HTML (II parte) - Appunti

Concetti Avanzati HTML e DOM

Questo documento approfondisce aspetti avanzati di HTML e introduce il Document Object Model (DOM).

Struttura del Documento HTML e Tipi di Elementi

- **Elementi Inline:** Elementi che si trovano all'interno di blocchi di testo (es. `<a>`, `<b>`, `<i>`). Gli elementi `<a>` (àncore) definiscono i link ipertestuali. Non possono essere annidati.
    - Attributi principali di `<a>`:
        - `href`: Specifica l'URI di destinazione del link.
        - `name`: Definisce un nome per l'àncora, utilizzabile come destinazione di un link.
- **Elementi di Blocco e di Lista:** Definiscono blocchi strutturali nel documento.
- **Elementi Generici:** Elementi con scopi generali (es. `<div>`, `<span>`).
- **Elementi di Struttura:** Elementi che definiscono la struttura semantica del documento (es. `<header>`, `<footer>`, `<article>`).
- **Link e Immagini:**
    - **`<a>` (Ancore):** Creano collegamenti ipertestuali.
    - **`<img>` (Immagini):** Incorporano immagini inline. È un elemento vuoto (definito interamente dai suoi attributi).
        - Attributi principali di `<img>`:
            - `src` (obbligatorio): URL dell'immagine.
            - `alt`: Testo alternativo (per accessibilità e se l'immagine non viene caricata).
            - `name`: Nome dell'immagine.
            - `width`: Larghezza forzata (in pixel).
            - `height`: Altezza forzata (in pixel).
        - **Immagini Responsive (`srcset` e `sizes`):**
            - `srcset`: Specifica diverse versioni dell'immagine con dimensioni differenti.
            - `sizes`: Indica le dimensioni dello schermo a cui associare ciascuna versione dell'immagine in `srcset`.
            - `src`: Fallback per browser che non supportano `srcset`.
- **Embedding di contenuti multimediali**
    - `<video>`: Incorpora video
    - `<audio>`: Incorpora audio
    - `<object>`: Embedding generico di oggetti multimediali, tipicamente tramite plugin.
    - `<embed>`: Embedding generico senza plugin.
    - `<iframe>`: incorpora una pagina HTML all'interno della pagina corrente.
- **Tabelle:**
    - Definite riga per riga (`<tr>`).

    - Celle di intestazione (`<th>`) e celle di dati (`<td>`).
    - `<caption>` : Didascalia della tabella.
    - `<thead>` , `<tbody>` , `<tfoot>` : Sezioni della tabella (intestazione, corpo, piè di pagina).
    - `colspan` e `rowspan` : Attributi per unire celle su più colonne o righe.
    - **Attenzione all'accessibilità:** Evitare tabelle di layout; usare CSS per il layout.
- **Form:**
    - Permettono l'interazione dell'utente e l'invio di dati al server.
    - Legati ad applicazioni server-side.
    - Elementi principali:
        - `<form>` : Contenitore dei controlli del form.
            - `method` : Metodo HTTP (GET, POST).
            - `action` : URI dell'applicazione server-side.
        - `<input>` , `<select>` , `<textarea>` : Controlli del form (widget).
            - `name` : Nome del controllo (usato dal server).
            - `type` : Tipo di controllo (text, checkbox, radio, submit, etc.).
            - I controlli dello stesso gruppo (checkbox, radio) condividono lo stesso `name` .
        - `<button>` : Pulsante cliccabile (diverso da submit).
        - `<label>` : Etichetta visibile del controllo.
        - `<datalist>` : Fornisce un elenco di opzioni predefinite per gli `<input>` . Usato con l'attributo `list` dell'input.
- **HTML Living Standard (LS) e Form:**
    - Nuovi attributi per `<input>` :
        - `placeholder` : Testo di suggerimento.
        - `required` : Campo obbligatorio.
        - `readonly` : Campo non modificabile.
        - `list` : Associa un `<datalist>` per suggerimenti.
    - Nuovi tipi di `<input>` :
        - `email` , `url` , `number` , `range` , `date` , `search` , `color` .
        - I browser forniscono interfacce specifiche per questi tipi.

Approfondimenti Sintassi HTML

- **Attributi Globali:**
    - `id` : Identificatore unico.
    - `class` : Lista di classi (per semantica e CSS).
    - `style` : Stile CSS inline.
    - `title` : Testo aggiuntivo (per accessibilità).
    - Attributi `i18n` (internationalization): `lang` (lingua), `dir` (direzione del testo).
    - Attributi di interattività: `accesskey` , `autofocus` , `tabindex` , `inputmode` .
    - Attributi di evento: `onclick` , `ondblclick` , `onmouseover` , etc.
- **Attributi `data-*` :**
    - Attributi personalizzati per dati utilizzati da script.
    - Accessibili via CSS e Javascript (`element.dataset`).
- **Attributi ARIA (WAI-ARIA):**

    - Migliorano l'accessibilità delle pagine web.
    - `role`: Specifica il ruolo semantico dell'elemento.
    - `tabindex`: Rende l'elemento selezionabile con la tastiera.
    - `aria-*`: Attributi specifici per diverse funzionalità.
- **Entità HTML:**
    - Rappresentano caratteri speciali (es. `<` per `<`, `>` per `>`, `&` per `&`, `&quot;` per `"`, caratteri non-ASCII).
    - Entità numeriche (es. `&#224;` per `à`).
- **Tipi di Dati:**
    - **Colori:**
        - Codice RGB esadecimale (es. `#FF0000` per il rosso).
        - Nomi di colori predefiniti (es. `red`, `blue`, `green`).
        - *Nota*: HTML 4 deprecava l'uso esplicito dei colori, preferendo i fogli di stile.
    - **Lunghezze:**
        - Pixel (numeri assoluti).
        - Percentuali (relative al contenitore).
        - Multi-lunghezze (lista di valori, `*` per dividere lo spazio rimanente).
- **Peculiarità Sintattiche (HTML vs. XHTML):**
    - HTML non è case-sensitive (XHTML sì).
    - Whitespace collassato in un singolo spazio (tranne ` `).
    - Estensioni dei browser (elementi/attributi non standard).
    - **Bizzarrie sintattiche (permesse in HTML, rimosse in XHTML, reintrodotte in HTML LS):**
        - Virgolette opzionali per valori di attributi che sono TOKEN.
        - Omissione di tag `<HTML>`, `<BODY>`.
        - Omissione del tag di chiusura per `<p>`, `<li>`, `<option>`.
        - Attributi di solo valore (es. `selected` invece di `selected="selected"`).

Elemento `<head>`

- Contiene informazioni rilevanti per l'intero documento.
- Elementi principali:
    - `<title>`: Titolo del documento (per finestra, bookmark, motori di ricerca).
    - `<link>`: Collega documenti esterni (es. fogli di stile).
    - `<script>`: Codice Javascript (inline o esterno).
    - `type="module"` per importare moduli JS.
    - `defer` e `async` per il caricamento asincrono degli script.
    - `<style>`: Stili CSS inline.
    - `<meta>`: Meta-informazioni.
        - `charset`: Codifica caratteri.
        - `http-equiv`: Intestazioni HTTP simulate.
        - Meta-informazioni per motori di ricerca (keywords, description).
        - `viewport`: per la visualizzazione su dispositivi mobili (proposta Apple in standardizzazione)
    - `<base>`: URL di base per URL relativi.

`<object>`, `<canvas>`, Web Components

- `<object>`: Embedding di contenuti, esteso in HTML5.

- `<canvas>` : Area rettangolare per disegnare grafica 2D con Javascript.
  - API "HTML Canvas 2D Context" (Recommendation W3C).
- `<video>` , `<audio>` : Incorporano contenuti multimediali, con eventi e API per controllarli.
- Web Components: Componenti personalizzati HTML.
  - Custom Elements
  - Shadow DOM
  - HTML Templates

Document Object Model (DOM)

- Interfaccia di programmazione (API) per HTML e XML.
- Definisce la struttura logica dei documenti e come accedervi e manipolarli.
- Permette di:
  - Costruire documenti.
  - Navigare la struttura.
  - Aggiungere, modificare, cancellare elementi e contenuti.
- **Struttura del DOM:**
  - Rappresentazione ad albero del documento.
  - Nodi: elementi, attributi, testo, commenti, etc.
  - Relazioni genitore-figlio tra i nodi.
- **Classi Principali:**
  - `DOMNode` : Classe base per tutti i nodi.
  - `DOMDocument` : Rappresenta l'intero documento.
  - `DOMElement` : Rappresenta un elemento HTML.
  - `DOMAttr` : Rappresenta un attributo.
  - `DOMText` : Rappresenta un nodo di testo.
- **Metodi e Proprietà (Esempi):**
  - `DOMNode` : `nodeName` , `nodeType` , `childNodes` , `parentNode` , `appendChild` , `removeChild` , etc.
  - `DOMDocument` : `documentElement` , `createElement` , `getElementById` , `getElementsByTagName` , etc.
  - `DOMElement` : `getAttribute` , `setAttribute` , `removeAttribute` , etc.
- **Selettori in DOM:**
  - `getElementById`
  - `getElementsByName`
  - `getElementsByTagName`
  - `getElementsByClassName`
  - `querySelector` (selettori CSS, restituisce il *primo* elemento)
  - `querySelectorAll` (selettori CSS, restituisce *tutti* gli elementi)
- **Manipolazione del DOM (Esempio Javascript):**
  - Creazione di elementi (`createElement`).
  - Aggiunta di testo (`createTextNode`).

    - Aggiunta di nodi al DOM (`appendChild`).
    - Accesso ai nodi esistenti (es. `getElementById` , `childNodes`).
- `innerHTML` e `outerHTML` : proprietà che permettono di leggere/scrivere il contenuto (incluso/escluso il tag) di un elemento come stringa HTML.

Parsing di HTML 5 (WhatWG)

- Algoritmo standardizzato per il parsing di HTML (anche malformato).
- Focus sulla creazione di una struttura dati consistente (DOM).
- Non impedisce la creazione di pagine "non strict", ma incoraggia la buona pratica di usare il markup in modo corretto.

Questo riassunto copre i punti chiave del documento "12-HTML (II parte).txt", fornendo una panoramica teorica degli argomenti trattati, utile per la preparazione dell'esame.