1 Lezione del 07-10-24

1.1 CSS

I Cascading Style Sheets (CSS) sono uno standard per la descrizione della presentazione degli elementi HTML. In CSS, possiamo assegnare propietà come:

- Font;
- Colori;
- Dimensioni degli elementi;
- Posizioni degli elementi;
- Bordi;
- Immagini di sfondo;
- ...

In generale, il CSS è un linguaggio a sé stante. Si può aggiungere del CSS direttamente ad un elemento HTML attraverso l'attributo `style`, all'interno dell'intestazione `<head>`, o, più comunemente, in un file separato che contiene solo codice CSS.

1.1.1 Vantaggi del CSS

Ci sono diversi vantaggi all'uso del CSS:

- Il grado di controllo della formattazione in CSS è significativamente migliore di quello fornito da HTML;
- I siti web diventano più facili da mantenere quando tutta la formattazione è essere inserita all'interno di uno, o una piccola quantità, di file CSS;
- Si ottiene più facilmente uno stile grafico consistente fra pagine;
- I siti basati sul CSS sono più accessibili, in quanto l'HTML non deve preoccuparsi della presentazione grafica;
- Un sito creato con del CSS serializzato sarà anche più facile da scaricare in quanto ogni pagina conterrà meno codice (basta scaricare il CSS una volta sola);
- Col CSS si può adattare una pagina a più formati;

Detto questo, ci sono anche varie difficoltà associate al fatto che il CSS non è stato pensato come una lingua per il layout, ergo è complicato gestire elementi fluttuanti, posizioni relative, gestione delle altezze e dei margini, ecc...

1.1.2 Versioni del CSS

Il comitato W3C ha pubblicato la raccomandazione di CSS livello 1 nel 1996. Un'anno dopo è arrivata la raccomandazione di livello 2, nota semplicemente come CSS2. Una versione aggiornata della raccomandazione di livello 2, la CSS2.1, fu pubblicata dopo circa dieci anni di lavoro, nel 2011. Nel frattempo, la W3C stava lavorando anche ad una versione completamente diversa, la CSS3.

Alcune funzionalità della CSS3 sono entrate a far parte della raccomandazione W3C:

- Selettori;
- Namespace;
- Media query;
- Colori;
- Attributi di stile.

1.1.3 Sintassi del CSS

Un documento CSS consiste di una o più **regole** di stile. Una regola consiste in un selettore che identifica l'elemento o lgi elementi HTML interessati, seguita da una serie di dichiarazioni di coppie proprietà - valore. Ad esempio:

```
selector {
  property: value;
  property2: value;
}
```

Ogni lista di dichiarazioni si chiama anche **blocco** di dichiarazioni. Lo whitespace è ignorato, l'unica cosa importante sono i punti e virgola.

1.1.4 Commenti in CSS

Si possono inserire commenti nella forma classica: `/* Questo e' un commento */`.

1.2 Selettori

Quando si definiscono regole CSS, dobbiamo prima usare un selettore per indicare quali elementi verranno effettati. I selettori CSS possono riferirsi a:

- Elementi singoli;
- Elementi multipli;
- Elementi accoppiati in qualche modo;
- Elementi che sono posizionati in un modo specifico nella gerarchia.

1.2.1 Selettori di elemento

Usano il nome dell'elemento HTML. Esiste un selettore universale, indicato dal carattere asterisco.

1.2.2 Selettori raggruppati

Si possono raggruppare i selettori, usando le virgole:

```
p, div, aside {
  margin: 0;
  padding: 0;
}
```

Questo è un buon modo per ridurre le dimensioni dei file CSS (elementi con le solite proprietà possono essere ragguppati).

Spesso si usano i selettori raggruppati per annullare gli stili di default del browser, in modo da ridurre inconsistenze date dalla visualizzazione su software diversi.

1.2.3 Selettori di classe

Esiste un'attributo HTML, il `class`, che permette di specificare classi per elementi di tipo uguale o diverso. Tutti gli elementi di una solita classe possono essere selezionati in CSS, usando semplicemente un punto seguito dal nome della classe.

Poniamo di avere dell'HTML:

```
<h1 class="first">Primo</h1>
<p class="first">Sottotitolo</p>
...
```

Potremo stilizzare sia l'header che il paragrafo con il CSS:

```
.first {
  font-style: italic;
  color: red;
}
```

1.2.4 Selettori di ID

Un selettore di id permette di indicare un'elemento specifico, attraverso l'attributo `id`, usando un cancelletto seguito dall'id dell'elemento:

Poniamo di avere dell'HTML:

```
<h1 id="second">Primo</h1>
```

Potremo stilizzare l'header con il CSS:

```
#second {
  font-style: italic;
  color: red;
}
```

La differenza fra i selettori di classe e i selettori di id è che i secondi dovrebbero essere usati solo per riferirsi ad un **unico** elemento, mentre i primi vanno bene per un numero qualsiasi di elementi. Inoltre, come vedremo in seguito, la specificità dei selettori di id è maggiore di quella dei selettori di classe.

1.2.5 Selettori di attributo

Si possono selezionare elementi in base alla presenza o meno di determinati attributi all'interno di elementi, o di valori specifici dati ad attributi di determinati elementi. Per questo si usano le parentesi quadre:

```
[title] {
  cursor: help;
}
```

Questo codice significa che ogni elemento con l'attributo title mostrerà un cursore col simbolo del punto interrogativo (o qualsiasi sia lo stile definito dal sistema operativo per `help`).

Esistono, in verità, modi più sofisticati di selezionare su attributi:

- `[attribute]`: cerca elementi con un attributo specifico;
- `[attribute=value]`: cerca elementi con un attributo specifico impostato ad un certo valore;

- `attribute~=value1, value2`: cerca elementi con un attributo specifico impostato ad almeno uno dei valori definiti in una lista separata da virgole;
- `attribute^=value`: cerca elementi con un attributo specifico il cui valore inizia col valore specificato;
- `attribute*=value`: cerca elementi con un attributo specifico il cui valore contiene la stringa specificata;
- `attribute$=value`: cerca elementi con un attributo specifico il cui valore finisce col valore specificato.

1.2.6 Pseudo-selettori

Un selettore di pseudo-elemento è un modo di selezionare qualcosa che non esiste esplicitamente come un elemento nel documento HTML ma che rappresenta comunque un oggetto selezionabile. I selettori di pseudo-elemento cominciano con una coppia di due punti, anche se a volte va bene un singolo due punti.

Un selettore di pseudo-classe si applica ad un elemento HTML esistente, ma lo indirizza quando si trova in un certo stato, o in una certa relazione familiare.

Alcuni esempi sono:

- **Pseudo-classe:**
  - `a:link`: si applica a link non visitati;
  - `a:visited`: si applica a link visitati;
  - `:focus`: si applica all'elemento che attualmente ha focus;
  - `:hover` si applica all'elemento su cui sto facendo hover;
  - `:active`: si applica all'elemento attivo (e.g. un bottone cliccato);
  - `:checked`: si applica a elementi attivati, come i radio button;
  - `:first-child`: si applica al primo figlio di un'elemento.
- **Pseudo-elemento:**
  - `::first-letter`: si applica alla prima lettera di un elemento;
  - `::first-line`: si applica alla prima linea di un elemento;
  - `::before`: inserisce contenuti prima di un elemento;
  - `::after`: inserisce contenuti dopo un elemento.

1.2.7 Selettori contestuali

I selettori contestuali sono selettori di pseudo-classe che permettono di selezionare elementi basandosi sui suoi antenati, discendenti, o fratelli, ergo sulla base della loro relazione gerarchica con altri elementi nell'albero del documento.

Questi sono:

- **Discendente:** indica un elemento che è contenuto da qualche parte dentro un altro elemento. Ad esempio: `div p` seleziona un paragrafo dentro una div;
- **Figlio:** indica un'elemento specifico che è figlio diretto dell'elemento specificato. Ad esempio: `div>h2` seleziona un header (di peso 2) figlio diretto di una div;

- **Fratello adiacente:** indica un elemento che è il prossimo fratello (direttamente dopo) dell'elemento specificato. Ad esempio: `h3+p` seleziona il primo paragrafo dopo qualsiasi heading (di peso 3).
- **Fratello generale:** indica un qualsiasi elemento che condivide il genitore con l'elemento specificato. Ad esempio: `h3~p`: indica un qualsiasi paragrafo che condivide il genitore con un heading (di peso 3).

Un caso particolare è rappresentato dai selettori di tipo discendente, in quanto sono i più usati. Questi selezionano elementi contenuti da altri elementi, e si specificano usando il carattere spazio. Possono anche includere ulteriori informazioni sul tipo di discendenza dell'elemento, come ad esempio l'essere primi figli:

```
#main div p:first-child { ... }
```

1.3 Proprietà

Ogni dichiarazione in CSS deve contenere una proprietà (ce ne sono centinaia). I nomi delle proprietà sono definiti dallo standard.

Vediamone alcuni esempi:

- **Font:**
  - `font`;
  - `font-family`;
  - `font-size`;
  - `font-style`;
  - `font-weight`;
  - `@font-face`.
- **Testo:**
  - `letter-spacing`;
  - `line-height`;
  - `text-align`;
  - `text-decoration`;
  - `text-indent`.
- **Colori e sfondo:**
  - `background`;
  - `background-color`;
  - `background-image`;
  - `background-position`;
  - `background-repeat`;
  - `box-shadow`;
  - `color`;
  - `opacity`.

- **Bordi:**
  - `border`;
  - `border-color`;
  - `border-width`;
  - `border-style`;
  - `border-top`, `border-left`, `border-right`, `border-bottom`;
  - `border-image`;
  - `border-radius`.
- **Spaziature:**
  - `padding`;
  - `padding-bottom`, `padding-left`, `padding-right`, `padding-bottom`;
  - `margin`;
  - `margin-bottom`, `margin-left`, `margin-right`, `margin-bottom`;
- **Dimensioni:**
  - `height`;
  - `max-height`;
  - `min-height`;
  - `width`;
  - `max-width`;
  - `min-width`;
- **Layout:**
  - `bottom`, `left`, `right`, `top`;
  - `clear`;
  - `display`;
  - `float`;
  - `overflow`;
  - `position`;
  - `visibility`;
  - `z-index`.
- **Liste:**
  - `list-style`;
  - `list-style-image`;
  - `list-style-type`.
- **Effects:**
  - `animation`;
  - `filter`;
  - `perspective`;
  - `transform`;
  - `transition`.

1.4 Valori

Ogni dichiarazione deve contenere poi un valore per ogni properietà. Alcuni valori di proprietà vengono da una lista predefinita di parole chiave. Altri sono valori come lunghezze, percentuali, numeri puri, colori od URL.

Vediamoli nel dettaglio:

1.4.1 Colori

Esistono più modi di esprimere il colore in CSS:

- **Nome:** riferendosi per nome ai colori cercati, ad esempio `red`, `lightblue`, `hotpink` (solo CSS3), ecc...
- **RGB:** si puo indicare il colore in modalità RGB (Rosso, Verde e Blu) come `rgb(255,255,255)`. Inoltre, si può usare un quarto attributo *alpha*, che rappresenta la trasparenza del colore, in forma `rgb(255,255,255,0)`.
- **HSL:** si può indicare il colore anche in modalità HSL (Hue, Saturation, Lightness (sarebbe Value)) come `hsl(255,255,255)`. Anche qui è supportato l'attributo *alpha*.
- **Esadecimale:** infine, si possono indicare i colori in modalità RGB con caratteri esadecimali, ad esempio `#FFFFFF`.

1.4.2 Unità di misura

Si possono usare **unità relative**, cioè basate sul valore di qualcos'altro, come la dimensione di un genitore; e **unità assolute**, cioè basate su unità di misura reali.

Esempi di unità relative sono:

- `em`: uguali al valore della proprietà `font-size` dell'elemento su cui è usato. Quindi, quando si usa per le dimensioni dei font, l'unità `em` dipende dalle dimensioni del font del genitore;
- `%`: una misura relativa, in percentuale, alle dimensioni dell'elemento su cui viene usata;
- `ex`: simile all'em, ma si basa sull'altezza x del font;
- `ch`: ancora simile all'em, ma si basa sulla dimensione del carattere 0;
- `rem`: sta per *root em*, che è la dimensione di font dell'elemento radice; A differenza di `em`, che potrebbe variare fra elementi, `rem` è assoluto nella pagina;
- `vw`, `vh`: stanno per *viewport width* e *viewport height*, cioè le dimensioni della finestra dove viene visualizzata la pagina. Sono entrambe misure percentuali, e possono essere usate per creare elementi che scalano insieme alla pagina.

Ed esempi di unità assolute sono:

- `in`: pollici;
- `cm`: centimetri;
- `mm`: millimetri;

- `pt`: points, è la misura predefinita di `font-size`, e vale $\frac{1}{72}$ pollici;
- `pc`: pica, vale $\frac{1}{6}$ pollici;
- `px`: pixel, vale $\frac{1}{96}$ pollici.

1.5 Tipi di stylesheet

Uno stylesheet è un file CSS che viene applicato ad una pagina web. Esistono più tipi di stylesheet:

- **Stylesheet d'autore:** creati da zero per un certo sito o una certa pagina;
- **Stylesheet utente:** creati dagli utenti (o dagli sviluppatori dei browser), solitamente per motivi di accessibilità;
- **Stylesheet del browser:** forniti insieme al browser per visualizzare pagine senza CSS.

Inoltre, possiamo (come già visto) mettere il CSS in più posizioni all'interno del codice. Vediamole nel dettaglio:

- **Stili inline:** sono regole di stile inserite direttamente dentro un'elemento HTML attraverso l'attributo style:

```
<h2 style="font-size: 24pt">Questo e' un grande titolo</h2>
<h2 style="font-size: 24pt; font-weight: bold">Questo e' un
    grandissimo titolo</h2>
```

  Solitamente, conviene evitare di usare questo attributo, in quanto esistono modi migliori e più mantenibili per stilizzare gli elementi;

- **Stili embedded:** sono regole di stili incluse nell'elemento `style`, all'interno dell'head.

```
<head lang="en">
<title>La mia pagina</title>
  ...
<style>
h2 {
  font-size: 24pt;
  font-weight
}
</style>
</head>
```

  Rappresentano un'approccio migliore degli stili inline, ma sono comunque poco raccomandati in confronto all'opzione successiva;

- **Stili esterni:** sono file CSS esterni inclusi come riferimento nel documento HTML attraverso l'elemento `link`. Rappresentano l'opzione più mantenibile (lo stesso stylesheet può stilizzare più pagine, ergo aggiornare lo stylesheet significa aggiornare tutte le pagine), e più veloce (il browser deve fare una sola richiesta HTTP dello stylesheet per tutte le pagine del sito).

1.6 La cascata

CSS sta per *Cascading* Style Sheet. Cascading, in questo contesto, si riferisce al modo in cui si discriminano le regole da scegliere in caso di conflitti. Questo è necessario in quanto, come abbiamo appena visto, esistono più tipi di stylesheet che spesso si vanno a sovrapporre.

CSS usa i seguenti principi per la sua cascata:

- **Ereditarietà**;
- **Specificità**;
- **Posizione**.

Vediamoli nel dettaglio.

1.6.1 Ereditarietà

Molte (ma non tutte) le proprietà CSS si riferiscono non solo a loro stesse, ma anche ai loro discenedenti. Font, colori, properità di lista e testo sono ereditabili. Layout, dimensioni, bordi, sfondi e spaziature non lo sono di default, ma lo possono diventare attraverso la parola chiave `inherit` usata per specificare il valore della properità in un elemento discendente.

Ad esempio, potremo avere:

```
div {
  font-weight: bold;
  margin: 50px;
  border: 1pt solid green;
}
p {
  border: inherit;
  margin: inherit;
}
```

In questo caso, un `p` dentro un `div` avrà bordo e margini ereditati da quest'ultimo.

1.6.2 Specificità

La specificità determina quali regole hanno precedenza se applicabili allo stesso elemento. Più un selettore è specifico, più alta è la precedenza della regola che lo usa.

Questo è implementato con un sistema di **pesi**, dove diversi selettori hanno diversi pesi (ad esempio un selettore di id è più specifico di un selettore di classe, che a sua volta è più specifico di un selettore di elemento, ecc...) sulla specificità della regola.

La specificità viene calcolata come un numero concatenato $abcd$, dove a, b, c e d sono cifre. Le cifre si assegnano come segue:

- a: vale 1 se il CSS è inline a dell'HTML, 0 altrimenti;
- b: conta il numero di attributi id del selettore;
- c: conta il numero di altri attributi e pseudo-classi del selettore;
- d: conta il numero di elementi e pseudo-elementi del selettore.

1.6.3 Posizione

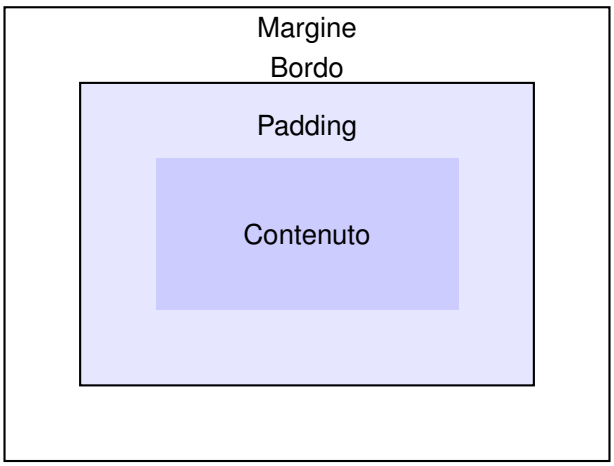
Esiste infine un'altro aspetto da considerare, cioè quello della posizione della regola. Il principio della posizione è che quando le regole hanno la stessa specificità, le ultime hanno più peso. Questo vale sia per le regole, sia per i blocchi di dichiarazioni.

Esiste un'eccezione a questa regola: il marchio `!important` fara in modo che la regola a cui è applicato sovrascriverà ogni altra regola definita dall'autore in altre posizioni.

1.7 Modello a scatola

Veniamo quindi a quello che è effettivamente il meccanismo secondo cui vengono disposti gli elementi sulla pagina. In HTML, tutti gli elementi esistono dentro una **scatola**. La scatola ha `width` e `height`, cioè larghezza e altezza, un `padding`, cioè dell'area in più *interna* alla scatola, un `margin`, cioè un'area di margine *esterna* alla scatola, è un bordo detto `border`.

Il contenuto dell'elemento stà all'interno della scatola, fino al padding. Lo sfondo del padding è il `background-color` o la `background-image` dell'elemento, mentre lo sfondo del margine è il `background-color` o la `background-image` del genitore dell'elemento.



1.7.1 Sfondi

Oggi gli sfondi degli elementi, o *background*, vengono realizzati con apposite proprietà CSS, e non con il tag `<img>`.

Vediamo quali sono queste proprietà:

- `background`: una proprietà combinata che permette di impostare più valori del background contestualmente, imposta comunque a valori di default gli attributi non specificati;
- `background-attachement`: specifica se l'immagine di sfondo scorre col documento (comportamento di default) o rimane fissa. I valori possibili sono quindi `fixed` o `scroll`;
- `background-color`: specifica il colore dello sfondo;
- `background-image`: specifica l'immagine di sfondo attraverso un URL;
- `background-position`: specifica dove l'immagine dovrebbe essere posta: ha possibili valori `top`, `bottom`, `center`, `left` e `right`. Si possono anche fornire posizioni in pixel o in percentuali, sempre per entrambe le dimensioni, che indicano la distanza dall'angolo in alto a sinistra dell'elemento;

- `background-repeat`: determina la ripetizione del background, che può essere `repeat`, `repeat-x`, `repeat-y` e `no-repeat`.
- `background-size`: specifica la dimensione dell'immagine di sfondo.

1.7.2 Bordi

I bordi possono vengono usati per separare gli elementi fra di loro. Hanno le seguenti proprietà:

- `border`: una proprietà combinata che permette di impostare più valori del bordo contestualmente, imposta comunque a valori di default gli attributi non specificati;
- `border-style`: specfica il tipo di bordo. Sono supportati, fra l'altro, `solid`, `dotted`, `dashed`, `double`, `groove`, `ridge`, `inset` e `outset`.
- `border-width`: specifica la larghezza del ordo in unità (ma non in percentuali). Sono supportate anche le parole chiave `thin`, `medium`, ecc...
- `border-color`: specifica il colore del bordo;
- `border-radius`: specifica il raggio delle smussature degli spigoli;
- `border-image`: specifica l'URL di un'immagine da usare come bordo.

Si possono indicare questi attributi su un solo lato del bordo usando notazioni come `border-top-color`. Esiste anche una nozione più veloce, che permette di impostare le properietà di ogni lato con una sola dichiarazione, assumendo l'ordinamento `top`, `right`, `bottom` e `left` per i lati. Quindi, ad esempio, si potrà avere:

```
border-color: red green orange blue;
/* equivale a: */
border-top-color: red;
border-right-color: green;
border-bottom-color: orange;
border-left-color: blue;
```

Infine, si ricorda che i bordi vengono allargati dal padding, ma non dai margini.

1.7.3 Margini e padding

I margini indicano la dimensione dello spazio lasciato **esternamente** alla scatola, mentre il padding indica la dimensione lasciata **internamente** alla scatola. Si nota che i margini possono **collassare**: quando abbiamo margini adiacenti verticali, viene usato il valore del margine più grande, e l'altro viene annullato. I margini orizzontali invece non collassano mai, ed esistono alcuni casi particolari quando nemmeno i verticali lo fanno (il piacere di scoprirli è lasciato al lettore).

1.7.4 Dimensioni

Avevamo le proprietà `width` e `height` per impostare le dimensioni. Queste non sono del tutto veritiere, in quanto la dimensione degli elementi dipende anche dai loro contenuti. Per gestire questa situazione, si hanno a disposizione la proprietà `overflow`, che indica come gestire la dimensione in eccesso:

- `visible`: lascia i contenuti in eccesso visibili:

- `hidden`: nasconde i contenuti in eccesso (si comporta quasi come una maschera);
- `scroll`: visualizza scrollbar per permettere di visualizzare tutti i contenuti nel fattore forma indicato da `width` e `height`;
- `auto`: lascia scegliere al browser (spesso creera una scroll verticale).

1.8 Stilizzazione del testo

Esistono due tipi di proprietà che influenzano il testo:

- **Proprietà dei font** che influenzano i font e il loro aspetto;
- **Proprietà di paragrafo** che influenzano il testo a discapito del font usato.

Bisogna notare che i font usati dal browser devono essere installati sul computer dell'utente: questo significa che i font installati sulla macchina dello sviluppatore potrebbero non trovarsi su quella degli utenti, e dare così problemi di compatibilità. Per questo motivo, solitamente si fornisce uno **web font stack**, cioè una lista di font alternativi da usare nel caso non si trovasse la scelta originale sul computer dell'utente.

Vediamo quindi tutte le proprietà dei font:

- `font`: una proprietà combinata che permette di impostare più valori del bordo contestualmente, imposta comunque a valori di default gli attributi non specificati;
- `font-family`: specifica quale font usare. Come già detto se ne possono (e se ne dovrebbero) specificare più di uno;
- `font-size`: specifica la dimensione del font in punti; `font-style`: specifica se il font è in corsivo, grassetto, ecc...
- `font-variant`: specifica la variante del font, ad esempio se è in stampatello minuscolo o no;
- `font-weight`: specifica il peso del font, con un valore da 100 a 900 in multipli di 100, o con una parola chiave fra `normal`, `bold`, `bolder` e `lighter`.

e del paragrafo:

- `letter-spacing`: specifica la spaziatura fra le lettere;
- `line-height`: specifica l'altezza di una linea di testo;
- `text-align`: specifica l'allineamento del testo;
- `text-decoration`: specifica le decorazioni (sottolineatura, sopralineatura, ecc...);
- `text-direction`: specifica la direzione del testo;
- `text-shadow`: imposta un'effetto di ombreggiatura, specificando gli offset orizzontali e verticali, la sfocatura e il colore dell'ombra.

1.8.1 Font-face

Il selettore `@font-face` permette di usare font non scaricati sulla macchina dell'utente, appoggiandosi a siti open source come Google Web Fonts.

1.8.2 Nota sulle unità di misura

Usare i pixel non è particolarmente sicuro: la densità dei pixel dei dispositivi moderni è variata in modo che un pixel CSS non corrisponde più tanto spesso a un pixel effettivo sul dispositivo. Se vogliamo creare layout che funzionino bene su più dispositivi, dobbiamo usare unità come gli `em` e le percentuali. Usare queste unità implica anche che l'utente potrà, a piacimento, cambiare la dimensione del testo. Ricordiamo poi l'unità `rem`, che si basa sull'origine del documento, ed è quindi sempre uguale in esso.

1.9 Tabelle HTML

Il linguaggio HTML ci permette di creare tabelle attraverso il tag `table`. Queste possono visualizzare una vasta gamma di contenuti, dalle immagini al testo, ai link e altre tabelle.

1.9.1 Struttura di una tabella

Una `table` contiene un certo numero di righe indicate da `tr`, e ogni riga contiene un certo numero di celle `td`. I contenuti vanno sempre dentro le celle. I titoli delle colonne si scrivono nei tag `th` anziché `td`. Ad esempio, potremmo avere:

```
<table>
  <tr>
    <th>Sintetizzatore</th>
    <th>Produttore</th>
    <th>Anno</th>
  </tr>
  <tr>
    <td>Roland TB-303</td>
    <td>Roland</td>
    <td>1981</td>
  </tr>
  <tr>
    <td>Yamaha DX7</td>
    <td>Yamaha</td>
    <td>1983</td>
  </tr>
  <tr>
    <td>ARP Odyssey</td>
    <td>ARP</td>
    <td>1972</td>
  </tr>
  <tr>
    <td>Sequential Prophet-5</td>
    <td>Sequential Circuits</td>
    <td>1978</td>
  </tr>
</table>
```

1.9.2 Celle non standard

Se vogliamo che una cella occupi più colonne, usiamo l'attributo `colspan`. Ad esempio:

```
<table>
  <tr>
    <th>Sintetizzatore</th>
    <th colspan=2>Anno e produttore</th>
  </tr>
  ...
```

```
</table>
```

Allo stesso modo, se vogliamo che una cella occupi più righe, usiamo `rowspan`. Ad esempio:

```
<table>
  <tr>
    <th>Produttore</th>
    <th>Sintetizzatore</th>
  </tr>
  <tr>
    <td rowspan=2>Roland</td>
    <td>MS-20</td>
  </tr>
  <tr>
    /* qui serve un'elemento solo! */
    <td>M1</td>
  </tr>
  <tr>
    <td>ARP Odyssey</td>
    <td>ARP</td>
    <td>1972</td>
  </tr>
  <tr>
    <td>Sequential Prophet-5</td>
    <td>Sequential Circuits</td>
    <td>1978</td>
  </tr>
</table>
```

1.9.3 Elementi addizionali

Esistono altri elementi da aggiungere alle tabelle:

- `caption`: indica un titolo per la tabella;
- `col` e `colgroup`: indicano gruppi di colonne;
- `thead`, `tbody` e`tfoot`: indicano una separazione semantica fra l'intestazione, il corpo e il footer di una tabella.