, foglio di stile dell'utente, fogli di stile esterni con link, fogli di stile esterni con import, fogli di stile embeded, inline style, attributi html). Usando l'indicatore !important dopo il valore della proprietà, possiamo aumentare al massimo la priorità di quella regola
3. In caso di ulteriore conflitto, si considera la specificità del selettore CSS, considerando come più importanti il riferimento ad id e il loro numero, il riferimento ad attributi e classi e il loro numero, il riferimento ad elementi html e il loro numero.
4. In caso di ulteriore conflitto, considera come più importante la regola che è stata scritta per ultima.

11 CSS - Parte 2

I fogli di stile considerano ogni elemento della pagina come se fosse contenuto all'interno di una scatola.
Il contenuto vero e proprio ha una determinata larghezza (width), è circondato da un margine interno (padding), da un bordo (border) e da un margine esterno (margin).
Qualsiasi sfondo applicato ad un elemento si estende nello spazio del padding e sotto il bordo.

11.1 Proprietà CSS dei contenitori

1. margin-top, margin-right, margin-bottom-margin-left: specifica le dimensioni del margine in maniera individuale nei quattro lati.

2. margin: forma abbreviata, specifichiamo tutte le dimensioni dei margini in una sola regola.
3. padding-top, padding-right, padding-bottom, padding-left: specifica la dimensione del padding.
4. padding
5. border-top-width, border-right-width, border-bottom-width, border-left-width
6. border-width
7. border-color
8. border-style: specificare lo stile del bordo.
9. border-top, border-right, border-bottom, border-left: specifica dimensione, colore, stile del bordo in una sola regola (una per ogni lato)
10. border
11. width: larghezza dell'elemento
12. height: altezza dell'elemento.

11.2 Come funziona il box model

Si assegnano in maniera addittiva valori al contenuto, al margine interno, al bordo e al margine eterno. La larghezza totale del box sarà quindi la somma di tutto.
Attenzione, perchè vecchie versione di IE (fino a 5.5) interpretavano in modo diverso il box model: larghezza e altezza del contenuto venivano riferite al box intero, margini esterni compresi.
Per i box senza margini e bordi non abbiamo problemi, per gli altri possiamo usare diversi stratagemmi:

11.2.1 Div nidificati

È possiible usare 2 div nidificati, uno per il box interno e uno per quello esterno. Per avere una grandezza corretta, dichiariamo indirettamente la larghezza complessiva del boxx interno attraverso la dichiarazione della larghezza del contenuto del box esterno. Per il box interno definiamo solo le caratterstiche del bordo interno e del padding.
Il limite è he vengono generati elementi di markup non strutturali.

11.2.2 Regola di Tantek

Usando un errore di interpretazione di IE 5, possiamo fornire una falsa dimensione e sovrascriverla in seguito con la dimensione reale.

11.2.3 Escape Hack

Si utilizza in questo caso una diversa interpretazione del carattere di escape da parte di IE 5.
La maggior parte dei browser ignora questa carattere, mentre su IE questo carattere rende illegibile il carattere seguente.

11.2.4 Commenti condizionati

Sono una forma particolare di commenti, visti come veri commenti HTML da parte di quasi tutti i browser ad eccezione di IE (dalla ver. 5 alla 9). Possiamo usare i commenti condizionati per definire sezioni specifiche di HTML leggibili solo da IE.

11.3 Modello di formattazione visuale

Ogni elemento dell'albero del documento genera zero o più box, secondo il modello di box definito da CSS. La disposizione di questi box dipende da diversi fattori:

- dimensioni e tipo di box
- schema di posizionamento
- relazioni tra gli elementi nell'albero del documento
- informazioni esterne

11.4 Tipi di box

Esistono due tipi di box: block box (generato da elementi come p, div o table: box come un rettangolo), o inline box(generati da span, img o da testo non struttura, ha una struttura visuale meno caratterizzata e può essere renderizzato su più righe). È possibile modificare la tipologia di default atraverso la proprietà display:

1. display:block: l'elemento genera un block box, anche se di default era inline.
2. display:inline: l'elemento genera un inline box, anche se di default era un block.
3. display:none: non associa un blocco all'elemento (quindi in teoria non mostra il blocco).
4. display:inline block: elemento posizionato come un inline ma contenuto formattato come block.
5. display:list-item: elemento genera un block box e un elemento di lista preceduto da un marcatore.

6. display: table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell, table-caption: l'elemento si comporta come una tabella o una sua componente.

11.5 Schemi di posizionamento

Un box può essere disposto secondo tre schemi di posizionamento: flusso normale (formattazione a blocco, inline, posizionamente relativo e posizionamento di box run-in), float (box disposto secondo il flusso normale, poi estrato e spostato il più possibile a destra o a sinistra), assoluto (box disposto secondo il flusso normale, senza impatto agli elementi fratelli che seguono, ad esso viene assegnata una posizione in riferimento ad un blocco contenitore).

11.5.1 Proprietà position e float

Queste due proprietà ci dicono quale algoritmo di CSS deve essere utilizzato per calcolare la posizione di un box.
position:

- static: box disposto secondo il flusso normale
- relative: box disposto secondo il flusso normale, poi il box viene spostato: se un box B viene spostato in maniera relativa, la posizione dei box sucessivi viene calcolata come se il box B non fosse stato spostato.
- absolute: la posizione e la dimensione vengono specificati attraverso le proprietà top, right, bottom e left. I box posizionati in maniera assoluta vengono estratti dal flusso normale. I margini dei box posizionati in maniera assoluta non collassano con altri margini.
- fixed: posizione calcolata come in absolute, ma viene fissata rispetto a qualche riferimento (ad esempio viene stampata su ogni pagina, non si muove se si scrolla)
- inherit: lo stesso valore dell'elemento genitore.

float: right, left, none, inherit.

11.5.2 Proprietà top, right, bottom, left

Un elemento viene definito come posizionato se il valore della proprietà position è diverso da static. I box posizionati vengono disposti secondo 4 proprietà:

- top length | percentage | auto | inherit: per posizionamenti relativi, l'offset è calcolato rispetto al limite superiore esterno del box stesso nel flusso normale. Per posizionamenti assoluti, si considera un offset dal limite superiore esterno del padding dell'elemento contenitore.
- right length | percentage | auto | inherit

- bottom length | percentage | auto | inherit
- left length | percentage | auto | inherit

Se viene dichiarato un valore positivo per la proprietà top,il box viene spostato verso l'alto, lo stesso valore per bottom. Per right verso sinistra, per left verso destra. Le 4 proprietà hanno il seguente significato:

- length: distanza fissa dal limite di riferimento (può essere negativa).
- percentage: percentuale della larghezza o dell'altezza del box contenitore (può essere negativa).
- auto: dipende.
- inherit: dipende dal genitore.

11.5.3 Flusso normale

È lo schema di default. Si applica a qualsiasi elemento non floated per cui position sia static o relative.
I block box scorrono verticalmente dalla sommità del blocco contenitore, ognuno disposto sotto al precedente.
I margini verticali di box consecutivi vengono collassati (si considera il maggiore).
Gli inline box scorrono orizzontalmente da sinistra verso destra. È consentita il ritorno a capo su una nuova linea quando si eccede la grandezza disponibile.

11.5.4 Posizionamento relativo

relative consente di spostare il box rispetto alla posizione determinata dal flusso normale.
Il box riposizionato può sovrapporre altri box.
Lo spostamento viene determinato attraverso una combinazione di valori delle proprietà top, right, bottom, left (se si specificano left e right, uno dei due viene ignorato, stesso discorso per top e bottom).
Ogni valore viene interpretato come la distanza alla quale il limite esterno del box corrispondente deve essere spostato rispetto alla sua posizione originaria nel flusso.
Per gli elementi diversi dall'elemento root, il riferimento per il loro posizionamento è quello del content edge del box contenitore generato dall'antenato più prossimo.
Per box che si sovrappongono, non è chiara la priorità di visualizzazione. Un browser, generalmente, mostra un box posizionato relativamente davanti agli elementi fratelli che lo precedono nel codice e dietro a quelli che lo seguono.
Se l'elemento A, posizionato relativamente, è un block, questo stabilisce un nuovo blocco contenitore e un nuovo sistema di coordinate sul quale si baseranno i discendenti.
Viene creata una nuova posizione per l'inizio del flusso normale, che verrà utilizzata dagli elementi nidificati.

11.5.5 Floats

Si ottiene attribuendo alla proprietà float di un elemento il valore di left o right. Il box viene posizionato vericalmente come se fosse nel flusso normale, ma viene spostato orizzontalmente quanto più possibile a destra o a sinistra nei limiti del box contenitore. I contenuti inline che lo circondano possono fluire sul lato opposto. Un box floated ha associata una grandezza (in assenza, riempirà orizzontalmente il box contenitore), vengono sempre considerati di tipo block anche se sono inline.
I margini verticali di un box floated non collassano con i margini di box sopra o sotto di esso.
Un box floated può sovrapporsi a block box adiacenti ad essi che si trovano nel flusso normale (usando la proprietà clear, il margine esterno specificato viene incrementato per permettere al suo bordo e al suo contenuto di posizionarsi sotto il margine del floated box).
Quando due o più elementi adiacenti vengono floated, il loro limite superiore viene posizionato sulla stessa linea se c'è spazio orizzontale sufficente. Se non c'è, gli elementi successivi vengono spostati in basso dove c'è spazio.

11.5.6 Absolute

Per position settata ad aboslute o fixed.
Il box viene rimosso dal flusso e non condiziona e non viene condizionato da altri elementi del flusso.
Gli elementi absolute vengono tratti come block box.
La posizione viene determinata dai valori di offset attributi a top, right, bottom, left. Gli spostamenti vengono calcolati in base al box contenitore (antenato più prossimo che ha position pari a relative, absolute o fixed. Se non c'è viene usato il blocco contenitore iniziale), e non in base al flusso normale.
Se l'elemento contenitore è di tipo block, gli offset vanno misurati dal limite esterno del padding, e non del contenuto.

11.5.7 Fixed

Fixed è un caso speciale dell'absolute. Un elemento fixed non si muove quando la pagina web viene navigata con lo scrolling, e nella stampa viene stampato su ogni pagina.

11.6 Stacking Order

Gi elementi posizionati in maniera assoluta possono sovrapporsi. Lo stacking context e la proprietà z-index determinano quale elemento dovrà apparire di fronte ad un altro.
Un elemento in posizone assoluta crea uno stacking context locale: consideriamo il valore di z-index associata ai diversi elementi del contesto, valori elevati hanno una precedenza nella visualizzazione. Se due elementi hanno lo stesso valore, l'ordine è quello del codice sorgente (elemento scritto prima sta dietro). È

possibile applicare valori negativi di z-index.
Quando si sovrappongono elementi che appartengono a stacking context diversi, questi vengono mostrati davanti o dietro a seconda del livello di stack del loro elemento contenitore definito da z-index.

12 CSS - Parte 3

12.1 Layout tabellari

12.1.1 Layout ibridi (tabelle + CSS)

Vantaggi: miglior supporto da parte di browser di vecchia generazione. Maggior facilità nel definire le relazioni tra i diversi elementi che costituiscono il layout (ogni elemento è associato ad una cella di tabella, è possibile anche avere tabelle nidificate). Parziale integrazione con CSS.
Svantaggi: l'uso delle strutture tabellari per costruire layout è solo tollerato dal W3C. Difficoltà di produrre versione specifiche di layout per smartphone o palmari a causa della rigidità della struttura tabellare.

12.1.2 Layout CSS

Vantaggi: la costruzione dei layout CSS basati su marcatori strutturali (come div) permette una migliore separazione tra contenuto e presentazione. Esiste una vasta gamma di relazioni spaziali definibili per la presentazione dei diversi elementi strutturali. Maggiore libertà nel produrre presentazioni specifiche per palmari o altri formati.
Svantaggi: buon supporto per le regole più complesse solo per browser di ultima generazione. Difficoltà maggiore nella fase di authoring nel definire le relazioni tra i diversi elementi che costituiscono il layout.
Riassumendo, a partire dalla definizione di un insieme di elementi strutturali nel file XHTML è possibile definire nelle regole di stile associate una vasta gamma di relazioni spaziali tra questi elementi. Gli elementi strutturali possono essere:

- definite in piena libertà nelle loro caratteristiche dimensionali
- affiancati orizzontalmente o verticalmente
- nascosti (parzialmente o totalmente)
- sovrapposti
- presentati secondo una sequenzialità diversa rispetto a quella definita nel codice XHTML.

12.2 Metodologie di progettazione per layout CSS

1. Definizione nel file XHTML di una sequenza di elementi strutturali. La sequenza deve essere definita in modo tale che la semantica dei contenuti

infomatici della pagina venga correttamente comunicata all'utente nel caso in cui venga utilizzato un convertitore TTS.

2. Definizione nel file XHTML di una sequenza di elementi strutturali compatibili con il layout già definito. La sequenza risultante deve essere tale che la semantica dei contenuti informativi della pagina venga correttamente comunicata all'utente nel caso in cui venga utilizzato un convertitore TTS.

13 Responsive Web Design

13.1 Motivazioni

I siti responsive rispondono all'esigenza di avere una presentazione flessibile in corrispondenza della tipologia di dispositivo utilizzata dall'utente. Queste nuove esigenze hanno evoluto le specifiche dei linguaggi per il web e dei dispositivi in grado di interpretarle.
Oggi, la mancanza di un sito responsive è un problema.

13.2 Come viene utilizzato il responsive web design

Si può usare HTML, CSS, JavaScript. Esistono anche dei framework, che semplificano l'uso dei linguaggi e che incorporano tecniche di responsive web design. Anche i moderni CMS permettono di presentare i contenuti in modo appropriato per i diversi dispositivi attraverso l'uso di temi specifici che utilizzano tecniche di responsive web design.

13.3 La piramide di Layon

Layon ha definito con una piramide gli aspetti basiliari di un'esperienza per gli utenti del web mobile, che comprende alla base l'accesso all'informazione e alla sommità gli aspetti più evoluti dell'esperienza, permessi dall'utilizzo di tecniche legate ad HTML5.

13.3.1 Accesso

La base della piramide. Gli utenti necessitano di avere accesso all'esperienza. È necessario garantire agli utenti mobile gli stessi contenuti degli utenti desktop (content-parity) e garantire accesso ad un'esperienza completa.

13.3.2 Interazione

Il visitatore necessita di interagire con il contenuto e spostarsi attraverso l'informazione. Ad esempio, per dispositivi mobile è posibile usare un menù fly-out o un drawer, oppure una dashboard.
Gli strumenti per la navigazione devono essre equilibrati, nè troppo eccessivi nè troppo scarsi e difficili da raggiungere.

È possibile anche usare tecniche di conditional loading, per fornire accesso a tutto il contenuto del sito senza che questo venga fornito tutto allo stesso momento.

13.3.3 Performance

È necessario tenere in considerazione le performance di caricamento delle pagine web. Pagine troppo pesanti non vanno bene in mobilità, devono caricarsi velocemente sennò l'utente abbandona la navigazione.

13.3.4 Enhance

È possibile rendere unica la navigazione da mobile fornendo funzioni che i desktop non hanno, migliorando l'esperienza.
Ad esempio, è possibile inserire un bottone per fare una chiamata, usare la localizzazione...

13.4 Griglia tipografica flessibile

È possibile usare un layout scalabile, specificando la dimensione come percentuale della misura dell'eleento contenitore. Il design complessivo rimarrà invariato al variare delle dimensioni della finestra.
È opportuno usare la stessa logica anche per il margine e il padding.

13.5 Reset Stylesheet

Sono collezioni di regole CSS associate alla pagine HTML, vengono introdotte per avere un punto di partenza comune per la presentazione per tutti i browser, rispetto al quale specificare poi le regole di stile.
Se non si specifica un reset stylesheet, in assenza di specifiche i browser presentano le informazioni secondo un formato di default, diverso da ogni browser.

13.5.1 Caratteri tipografici flessibili

È possibile stabilire la regola di reset per i caratteri. Da questa regola è possibile poi, usando em, definire le grandezze dei font nelle varie situazioni.

13.5.2 Strumenti di navigazione e layout

È necessario fornire widget diversi a seconda della grandezza dello schermo a disposizione (drawer a scomparsa per smartphone/tablet, menu orizzontale per desktop).

13.6 Immagini flessibili

L'inserimento di un immagine di grandezza fissa in un layout flessibile è un problema.

È possibile stabilire dei vincoli, ad esempio max-width nel css, per stabilire una grandezza massima per l'immagine. Se l'immagine è più piccola del box contenitore, sarà renderizzata secondo le sue dimensioni, se è più grande, verrà riscalata in modo da non superare i limiti del contenitore.
La suddetta regola è possibile applicarla anche ad altri elementi a larghezza fissa (video, rich media).
Il nostro caro IE fino alla versione 6 non supporta la proprietà max-width (REALLY?). Per risolvere i problemi di compatibili o si usa JavaScript oppure è possibile usare i commenti condizionati.
Per evitare problemi di visualizzazione dovuto a riscalatura dell'immagine è possibile utilizzare AlphaImageLoader, un filtro CSS che permette di incrementare drasticamente la qualità del rendering.

13.6.1 Immagini di background flessibili

Usare immagini di sfondo può essere utile per distinguere il fondo di due colonne di testo. Viene utilizzato un background sull'elemento che contiene entrambe le colonne, simulando in questo modo l'esistenza di due colonne colorate. Con div, infatti non si riesce ad ottenere un background di eguale altezza.
Per risolvere problemi di dimensioni dell'immagine (questa soluzione di basava su layout a dimensione fissa), è possibile creare una immagine più estesa della dimensione degli schermi finora utilizzati nella quale vengono definite proporzionalmente le due partizione delle colonne, usando un tool di manipolazione di immagini raster.
L'immagine viene poi posizionata come sfondo dell'oggetto contenitore delle due colonne, utilizzando una regola CSS nella quale viene indicata la sorgente, la ripetizione verticale e la posizione in percentuale.

13.6.2 Alternativa per il background

È possibile utilizzare la regola CSS overflow:auto o overflow:hidden applicata al contenitore delle colonne.
Questa regola permette di ottenere due colonne con campiture colorate differenziate.
In pratica viene assegnato un colore di sfondo al contenitore delle due colonne e alla colonna con il contenuto più lungo. L'altra colonna non viene colorata.

13.6.3 Overflow delle immagini

Si può usare il clipping (visualizzazione di una sola parte dell'immagine) per mettere vincoli sulla grandezza dell'immagine.

13.7 Media queries

La specifica CSS ha definito nella versione 2 i media type, dando la possibilità di assegnare un set di regole specifiche ad un media specifico: braille, embossed,

handheld, print, projection, screen, speech, tty, tv.
È possibile avere tanti CSS, uno per ogni stile, oppure definendo specifiche sezione all'interno di uno stesso foglio di stile. Purtroppo, per i dispositivi handheld, questa soluzione si è rilevata troppo grossolana (dispositivi sono troppi e troppo differenti).
CS3 ha permess0, grazie alle media queries, di rendere meno grossolana la suddivisione tra le classi diverse di dispositivi, consentendo di identificare un insieme di proprietà che caratterizzano il browser utilizzato dall'utente.

13.7.1 Alcune definizioni

- superfice di rendering: area dello schermo del dispsitivo.
- display area: spazio utile per la presentazione dei contenuti nella pagina web. Per i media continui, questo spazio corrisponde alla viewport (area attraverso la quale l'utente può visualizzare il contenuto renderizzato sulla canvas). Per i media non continui (paged media), ossia che suddividono il contenuto in pagine, corrisponde al page box.

13.7.2 Breakpoints

Possiamo utilizzare le media queries per descrivere regole diverse di presentazione. Possiamo definire dei breakpoint rispetto ai quali poi definire regole di presentazione che forniscono il risultato migliore nelle diverse situazioni.

breakpoint	descrizione
320 pixels	For small screen devices, like phones, held in portrait mode.
480 pixels	For small screen devices, like phones, held in landscape mode.
600 pixels	Smaller tablets, like the Amazon Kindle (6007800) and Barnes & Noble Nook (60071024), held in portrait mode.
768 pixels	Ten-inch tablets like the iPad (76871024) held in portrait mode.
1024 pixels	Tablets like the iPad (10247768) held in landscape mode, as well as certain laptop, netbook, and desktop displays.
1200 pixels	For widescreen displays, primarily laptop and desktop browsers.

13.7.3 Mobile first

Applicando la filosofia mobile first avremo un pieno supporto ai disposivi mobili. Con questa filosofia vengono prima descritte le regole di stile per una presentazione lineare, poi vengono redatte sezioni specifiche utilizzando la proprietà min-width descrivendo via via che cosa succede a mano a mano che la finestra si allarga. Oggi le media queries hanno un buon supporto da parte dei browser, ma quella vecchia volpe di IE, fino alle versione 8 compresa, non le supportano. È possibile usare librerie JavaScript per far supportare le media queries ai browser.

13.8 Relazione tra viewport e dispositivo

La dimensione dela viewport viene definita dall'user agent (il browser).
Non è detto che la dimnesione in pixel della viewport corrisponda alla dimensione in pixel hardware o pixel CSS del dispositivo che mostrerà la pagina web. È possibile dichiarare esplicitamente il valore in pixel della viewport, tramite un apposito tag meta.
Si può inoltre indicare, al posto dei pixel, un valore di zoom iniziale, e specificando come larghezza la larghezza del dispositivo. In questo modo, la viewport si adatterà a diversi dispositivi e anche all'orientamento del dispositivo.

13.9 Non ci sono più i pixel di una volta

Un pixel CSS è ancorata al pixel di riferimento, rispetto al quale va definito il rapporto con i pixel hardware.
Il pixel di riferimento è quello che viene visto da un occhio che osserva un pixel fisico di densità pari a 26 dpi ad una distanza di un braccio (28 inches). Il pixel di riferimento viene dunque definito attraverso un angolo.
Nella definizione di reference pixel (pixel di riferimento) vengono considerati sia la densità della matrice dei pixel HW sia la distanza a cui questa matrice viene vista dall'utente.
La scelta di questi parametri deriva dal fatto che molti dei contenuti web sono stati finora progettati per schermi desktop (utilizzati a distanza di circa un braccio) con una densità di 96 dpi.
Se, quindi, la densità della matrice fisica del dispositivo e la distanza di utilizzo dallo sguardo dell'utente corrispondono a quanto specificato prima, il mapping tra pixel CSS e pixel harware sarà un rapporto 1:1. Altrimenti, il costruttore di dispositivi dovrà tenere in considerazione il rapporto tra le 2 grandezze per utilizzarlo nel mapping tra specifica CSS e rendering della pagina. Deve, quindi:

- calcolare la densità equivalente, basandosi sulla conoscenza della densità di riferimento (96 dpi), della distanza di riferimento (28 pollici) e della distanza d'uso del nuovo dispositivo.
- densità equivalente = (distanza di riferimento / distanza d'uso nuovo dispositivo)*densità di riferimento.

- si calcolerà poi la devicepixelratio, ossia il rapporto tra pixel HW e pixel CSS, che è uguale a densità nuovo dispositivo effettiva / densità equivalente.

Un dip (device-independent pixel), introdotto per Android, corrisponde alla definizione del pixel CSS (1 dip corrisponde ad un pixel fisico ad una densità di 160 dpi).
Possiamo dire che la definizione di un pixel CSS si sovrappone a quella di un dip, o meglio, il numero di dips è uguale al numero di pixel CSS ottimale per visualizzare un sito web al 100% di zoom.

13.10 Immagini High-DPI

È necessario garantire la rappresentazione delle immagini su dispositivi a densità bassa.
Non sempre è possibile utilizzare formati vettoriali, ma ci sono delle tecniche per mostrare le immagini raster con la qualità migliore permessa dal dispositivo.

13.10.1 Tecniche che operano sulla singola immagine

Queste tecniche sacrificano le performance, dal momento che dovranno essere scaricata un'immagine High-DPI anche per dispositvi con DPI basso.Una tecnica può essere l'utilizzo di immagini fortemente compresse, per ridurre lo spazio. Di recente sono nati nuovi formati di immagine, ad esempio il Webp, che comprime in maniera molto efficente. Oppure il formato Flif. Un altro metodo è quello di usare immagini di tipo progressive, insieme a tecniche che fermano il caricamento dell'immagine quando si raggiunge una qualità discreta.

13.10.2 Tecniche che operano su immagine multiple

Hanno in comune un overhead nella creazione di versioni diverse della stessa immagine. Le opzioni possibili sono l'utilizzo di JavaScript, per determinare il devicepixelratio e altre feature del dispositivo.
Oppure la selezione server side che implica il riconoscimento dello user-agent e la scrittura handler per ogni immagini. Non è possibile ricavare dal lato server tutte le features del server.
O utilizzando le media queries.
O utilizzo regole CSS e nuovi attributi HTML (tramite l'attributo srcset).

13.10.3 Raccomandazioni

Le raccomandazioni principali sono:

- CSS regola image-set con fallback per immagini di sfondo
- srcset con fallback per immagini del contenuto

- immagini 2X fortemente compresse per le situazioni in cui si è disposti a sacrificare la qualità ma si vuole evitare l'overhead di creare e gestire versioni multiple di una stessa immagine.

13.11 Conditional Loading

Queste tecniche permettono di creare in maniera selettiva sulla pagina parti di contenuto a seconda delle caratteristiche del dispositivo e delle features del browser, utilizzando principalmente JavaScript.
Quando viene caricata la pagina, viene verificata la larghezza del dispositivo.
Se la grandezza è maggiore di X, allora presentiamo più dati, altrimenti possiamo mettere un pulsante ad esempio "Visualizza altro".

13.12 Lazy loading

Il lazy loading, invece, anzichè omettere il caricamento di parti del contenuto, lo ritarda, fino al momento in cui serve (ad esempio quando l'utente finisce di scrollare la pagina, carichiamo altri dati).
In questo modo possiamo risparmiare dati, scaricando solo gli elementi che effettivamente servono.

13.13 Responsive Tables

Sebbene abbiamo visto che i layout non devono basarsi su tabelle, a volte queste devono essere utilizzate, ad esempio per visualizzare i dati. È difficile adattare una tabella di dati in modo responsive al dispositivo.
Possiamo, ad esempio, usare regole CSS per linearizzare le strutture tabellari.
Alcuni framework come JQueryMobile permettono di ottenere facilmente una tabella responsive, utilizzando tecniche di linearizzazione che di visualizzazione selettiva delle colonne.
Bootstrap3 invece permette di scorrere orizzontalmente le colonne che compongono la tabella.

14 HTML 5

Con CSS, JavaScript e HTML è possibile costruire molte cose. Tuttavia, ci sono alcuni buchi, riguardandti le palette delle possibilità degli standard web. HTML5 ci viene in auto, evitando così di utilizzare plugin proprietari.

14.1 Audio

HTML5 permette di incorporare, con il tag audio, un documento audio in un documento HTML.
È possibile, mediante l'attributo booleano autoplay, far avviare automaticamente l'audio.

Usando l'attributo controls, gli utenti possono stoppare, mettere in pausa e gestire la sorgente audio.
Con JavaScript e la Audio API è possibile creare i propri controlli.
Con l'attributo autobuffer possiamo richiedere al borwser di precaricare l'audio in background. Ora questo attributo booleano è stato sostituito da preload, che può assumere none, auto e metadata.
Il problema dell'audio non è comunque del tutto risolto, perchè molti formati sono proprietari e per decodificarli è necessario pagare.
Usando un container audio, possiamo specificare più formati audio, per garantire maggiore compatibilità con i dispositivi e i browser. Possiamo anche, per aumentarne l'accessibilità, fornire una descrizione testuale del contenuto audio.

14.2 Video

HTML5 definisce il tag video, con gli attributi autoplay, loop, preload, per poter caricare e visualizzare un libro. È necessario specificare le dimensioni del video. Anchhe in questo caso, possiamo avere problemi di formati proprietari.

14.3 Canvas

Il canvas permette di definire un ambiente per la creazione di immagini dinamiche o anche applicazioni complesse.

All'interno del canvas è presente codice JavaScript. L'elemento canvas, purtroppo, non ha un DOM: il contenuto rappresentato all'interno del canvas non può essere manipolato, ad esempio da tecnologie assistive (problema di accessibilità). Si potrebbe però usare i canvas per riciclare contenuto esistente piuttosto che crearlo, utilizzando tecniche come ad esempio l'obnotrusive JavaScript.

15 HTLM5 - Parte 2

15.1 Semantica

HTML non fornisce un numero elevato di elementi con i quali operare. Possiamo definire paragrafi, liste, ma non possiamo definire, ad esempio, eventi, notizie, ricette di cucina.
Altri linguaggi di markup permettono di definire nuovi elementi, ma purtroppo è necessario avere un parser per il quale questo nuovo elemento abbia un significato.
Con l'attributo class, possiamo etichettare specifiche istanze di un elemento con rappresentanti di una classe o tipo speciale.
Questo tipo di elementi vengono chiamati microformati.

15.2 Microformati

Utilizzano un set di convenzioni, utilizzano l'attributo class per connotare semanticamente alcune parti del codice html: hCard per i contatti, hCalendar per gli eventi, hAtom per le news.
Dato che c'è il consenso di una comunità sui valori assegnati alle classi, sono stati creati parser ed estensioni del browser che lavorano con questi microformati.
Un microformato è una parte di markup che consente espressioni semantiche in una pagina HTML, sfruttando gli attributi class, rel, rev.
I programmi, tramite i microformati, possono estrarre i dati semantici da una pagina.
Permettono di creare codice X/HTML leggibile dai programmi ma continuando a garantire un'elevata comprensibilità da parte delle persone.

15.3 Modelli di contenuto

HTML5 fornisce un nuovo metodo per risolvere questo problema, definendo un nuovo set di tags.
Gli elementi, ora, non saranno più di tipo inline e block, ma avremo: text-level semantics, grouping content, forms, sectioning content.

15.3.1 Text-Level Semantics

Sono i vecchi inline, con l'aggiunta di, ad esempio, mark (mostra che il contenuto è in questo momento interessante), time (per date, orari, combinazioni di entrambi), meter (per denotare misure, con un minimo e un massimo), progress (per marcare qualcosa che sta mutando).

15.3.2 Sectioning content

- section: utilizzato per raggruppare insieme contenuti correlati telematicamente. Si differenzia da div, perchè viene utilizzato esplicitamente per raggruppare contenuti correlati.

- header: definito come contenitore per un gruppo di elementi introduttivi o navigazionali. Un documento può avere più header.
  È possibile usare header dentro un section.
  Un header appare in testa ad un documento o ad una sezione, ma non necessariamente. È definito dal contesto, piuttosto che dalla sua posizione.

- footer: descrive informazioni relative ad un contenuto definito all'interno di un contenitore comune, ad esempio chi ha scritto questo contenuto, informazioni di copyright, link a contenuto correlati...
  È possibile avere più elementi footer all'interno di un documento.

- aside: identifica il concetto editoriale di supplemento o nota integrativa (sidebar).
  Dovrebbe essere utilizzato per contenuto collegato tangenzialmente al contenuto principale.
  È un buon contenitore per descrivere un frammento di contenuto che si considera separato dal contenuto principale. Le citazioni costituiscono un buon esempio di contenuto collegato tangenzialmente.

- nav: contiene informazione navigazionale, ad esempio una collezione di links. Viene utilizzato per la navigazione principale. Molto spesso, un nav appare all'interno di un header.

- article: sembra che header, footer, nav e aside siano forme specializzate dell'elemento section.
  Una sezione è un frammento di contenuto correlato generico, mentre header, nav, aside, footer sono frammenti correlati di tipi specifici.
  L'article viene utilizzato per descrivere contenuto correlato specifico autoconclusivo (ha una sua completezza dal punto di vista semantico).
  È utile per post di blogs, forum, news e commenti (come hAtom).
  Se si usa un time all'interno di article, è possibile mediante un attributo dichiararlo data di pubblicazione (pubdate).

Articolo e sezione sono molto simili, la distinzione è l'aggettivo autocontenuto.
È possibile avere momlti articoli all'interno di una sezione oppure molteplici sezioni all'interno di un articolo.
È possibile nidificare sezioni all'interno di sezioni e articoli all'interno di articoli.

15.4 Perchè usare il markup semantico?

Per 5 semplici ragioni:

- Regioni definite chiaramente per lo sviluppatore: è molto più semplice identificare dove le sezioni iniscono e finiscono.

- Agganci extra per CSS senza utilizzare class o id: possiamo ad esempio definire proprietà di posizionamento per tutte le sections, o stabilire in modo semplice regole di visibilità.

- Regioni definite chiaramente per i motori di ricerca: quando HTML5 diventerà molto usato, probabilmente i search engines useranno i tag HTML5 semantici per indicizzare le pagine.

- Tecnologie assistive: possono migliorare l'accessibilità dei lettori di schermo.

- Agganci per lo scripting: ad esempio, uno script può utilizzare la funzione getElementsByTagname() per trovare tutti gli elementi di aside e nasconderli dalla vista dell'utente.

15.5 WAI-ARIA

WAI-ARIA (Accessible Rich Internet Application Suite) rappresenta un modo alternativo per aggiungere semantica al documento web.
Permette di rendere il contenuto e le applicazioni web più accessibili, soprattutto per quanto riguarda il contenuto dinamico e i controlli di interfaccia avanzati che utilizzano Ajax, JavaScript e le tecnologie collegate.
Le estensioni critiche di WAI-ARIA sono così riassunte:
Abbbiamo nuovi attributi per definire proprietà e stati dinamici, che possono essere mappati ad API di accessibilità per assicurare l'interoperabilità con le tecnologie assistite.
Abbiamo un attributo di ruolo per annotare elementi HTML con informazioni riguardanti lo scopo dell'elemento.
Valori di ruolo per i landmark del documento.

15.5.1 I ruoli di WAI-ARIA

WAI-ARIA definisce un insieme di ruoli per definire la struttura della pagina e widget specifici.
Oltre a questo, alcune tecnologie assistive forniscono modalità di interazione speciali per regioni marcate con il ruolo application o document.
I ruoli sono classificati come segue; ruoli astratti, ruoli widget, ruoli relativi alla struttura del documento, ruoli landmark.

- Abstract role: definiscono concetti generali, sono utilizzati nell'ambito della tassonomia dei ruoli di WAI-ARIA. Gli autori non devono utilizzare i ruoli astratti nel contenuto. Esempi di abstract role: command, composite, input, landmark, range, roletype, section, sectionhead, select, structure, widget.

- Widget role: elementi distinti dell'interfaccia o parte di elementi compositi: alert, alertdialog, checkbox, dialog, gridcell, link, log, marquee, menuitem, treeitem, progresbar, radio... I ruoli seguenti sono widget compositi (agiscono come contenitori che gestiscono altri ruoli): combobox, grid, menu, radiogroup, menubar, listbox, tablist, tree, treegrid.

- Document structure role descrivono strutture che organizzano il contenuto della pagina, non sono interattive: article, columnheader, definition, directory, document, group, heading, img, list, listitem...

- Landmark role: descrivono regioni della pagina, sono intesi come landmark navigazionali: banner, complementary, contentinfo, form, main, navigation, search.

15.5.2 Denotare le regioni di un documento

- HTML4: vengono denotate usando attributo class e id (senza estensione del linguaggio)

- HTML4 +WAI-ARIA: denotate attraverso l'uso di nuovi attributi (role) e di un set di valori (banner, article, navigation...), richiede un estensione del linguaggio
- HTML5: denotate attraverso l'uso di un nuovo set di tag: header, nav, article, footer..
- HTML5 + WAI-ARIA: denotate con il nuovo set di tag introdotti da HTML5, usando anche gli attributi role.

16 Usabilità

- La facilità con la quale un utente può imparare ad operare, a predisporre l'input e a interpretare l'output di un sistema o di una componente. (IEEE, 1990)
- L'efficacia, l'efficenza e la soddisfazione con la quale determinati utenti ragigungono scopi specifici in un determinato ambiente d'uso (ISO 9241-11 1998)

  Efficacia: accuratezza e completezza con la quale determinati utenti raggiungono scopi specifici in determinanti ambienti.

  Efficienza: rapporto tra le risorse impiegate e l'accuratezza e la completezza degli scopi raggiunti.

  Soddisfazione: comfort e accettabilità del sistema rispetto agli utenti e ad altri soggetti condizionati dal sistema stesso.
- È la misura della qualità dell'esperienza utente che interagisce con qualcosa (sito web, applicazione) (Nielsen)

16.1 Usabilità di Nielsen

Nielsen definisce queste caratteristiche dell'usabilità:

1. Facilità di apprendimento: con che velocità l'utente impara?
2. Efficienza d'uso: quanto velocemente un utente può svolgere i suoi compiti?
3. Memorizzabilità: l'utente può ricordare in maniera sufficente le modalità d'uso per utilizzare il sistema in modo efficace?
4. Frequenza e gravità degli errori: con che frequenza gli utenti fanno errori utilizzando il sistema? E quanto sono gravi gli errori?
5. Soddisfazione soggettiva: quanto piace all'utente utilizzare il sistema?

16.2 Ciclo di vita del software

È una mappatura accettata del processo di sviluppo del software. L'usabilità non è una caratterista da aggiungere alla fine del ciclo di sviluppo di un software.
Il ciclo di vita proposto dalla disciplina dell'Ingegneria dell'Usabilità si sviluppa in parallelo con le fasi tradizionali di sviluppo software ed è strettamente interconnesso con esse.
L'ingegneria dell'usabilità è un approccio metodico per raggiungere l'usabilità durante lo sviluppo di un software. coinvolge molti metodi da applicare nelle diverse fasi dello sviluppo di un software. Questi metodi coinvolgono la fase di definizione dei requisiti, lo sviluppo e il test di prototipi, la valutazione di design alternativi, la proposta di soluzioni e il test del sito, o dell'interfaccia, con gli utenti.

16.3 Usability Engineering Life Cycle (Nielsen)

1. Analisi degli utenti
   - caratteristiche individuali
   - attività correnti e desiderate
   - analisi funzionale
   - analisi evolutiva
2. Analisi competitiva
3. Definizione obiettivi usabilità
4. Design parallelo
5. Design partecipatorio
6. Coordinamento dell'interfaccia complessiva
7. Linee guida e valutazione euristica
8. Prototipazione
9. Valutazione interfaccia
10. Design iterativo
11. Studio e valutazione sistema

Possiamo suddividere tutte queste attività in due macro-aree:

- Attività di pianificazione e progetto
- Attività di valutazione

16.3.1 Attività di pianificazione e progetto

Un'attività importante di questa area è la progettazione di prototipi.
Il prototipo riduce la complessità del sistema, eliminando alcune parti: i prototipi orizzontali riducono il livello di funzionalità del sistema, e danno luogo ad interfacce con funzionalità non completamente implementate. I prototipi verticali, invece, riducono il numero di features, ma quelle scelte sono pienamente implementate. Gli scenari sono una riduzione estrema del sistema: simulano l'interfaccia solo se l'utente segue un percorso precedentemente pianificato.
I prototipi possono anche venire costruiti velocemente, ad esempio attribuendo meno enfasi all'efficienza dell'implementazione, scrivendo codice meno accettabile, usando algoritmi semplificati, usando una descrizione verbale dell'interfaccia, usando dati falsi ecc.

16.3.2 Attività di valutazione dell'usabilità

- Test: utenti rappresentativi operano sul sistema, o sul prototipo, svolgendo i loro compiti. Dei valutatori registrano quello che l'utente fa o dice mentre svolge il compito. I test vengono condotti con uno o due utenti alla volta. I valutatori utilizzano i risultati del test per verificare come l'interfaccia supporta gli utente nello svolgimento dei loro compiti.
- Tecniche di ispezione: specialisti di usabilità esaminano gli aspetti di usabilità relativi ad una determinata interfaccia utente.
- Raccolta dati: i valutatori dell'usabilità ottengono informazioni intervistando gli utenti o chiedendo loro di rispondere ad un insieme di domande in forma scritta.

16.4 Euristiche di usabilità

Sono un insieme di principi di usabilità applicati ad una grande varietà di interfacce, sia GUI che CLI.
Le euristiche possono venire usate sia come supporto alla progettazione sia come guida di usabilità di un'interfaccia.
Nel secondo caso, parliamo di valutazione euristica; è una tecnica a basso costo, facile e veloce. In questo caso, un gruppo di valutatori studia l'interfaccia e verifica che le euristiche siano state soddisfatte. Nielsen ha proposto queste 10 euristiche:

1. Visibilità dello stato del sistema: il sistema deve tenere aggiornati gli utenti su ciò che sta accadendo, inn un tempo ragionevole e con un feedback appropriato.
2. Corrispondeza tra sistema e mondo reale: il sistema deve parlare il linguaggio dell'utente, parole semplici capibili da tutti e non termini tecnici. Inoltre deve seguire le convenzioni del mondo reale, facendo apparire l'informazione secondo un ordine logico e naturale.

3. Controllo utente e libertà: supportate funzioni di Undo e Redo.

4. Consistenza e standard: seguire le convenzioni delle piattaforma su cui si sta lavorando.

5. Prevenzione degli errori: es. tastiera predittiva

6. Riconoscimento anzichè ricordo; oggetti, azioni e opzioni visibili e identificabili.

7. Flessibilità ed efficienza d'uso: usare shortcuts, acceleratori. Permettere all'utente di personalizzare lo svolgimento delle azioni frequenti.

8. Estetica e design minimalisti: le finestre di dialogo e l'estetica dovrebbero contenere solo l'essenziale, niente informazioni irrilevanti.

9. Aiuto per l'utente per il riconoscimento, la diagnosi e il recupero da situazioni di errore: i messaggi d'errore dovrebbero essere espressi in linguaggio semplice. È necessario indicare il problema con precisione e suggerire costruttivamente una soluzione.

10. Aiuto e documentazione: queste informazioni devono essere facili da cercare

16.5 Usabilità per il web

È importante, un sito web poco usabile porta a perdite di tempo e denaro per l'utente.
In Internet, gli utenti hanno esperienza dell'usabilità di un sito prima dell'interagire con esso e prima che essi abbiano speso soldi in possibili acquisti.
Le euristiche per il web sono le seguenti:

1. Navigazione (da euristica 1)
   Requisiti: localizzazione corrente mostrata chiaramente, collegamento alla pagina iniziale mostrato in modo chiaro, sezioni principali accessibili dalla pagina principale, fornita una mappa del sito per siti grandi e se necessarui una facile funzione di ricerca.

2. Funzionalità (da euristica 7)
   Requisiti: funzioni etichettate chiaramente, funzionalità essenziali disponibili senza abbandonare il sito, plug-in utilizzati solo se aggiungono valore (attenzione ai problemi di accessibilità), sito deve soddisfare sia gli esperti che i neofiti.

3. Controllo utente (da euristica 3)
   Requisiti: sito riflette workflow dell'utente, utente può annullare qualsiasi operazione, presente un punto di uscita ad ogni pagina, dimensioni pagina inferiori a 50kbps, tutti i browser supportati.

4. Linguaggio e contenuto (da euristica 2)
   Requisiti: precendenza a informazione e servizi importanti. No inclusa informazione non rilevante o poco utilizzata. Le informazioni o i servizi correlati sono raggruppati (sulla stessa pagina pagina o menu o nella stessa area all'interno della pagina). Linnguaggio semplice, paragrafi brevi, link concisi, espressivi e visibili, spiegazione dei termini utilizzati.

5. Help Online e guide per l'utente (da eurisitca 10)
   Requisiti: minimizzare richieste di aiuto e di istruzioni. Facilmente accessibili aiuto e istruzioni.

6. Feedback del Sistema e dell'Utente (da eurisitca 1)
   Requisiti: è chiaro quello che sta succedendo, utenti possono ricevere ed inviare un feedback via mail o (solo per invio) attraverso un modulo online, schermata di conferma modulo, feedback fornito in tempo utile. Ogni pagina deve presentare la data di ultima modifica, gli utenti devono essere informati della necessità di una particolare versione del browser o pugin.

7. Consistenza (da eurisitca 4)
   Requisiti: stessa parola o frase usata in modo consistente, menu riflettono titolo e intestazione pagina, titolo pagina significativo e riflette intestazione.

8. Prevenzione e Correzione degli Errori (da eurisitiche 5-9)
   Requisiti: utenti possono fare affidamento sul riconoscimento per utilizzare con successo il sito, sito accetta una varietà di azioni dall'utente, sito fornisce istruzioni concise per le azioni (anche indicando i formati di input). I messaggi d'errore visibili e scritti in un linguaggio chiaro, descrivono le azioni per risolvere un problema, forniscono un chiaro punto d'uscita, forniscono riferimenti per chiedere assistenza.

9. Architettura e Chiarezza Visuale(da euristica 8)
   Requisiti: sito organizzato secondo la prospettiva dell'utente, facilmente scansibili. Il design e layout sono concisi, e ridondanti solo quando è questo è produttivo per l'utente. Lo spazio bianco è adeguato, evitate animazioni non necessarie, colori per link chiaramente idenificabili, grassetto e corsivo usati con moderazione.

17 Il protocollo eGLU

Con le linee guida per isiti web delle PA (Direttiva 26 Novembre 2009), si è inteso fornire alle pubbliche amministrazioni criteri guida per la razionalizzazione del sistema dei siti web, in termini di principi generali, gestione, sviluppo e aggiornamento di contenuti e servizi online.
Le linee guida si accompagnano con i Vademecum di approfondimento.

17.1 Scopo del vademecum

Lo scopo è approfondire questo tema.
I protocolli eGLU 2.1 E EGLU-M (test semplificati su dispositivi mobili) sono semplici da usare per la pratica dell'usabilità a basso costo messa a disposizione delle redazioni delle PA di tutti i livelli di governo. I destinari del protocollo eGLU sono tutti coloro che, con diversi ruoli, partecipano al processo di creazione, gestione e sviluppo dei siti e dei servizi online delle PA.

17.2 Il protocollo eGLU 2.1

L'obiettivo del protocollo è quello di compilare, alla fine del test, un report con l'indicazione de punti di criticità più rilevanti, individuati mediante l'osservazione empirica e giustificati dai dati riportati sui moduli (si possono trovare nella sezione Allegati). I test che vengono svolti sono test di valutazione semplificati, qualitativi, basati sull'osservazione del comportamento dei partecipanti durante la navigazione, finalizzati ad idetificare criticità sull'interfaccia.

17.3 Approfondimenti rispetto al protocollo di base

La seconda parte del vademecum riguarda tecniche di progettazione e valutazione. È possibile definire ulteriori percorsi di progettazione, ulteriori metodi di valutazione dell'usabilità (ad esempio, Usability Card, coinvolgimento nel panel del test di persone disabili).

17.4 Il protocollo eGLU-M

È un adattamente dell'eGLU pensato per supportare l'organizzazione di analisi esporative di interfacce per i dispositivi mobile. L'approccio, la metodologie e le fasi della procedura rimangono sostanzialmente invariati.

17.5 Glossario dell'usabilità

Contiene un elenco di termini che ruotano intorno al tema dell'usabilità web.

17.6 Obiettivi del protocollo

1. descrivere una procedura per incoraggiare il coinvolgimento diretto e l'osservazione di utenti nella valutazione dei siti, senza necessariamente far ricorso a risorse esterne.

2. favorire una maggiore attenzione da parte degli operatori sul tema dell'usabilità

Sebbene nata per i siti web, la procedura eGLU 2.1 qui descritta è indipendente dalla tecnologia e dal mezzzo. eGLU-M cositituisce un esempio specifico di questa applicazione declinata.
L'intera procedura può essere considerata un test minimo di usabilità, e può

essere svolta da non esperti.
Per svolgere test più complessi è necessario rivolgersi ad un esperto di usabilità.

17.7 Procedura di osservazione degli utenti

Il conduttore dell'osservazione stila dei compiti da sottoporre ad alcuni partecipanti.
Alcuni utenti vengono selezionati e invitati a partecipare.
Si chiede a ciascun utente di eseguire i task assegnati: durante l'osservazione non si pongoono domande dirette, ma si osservano le persone interagire con il sistema e le difficoltà che incontrano.
Al termine, si raccolgono informazione su gradimento e facilità mediante questionari.

17.7.1 Fasi della procedura

1. Preparazione: quanti utenti selezionare? Che tipologia di utenti? Quanti e quali task? Come prepariamo i moduli per la raccolta dati? Cosa fare prima dell'osservazione (test pilota)?

2. Esecuzione: è necessario preparare un ambiente idoneo, far integrare correttamente i participanti e condurre l'osservazione. Inoltre è necessario raccogliere i dati, tramite osservazioni e questionati da compilare alla fine del test. Alla fine bisogna congedare i partecipanti.

3. Analisi dei risultati: i dati di successo raccolti vanno inseriti in un modello, che servirà a calcolare il tasso di successo complessivo del task e a dare un dettaglio su quale abbia avuto il tasso di successo più alto.
   Va stilato un elenco dei problemi osservati, indicando per ogni problema il numero di partecipanti che lo ha incontrato.
   Oltre a ciò, va steso un report, contenente: numero partecipanti e task, descrizione di task e pagine di completamento (criteri di successo) di ognuno, tasso di successo del sito, il tasso di successo per task e per partecipante, il SUS o lo UMUX-LIKE, l'NPS, un elenco dei problemi riscontrati.
   È possibile e consigliato aggiungere: valutazione dei problemi per numero di partecipanti e gravità. suggerimenti per la risoluzione dei problemi, una connessione dei problemi riscontrati ai principi euristici violati dall'interfaccia.

17.7.2 Dali da raccogliere

È necessario che l'osservatore e il conduttore del test raccolga i seguenti dati: scheda personale anagrafica dei partecipanti (prima di iniziare), per ogni partecipante e per ogni task se il task è stato superato o meno (si suggerisce anche di compilare una scheda annotando eventuali difficoltà o successi e di usare un criterio dicotomico (si o no)), un questionario generale per ogni partecipante (da compilare al termine di tutti i task). Al termine del test, dopo aver

compilato i questonari, si può richiedere al partecipante di raccontare eventuali difficoltà e problemi incontrati, ed eventualmente chiedere chiarimenti sulle difficoltà riscontrate.

17.7.3 Percezione della facilità d'uso

Si può utilizzare almeno uno fra:

- SUS: System Usability Scale
- UMUX-LITE: Usability Metric for User Experience

Questi questionari servono per avere indicazioni sulla percezione di facilità d'uso da parte dei partecipanti.
Oltre a questi è possibili utilizzare l'NPS (Net Promoter Score), usando voti in una scala da 1 a 10.