Impostare una pagina con JS e HTML

Programmazione

HTML (HyperText Markup Language) è il linguaggio di base usato per creare la struttura e il contenuto di una pagina web.

Funziona attraverso tag che definiscono elementi come paragrafi, titoli, immagini e collegamenti.

Ogni pagina HTML parte con una struttura gerarchica che comprende il tag <html>, al cui interno troviamo <head> (informazioni e risorse della pagina) e <body> (contenuto visibile all'utente).

JavaScript è un linguaggio di programmazione che consente di aggiungere interattività e logica dinamica alle pagine web.

Mentre HTML gestisce la struttura e il contenuto, JavaScript può manipolare il DOM (Document Object Model), cioè la rappresentazione in memoria della pagina, permettendo di reagire a eventi come clic, input o caricamenti.

È possibile inserire il codice JavaScript direttamente nella pagina tramite il tag <script> oppure in file esterni .js collegati all'HTML.

Elementi Base di HTML

<!DOCTYPE html>

- Dichiara al browser che il documento è in HTML5. Deve sempre essere la prima riga della pagina.

<html>

- Contiene tutto il codice HTML della pagina. È l'elemento radice del documento.

<head>

- Include le informazioni non visibili della pagina, come il titolo, i meta tag e i collegamenti a file CSS o script JS.

Elementi Base di HTML

<title>

- **Imposta il titolo della pagina che appare nella scheda del browser.**

<body>

- **Contiene tutto ciò che è visibile all'utente: testi, immagini, pulsanti, link, ecc.**

<h1> – <h6>

- **Rappresentano i titoli e sottotitoli della pagina. <h1> è il più importante, <h6> il meno.**

Elementi Base di HTML

<p>

- **Definisce un paragrafo di testo. È uno degli elementi più usati.**

<a>

- **Crea un collegamento ipertestuale (link). Si usa con l’attributo href per indicare la destinazione.**
- **Esempio: Visita il sito**

<img>

- **Inserisce un’immagine nella pagina. Richiede l’attributo src per indicare il percorso dell’immagine e alt per il testo alternativo.**

Limiti di HTML

Non è un linguaggio di programmazione

- HTML è un linguaggio di markup, non di programmazione: non può eseguire logica, cicli, condizioni o calcoli. Serve solo a strutturare il contenuto, non a farlo “funzionare”.

Dipende da CSS per lo stile

- HTML da solo non gestisce la grafica o il design della pagina. Per definire colori, layout, spaziature o animazioni è necessario affiancarlo a CSS (Cascading Style Sheets).

Dipende da JavaScript per l’interattività

- Per rendere la pagina dinamica (clic, animazioni, modifiche in tempo reale) serve JavaScript. HTML non può reagire da solo alle azioni dell’utente.

Limiti di HTML

Non è un linguaggio di programmazione

- HTML è un linguaggio di markup, non di programmazione: non può eseguire logica, cicli, condizioni o calcoli. Serve solo a strutturare il contenuto, non a farlo “funzionare”.

Dipende da CSS per lo stile

- HTML da solo non gestisce la grafica o il design della pagina. Per definire colori, layout, spaziature o animazioni è necessario affiancarlo a CSS (Cascading Style Sheets).

Dipende da JavaScript per l’interattività

- Per rendere la pagina dinamica (clic, animazioni, modifiche in tempo reale) serve JavaScript. HTML non può reagire da solo alle azioni dell’utente.

Limiti di HTML

Scarsa adattabilità senza CSS responsivo

- Da solo, HTML non è in grado di adattarsi automaticamente ai diversi schermi (PC, tablet, smartphone). Serve una progettazione responsive tramite CSS o framework come Bootstrap.

Nessuna sicurezza integrata

- HTML non può proteggere dati, validare input o impedire attacchi. Questi aspetti devono essere gestiti da linguaggi di backend (come PHP, Python, C#, Java) o da JavaScript lato client.

Nessuna gestione dei dati o della logica applicativa

- HTML non può memorizzare dati, accedere a database o gestire file. È necessario un linguaggio lato server per queste operazioni (es. PHP o Node.js).

Spiegazione del codice:

- La sezione <head> contiene le informazioni di base della pagina.
- Nel <body> abbiamo un titolo (<h1>), un paragrafo (<p>) e un pulsante (<button>).
- Il codice JavaScript, inserito nel tag <script>, modifica il contenuto del paragrafo quando l'utente clicca sul pulsante.
- Il metodo document.getElementById() serve per selezionare un elemento HTML tramite il suo id, mentre textContent permette di cambiarne il testo.

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Esempio HTML + JS</title>
</head>
<body>
    <h1>Benvenuto nella mia pagina web</h1>
    <p id="testo">Questo testo cambierà quando clicchi il pulsante.</p>

    <button onclick="cambiaTesto()">Cliccami!</button>

    <script>
        // Funzione JavaScript che modifica il contenuto del paragrafo
        function cambiaTesto() {
            let paragrafo = document.getElementById("testo");
            paragrafo.textContent = "Hai cliccato il pulsante! Ora il testo è cambiato!";
        }
    </script>
</body>
</html>
```

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

Il linguaggio CSS (Cascading Style Sheets) serve per definire l'aspetto grafico di una pagina web: colori, font, layout e margini.

HTML consente di collegarlo in tre modi principali.

a) Foglio di stile esterno

È il metodo più usato e professionale. Si collega un file .css esterno tramite il tag <link> all'interno del <head>.

```
1. <head>
2.   <link rel="stylesheet" href="stile.css">
3. </head>
```

Spiegazione:

- **rel="stylesheet" indica che si tratta di un foglio di stile.**
- **href specifica il percorso del file CSS.**
- **Tutti gli stili si trovano in un file separato (es. stile.css), facilitando la manutenzione.**

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

b) Stile interno

Puoi scrivere il CSS direttamente nella pagina, dentro un blocco <style>.

```
<head>
  <style>body {
      background-color: lightblue;
      font-family: Arial;
    }
  </style>
</head>
```

Spiegazione:

- **È utile per pagine semplici o test rapidi.**
- **Non è consigliato per siti complessi, perché mischia struttura e stile.**

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

c) Stile inline

Lo stile è applicato direttamente a un singolo elemento tramite l'attributo style.

1. ** Testo in rosso **

Spiegazione:

- **È il metodo più immediato ma meno flessibile.**
- **Difficile da mantenere in progetti grandi.**

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

Collegamento a file JavaScript

JavaScript permette di rendere la pagina dinamica e interattiva.
Come per il CSS, ci sono diversi modi per collegarlo.

a) Script esterno

Il metodo più comune: collega un file .js esterno con <script>.

```
1. <body>
2.   <script src="script.js"></script>
3. </body>
```

Spiegazione:

- **src indica il percorso del file JavaScript.**
- **Posizionarlo prima della chiusura di </body> è una buona pratica per migliorare la velocità di caricamento della pagina.**

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

b) Script interno

Il codice JavaScript è scritto direttamente nella pagina, dentro un tag <script>.

```
<script>function saluta() {
    alert("Ciao dal tuo script interno!");
  }
</script>
```

Spiegazione:

- **Utile per test o esempi brevi.**
- **Non è indicato per progetti di grandi dimensioni, dove è preferibile separare il codice.**

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

c) Richiamo inline

Il JavaScript viene eseguito direttamente tramite un attributo HTML come onclick, onchange, ecc.

1. **<button onclick="alert('Hai cliccato!')">Cliccami</button>**

Spiegazione:

- **È un metodo semplice ma poco ordinato, perché mischia codice HTML e JavaScript nello stesso elemento.**

Elementi di collegamento verso JavaScript e CSS - Collegamento ai fogli di stile CSS

Tipo	Tag o metodo	Dove si scrive	Uso consigliato
CSS esterno	*<link rel="stylesheet" href="file.css">*	*<head>*	*Struttura professionale*
CSS interno	*<style>...</style>*	*<head>*	*Pagine piccole o test*
CSS inline	*style="..."*	*Dentro i tag HTML*	*Modifiche rapide*
JS esterno	*<script src="file.js"></script>*	*In fondo a <body>*	*Metodo consigliato*
JS interno	*<script>...</script>*	*<head> o <body>*	*Demo o test*
JS inline	*onclick="..."*	*Attributo HTML*	*Esempi semplici*

Spiegazione sintetica

- **<link rel="stylesheet" href="stile.css">**

Collega un file CSS esterno (es. stile.css) per gestire gli stili generali.

- **<style> ... </style>**

Contiene regole CSS scritte direttamente nella pagina.

- **style="..."**

Applica uno stile direttamente su un singolo elemento.

- **<script> ... </script>**

Contiene codice JavaScript interno alla pagina.

- **<script src="script.js"></script>**

Collega un file JavaScript esterno (es. script.js) per gestire funzioni aggiuntive.

- **onclick="..."**

Esegue un'azione JavaScript direttamente da un elemento HTML.

JS

```
// File JS esterno
console.log("Script esterno caricato correttamente.");
```

CSS

```
/* File CSS esterno */
h1 {
    font-size: 28px;
    color: navy;  }
```

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Esempio Completo HTML + CSS + JS</title>

    <!-- Collegamento a un foglio di stile esterno -->
    <link rel="stylesheet" href="stile.css">

    <!-- Stile interno -->
    <style>
        body {
            background-color: #f0f8ff;
            font-family: Arial, sans-serif;
            text-align: center;
        }

        h1 {
            color: darkblue;
        }

        .paragrafo {
            color: darkgreen;
            font-size: 18px;
        }

        button {
            background-color: lightblue;
            border: none;
            padding: 10px 15px;
            margin-top: 10px;
            cursor: pointer;
        }

        button:hover {
            background-color: skyblue;
        }
    </style>
</head>

<body>
    <!-- Titolo principale -->
    <h1 style="text-decoration: underline;">Pagina di esempio completa</h1>

    <!-- Paragrafo con classe (stile interno) -->
    <p class="paragrafo">Questo testo è gestito dal CSS interno.</p>

    <!-- Paragrafo con stile inline -->
    <p style="color: red; font-weight: bold;">Questo testo usa lo stile inline.</p>

    <!-- Pulsante con richiamo inline JavaScript -->
    <button onclick="alert('Hai cliccato il pulsante inline!')">Clicca qui</button>

    <!-- Paragrafo modificabile da JavaScript -->
    <p id="messaggio">Clicca il secondo pulsante per cambiare questo testo.</p>

    <!-- Pulsante che richiama funzione JS interna -->
    <button onclick="cambiaTesto()">Cambia il testo</button>

    <!-- Script interno -->
    <script>
        // Funzione JavaScript interna che modifica il contenuto di un paragrafo
        function cambiaTesto() {
            let p = document.getElementById("messaggio");
            p.textContent = "Il testo è stato cambiato dallo script interno!";
            p.style.color = "blue";
        }
    </script>

    <!-- Collegamento a uno script esterno -->
    <script src="script.js"></script>
</body>
</html>
```


Buon Davante a tutti