INTRO

I CSS servono per gestire tutto il layout di un sito Web. Grazie ad essi è possibile intervenire sulla formattazione del testo, sul posizionamento degli elementi grafici e sulla disposizione che questi elementi avranno rispetto a diversi media e device. Questa guida permette di comprendere e manipolare le nozioni più importanti per la creazione di layout e sfruttare caratteristiche universalmente condivise da tutti i browser.

L'acronimo CSS sta per Cascading Style Sheets (fogli di stile a cascata) e designa un linguaggio di stile per i documenti web. I CSS istruiscono un browser o un altro programma utente su come il documento debba essere presentato all'utente, per esempio definendone i font, i colori, le immagini di sfondo, il layout, il posizionamento delle colonne o di altri elementi sulla pagina, etc.

La storia dei CSS procede su binari paralleli rispetto a quelli di HTML, di cui vuole essere l'ideale complemento. Da sempre infatti, nelle intenzioni degli uomini del W3C, HTML dovrebbe essere visto semplicemente come un linguaggio **strutturale**, alieno da qualunque scopo attinente la **presentazione** di un documento.

Per questo obiettivo, ovvero arricchire l'aspetto visuale e la presentazione di una pagina, lo strumento designato sono appunto i CSS. L'ideale perseguito da anni si può sintetizzare con una nota espressione: separare il contenuto dalla presentazione.

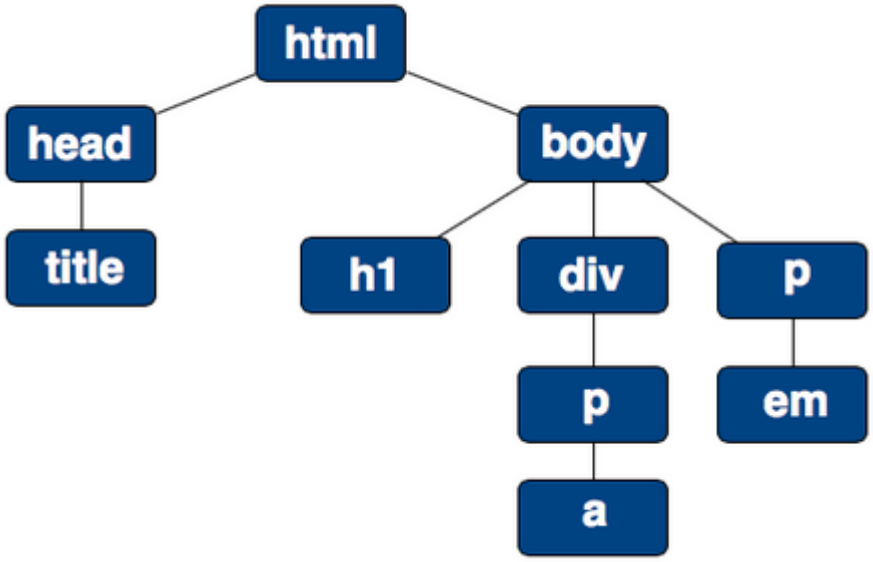
DOM

Un concetto fondamentale da assimilare per una corretta applicazione dei CSS è quello della struttura ad albero di un documento. Il meccanismo fondamentale dei CSS è infatti l'ereditarietà. Esso fa sì che molte proprietà impostate per un elemento siano automaticamente ereditate dai suoi discendenti. Sapersi districare nella struttura ad albero significa padroneggiare bene questo meccanismo e sfruttare al meglio la potenza del linguaggio. Tutti i concetti che spiegheremo qui di seguito sono definiti nel cosiddetto Document Object Model (DOM), lo standard fissato dal W3C per la rappresentazione dei documenti strutturati.

Es dato questo frammento di codice:

```
<html>
  <head>
    <title>Struttura del documento</title>
  </head>
  <body>
    <h1>Titolo</h1>
    <div>
      <p>Primo <a href="pagina.htm">paragrafo</a>.</p>
    </div>
    <p>Secondo <em>paragrafo</em>.</p>
  </body>
</html>
```

Ecco la sua rappresentazione strutturale:



Il documento è una perfetta forma di gerarchia ordinata in cui tutti gli elementi hanno tra di loro una relazione del tipo genitore-figlio (parent-child in inglese). Ogni elemento è genitore e/o figlio di un altro.

Un elemento si dice genitore (parent) quando contiene altri elementi. Si dice figlio (child) quando è racchiuso in un altro elemento. In base a queste semplici indicazioni possiamo analizzare il nostro documento.

Ad esempio, <body> è figlio di <html>, ma è anche genitore di <h1>, e <p>. Quest'ultimo è a sua volta genitore di un elemento .

Si potrebbe concludere che anche <body> sia in qualche modo genitore di . Non è esattamente così. Introduciamo ora un'altra distinzione, mutuata anch'essa

dal linguaggio degli alberi genealogici, quella tra antenato (ingl: ancestor) e discendente (ingl: descandant).

La relazione parent-child è valida solo se tra un elemento e l'altro si scende di un livello. Esattamente come in un albero familiare si indica la relazione tra padre e figlio. Pertanto possiamo dire che <head> è figlio di <html>, che <a> è figlio di <p>, etc. Tra e <a>, invece, si scende di due livelli: diciamo allora che è un **antenato** di <a> e che questo è rispetto al primo un discendente.