

10 - Markup

Introduzione

- **Markup:** Mezzo per rendere esplicita l'interpretazione di un testo, migliorandone la fruibilità e specificando modalità d'uso in sistemi informatici.
- **Non solo informatico:** Spazi, virgole, punti, numerazione pagine, margini sono esempi di markup pre-informatico.

Tipi di Markup

- **Proprietario vs. Pubblico:**
 - **Proprietario:** Creato da un'azienda, diritti detenuti, modificabile a discrezione.
 - **Pubblico:** Creato da un gruppo di interesse, specifiche pubbliche, standard ufficiale (a volte).
- **Binario vs. Leggibile:**
 - **Binario:** Memorizzazione esatta delle strutture in memoria, non leggibile da umani.
 - **Leggibile:** Pensato per essere letto da umani in casi di emergenza, richiede parsing.
- **Interno vs. Esterno:**
 - **Interno:** Istruzioni di presentazione inserite nel testo, richiede sintassi speciali (segnalatori, escaping).
 - **Esterno:** Contenuto e markup separati, collegati da indirizione (indirizzi, offset, identificatori).
- **Puntuazionale:** Segni per informazioni sintattiche (punteggiatura).
- **Presentazionale:** Effetti grafici per la presentazione (paragrafi, interlinea, ecc.).
- **Procedurale:** Istruzioni per sistemi automatici (es. Wordstar Dot Commands).
- **Descrittivo:** Identifica il tipo di elemento (titolo, paragrafo, citazione).
- **Referenziale:** Riferimenti a entità esterne (es. sigle trasformate in parole intere).
- **Metamarkup:** Regole di interpretazione del markup (es. definizione di macro).

Esempi di Markup

- **Testo senza markup:** Testo continuo senza formattazione (es. papiro).
- **Markup metabolizzato:** Punteggiatura, spazi, ritorni a capo, maiuscole/minuscole.
- **Markup procedurale (es. RTF):** Comandi per il sistema di lettura (es. a capo, margini).
- **Markup descrittivo (es. XML):** Informazioni sul ruolo degli elementi (es. `<TITOLO>` , `<AUTORE>`).

Linguaggi di Markup

- **TROFF/NROFF:**
 - Nato nel 1973, parte di Unix.
 - Usato per documentazione tecnica (manuali Unix).
 - Comandi esterni (preceduti da ".") o interni (escape "").
 - Macro limitate.
- **TEX e LATEX:**
 - Creato da Donald Knuth.
 - Linguaggio di programmazione completo.
 - Complesso, per programmatori e tipografi esperti.
 - **Metafont:** Descrizione dei caratteri con formule matematiche.
 - **LaTeX (Leslie Lamport):** Raccolta di macro per tipi di documento comuni.
- **Markdown & sintassi wiki:**

- Formati testuali con trucchi per effetti tipografici.

- **Adatti a testi lunghi, limitati per effetti tipografici specifici. - Esempio: `` An h1 header**

Paragraphs are separated by a blank line.

2nd paragraph. *Italic*, **bold**, and `monospace`. Itemized lists look like:

- this one
- that one
- the other one

- **JSON (JavaScript Object Notation):**

- Formato dati derivato dalla notazione JS per gli oggetti.
 - Esempio:

```
{
  "nome": ["Giuseppe", "Andrea", "Federico"],
  "cognome": "Rossi",
  "altezza": 180
}
```

- **YAML (YAML Ain't a Markup Language):**

- Linearizzazione di strutture dati simile a JSON.
- Sintassi ispirata a Python.
- Superset di JSON.
- Indentazione per annidamento.
- Supporta tipi scalari, liste, hash.
- Commenti.
 - Esempio:

```
nome:
- Giuseppe
- Andrea
- Federico
cognome: Rossi
indirizzo:
  via:
    strada: "Via Indipendenza"
    numero: 15
```

- **SGML (Standard Generalized Markup Language):**

- Standard ISO 8879 (1986).

- Meta-linguaggio non proprietario di markup descrittivo.
- Leggibile, generico, strutturale, gerarchico.
- **Meta-linguaggio:** Linguaggio per definire linguaggi di markup.
- **Non proprietario:** Non dipende da un singolo produttore.
- **Markup leggibile:** Markup in testo semplice.
- **Markup descrittivo:** Non solo per la stampa.
- **Markup strutturato:** Definisce regole strutturali.
- **Markup gerarchico:** Strutture a livelli di dettaglio.
- **Documenti SGML:**
 - **Dichiarazione SGML:** Istruzioni di partenza (es. set di caratteri).
 - **Dichiarazione di documento (DOCTYPE):** Regole per la validazione.
 - **Istanza del documento:** Testo con markup.
- **Componenti del markup:**
 - **Elementi:** Parti di documento con senso proprio (tag iniziale, contenuto, tag finale).
 - **Attributi:** Informazioni aggiuntive sull'elemento (nome="valore").
 - **Entità:** Frammenti di documento memorizzati separatamente.
 - **Testo (#PCDATA):** Contenuto vero e proprio.
 - **Commenti:** Note ignorate dalle applicazioni.
 - **Processing Instructions (PI):** Indicazioni su come gestire il documento.
- **XML 1.0:**
 - Raccomandazione W3C (1998).
 - Sottoinsieme di SGML.
 - Grammatica formalizzata (Extended Backus-Naur Form).
 - **Documenti ben formati:**
 - Tag corrispondenti e annidati.
 - Elemento radice.
 - Elementi vuoti con simbolo speciale (<vuoto/>).
 - Attributi tra virgolette.
 - Entità definite.
 - **Documenti validi:** Presentano un DTD e possono essere validati.

Conclusioni

- Il markup è fondamentale per la strutturazione e la presentazione del testo.
- Diversi linguaggi di markup si sono evoluti nel tempo, ognuno con caratteristiche specifiche.
- SGML e XML sono meta-linguaggi potenti per la definizione di linguaggi di markup personalizzati.