



# DelphiDay

italian conference

## Applicazioni Web in Delphi con WiRL e Alpine.js



**Marco Breveglieri**



Software Developer |



Trainer and Consultant |



Tech Content Creator




# MARCO BREVEGLIERI


 Software Developer |  Trainer and Consultant |  Tech Content Creator

 [www.breveglieri.it](http://www.breveglieri.it)

 [marco.breveglieri@abls.it](mailto:marco.breveglieri@abls.it)

 [twitter.com/mbreveglieri](https://twitter.com/mbreveglieri)

 [github.com/marcobreveglieri](https://github.com/marcobreveglieri)

 [linkedin.com/in/marcobreveglieri](https://linkedin.com/in/marcobreveglieri)



11-12 Giugno 2024  
Piacenza





# OPEN-SOURCE PROJECTS

[github.com/mbreveglieri](https://github.com/mbreveglieri)



**CompilaQuindiVa**

[www.twitch.tv/compilaquindiva](https://www.twitch.tv/compilaquindiva)

+  **YouTube**



**Delphi Podcast**

[www.delhipodcast.com](https://www.delhipodcast.com)



**Prometheus Client for Delphi**

[github.com/marcobreveglieri/prometheus-client-delphi](https://github.com/marcobreveglieri/prometheus-client-delphi)

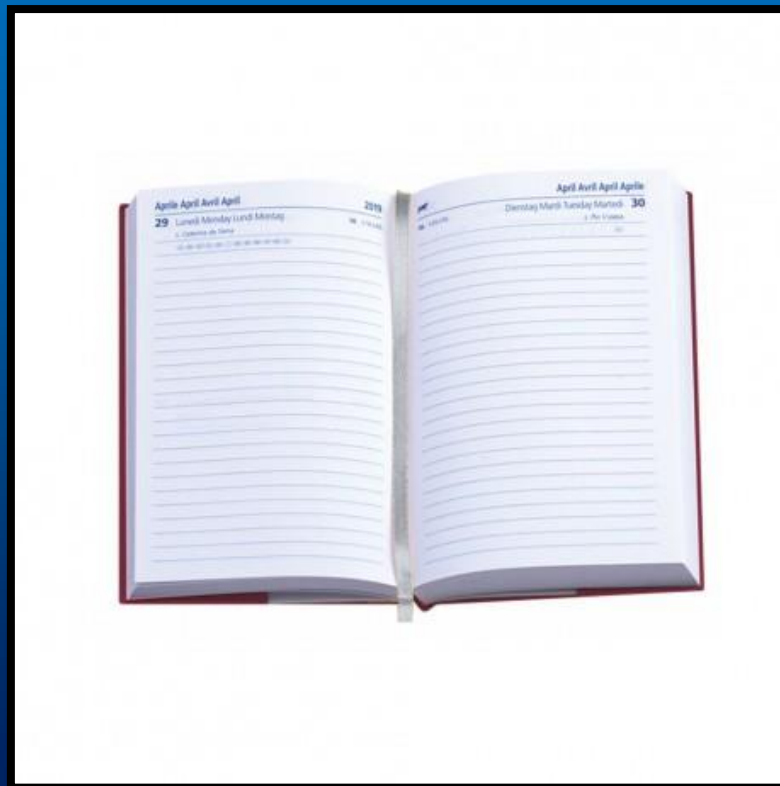


11-12 Giugno 2024  
Piacenza



# AGENDA

---



# Prima di iniziare...

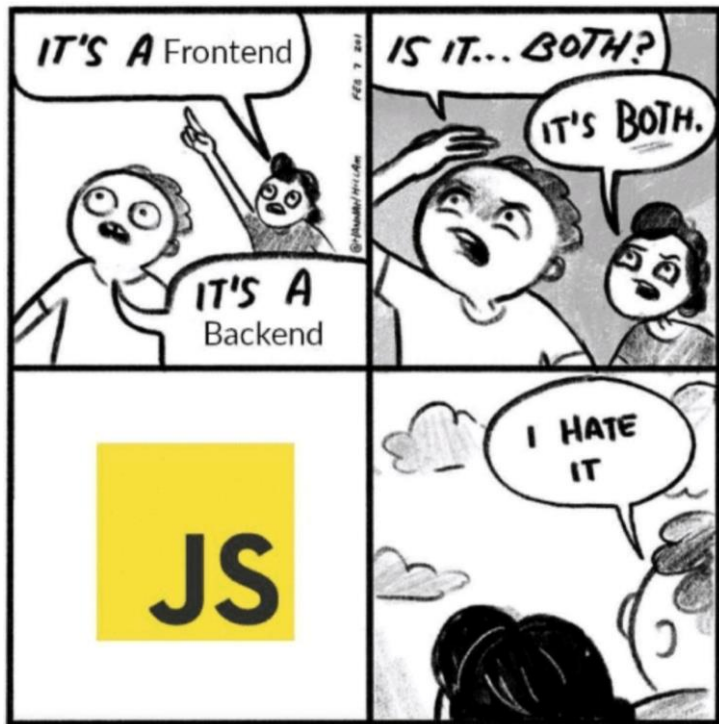
# Piccolo sondaggio...

---

- Chi ha sviluppato (o sta sviluppando) **applicazioni Web con Delphi**?
- Chi utilizza **soluzioni «Delphi-like»** (*IntraWeb, UniGUI, ...*)?
- Chi sta usando **librerie o framework JavaScript** integrate con Delphi come backend (es. *ExtJS*)?
- Chi usa solo **JavaScript e DOM**?



# Tutti odiano JavaScript... o quasi...



- Debolmente tipizzato
- Comportamento di «this»
- Global Scope pollution
- Design non uniforme
- Performance variabili
- Ambiente «ostile»
- Altro? 😊

# (per il meme)

➤ <code>typeof NaN</code>	➤ <code>true===1</code>
➤ <code>"number"</code>	➤ <code>true</code>
➤ <code>9999999999999999</code>	➤ <code>true===1</code>
➤ <code>10000000000000000</code>	➤ <code>false</code>
➤ <code>0.5+0.1==0.6</code>	➤ <code>(!+[[]+[]+![]).length</code>
➤ <code>true</code>	➤ <code>9</code>
➤ <code>0.1+0.2==0.3</code>	➤ <code>9+"1"</code>
➤ <code>false</code>	➤ <code>"91"</code>
➤ <code>Math.max()</code>	➤ <code>91-"1"</code>
➤ <code>-Infinity</code>	➤ <code>90</code>
➤ <code>Math.min()</code>	➤ <code>[]==0</code>
➤ <code>Infinity</code>	➤ <code>true</code>
➤ <code>[]+[]</code>	
➤ <code>""</code>	
➤ <code>[]+{}</code>	
➤ <code>"[object Object]"</code>	
➤ <code>{}+[]</code>	
➤ <code>0</code>	
➤ <code>true+true+true===3</code>	
➤ <code>true</code>	
➤ <code>true-true</code>	
➤ <code>0</code>	



## JavaScript



```
index.js
console.log(3 + 3);
6
console.log("3" + "3");
33
console.log("3" + + 3);
33
console.log(3 + + "3");
6
console.log(3 + 3 - 3);
3
console.log("3" + "3" - "3");
30
```





# Se ami JavaScript (o lo odi di meno)...



*Domani 12 giugno – 14.30*

## Vue.JS e Delphi

Esploriamo assieme *Vue.js*, uno dei framework più diffusi e apprezzati per lo sviluppo frontend, e scopriamo come integrarlo a un backend interamente scritto in *Delphi*.

Attraverso una panoramica dettagliata degli strumenti, scopriamo come integrare progetti di frontend e backend sfruttando le potenzialità di ciascuna tecnologia, lasciando ampio spazio a domande e curiosità con l'ottica di fornire tutte le nozioni necessarie ad affrontare sfide analoghe in futuro, che contemplino altri strumenti e librerie.

# Web Application $\neq$ SPA

---

- **Non è obbligatorio creare una SPA** (*Single Page Application*) tutte le volte
- Librerie e framework come *Angular, React, Vue, Svelte* ecc. sono spesso **complessi per alcuni scenari**
- Alcuni tool sono **difficili da comprendere** e sfruttare al pieno delle loro potenzialità
- Parlando di sviluppo in Delphi
  - Diversi tool basati su Node richiedono progetti separati
  - Il linguaggio JavaScript (e derivati) sono molto diversi da Delphi
    - Sviluppando per il Web si dovrebbe conoscerlo un po'...

Benvenuto  Alpine.js!

# Cos'è Alpine.js?

---

- **Alpine.js** è una **libreria JavaScript** dichiarativo e reattivo che ti consente di aggiungere interattività alle tue pagine HTML con il minimo del codice
- È simile a *Vue.js* o *React*, ma molto più semplice e facile da imparare (non intende sostituirli!)
- Pensa a una sorta di *jQuery* per il web moderno
- L'idea è «inserisci il tuo piccolo script e... fatto!» ⚡

 <https://alpinejs.dev>

# Qualcuno ha detto JQuery?!

---



Calma  
calma





# Alpine.js vs jQuery

## Stile imperativo

(jQuery, DOM, ecc.)

- Il codice è una generica sequenza di passaggi
- Devi specificare ogni fase del processo
- Porta a codice più complesso
- Diviene meno gestibile all'aumentare della complessità

```
// Get a reference to an HTML element
var container = document.getElementById("container");

// Create a new element
var newElement = document.createElement("div");

// Set attributes and content
newElement.id = "newDiv";
newElement.textContent = "This is a new div";

// Append the new element to an existing element
container.appendChild(newElement);

// Add event listeners
newElement.addEventListener("click", function() {
    alert("Element clicked!");
});

// Style the element
newElement.style.backgroundColor = "lightblue";
newElement.style.padding = "10px";
```

# Alpine.js vs JQuery

## Stile dichiarativo

(Alpine.js, React, Angular, Vue.js, ...)

- Il codice descrive come dovrebbe apparire la UI, con i dati e le interazioni
- Consente di concentrarsi sul risultato
- Rende il codice più facile da leggere e comprendere
- Garantisce maggiore manutenibilità al crescere del progetto

```
function App() {  
  // ...  
  return (  
    <div>  
      <h1>Declarative Style with React</h1>  
      <input  
        type="text"  
        value={newText}  
        onChange={e => setNewText(e.target.value)}  
        placeholder="Add a new div"  
      />  
      <button onClick={handleAddClick}>Add</button>  
      <ul>  
        {items.map(item => (  
          <li  
            key={item.id}  
            style={{ backgroundColor: item.backgroundColor, padding: item.padding }}  
          >  
            {item.text}  
          </li>  
        ))}  
      </ul>  
    </div>  
  );  
}
```

# Perché Alpine.js?

---

- **Alpine.js è perfetto per progetti medio-piccoli che richiedono qualche interattività ma non giustificano l'adozione di un framework completo**
  - E' facile da imparare
  - Ha un ridotto «footprint»
  - Può essere integrato in progetti esistenti con uno sforzo minimale

# Una libreria, molti benefit!

---

- Minimale: contiene solo pochi elementi
  - 15 attributi
  - 6 proprietà
  - 2 metodi
- Usabile: scaricalo o richiamalo direttamente nella pagina
  - Disponibile tramite CDN: <https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.min.js>
  - Non richiede Node.js, npm e affini (se non li vuoi o non ne hai bisogno)
- Moderno: supporta tutte le feature più recenti dei browser
  - Supporta pienamente le transizioni CSS
  - Si sincronizza con il «tick» del browser per gli aggiornamenti grafici
- Facile: bassa curva di apprendimento
  - Tutta la documentazione è disponibile
  - I principianti di JavaScript e gli allergici al linguaggio la apprezzano molto
  - Usa principalmente uno stile dichiarativo

# Cosa posso farci?

---

- Implementare le funzionalità tipicamente ricorrenti delle pagine Web
  - Validazione dei Form
  - Commutazione delle classi (CSS)
  - Mostrare/nascondere elementi
  - Ordinare e filtrare liste
  - Potenziare pagine statiche con funzionalità dinamiche
- Adatto a molti degli scenari di utilizzo più semplici
  - Raccolta di dati, questionari, sondaggi
  - Gestione di un carrello e-commerce
  - Sviluppo di giochi HTML5 per browser
  - Landing page accattivanti
  - Prototipazione di app e idee



# Dalla teoria alla pratica



# Il cuore di Alpine: «state»

---

- I dati che Alpine osserva per apportare cambiamenti alla pagina
- E' al centro di tutto quello che è possibile fare con Alpine
- Può essere
  - Local usando **x-data**
  - Globale usando **Alpine.store()**
- Alpine.js usa la direttiva **x-bind** per aggiornare l'HTML in tempo reale
- Gli aggiornamenti vengono eseguiti in base al cambiamento dei dati

# Le direttive

---

- Alpine.js usa le «direttive» per aggiungere funzionalità al tuo HTML
- Sono «attributi speciali» che iniziano con il prefisso x-
- Consentono di eseguire azioni come mostrare e nascondere elementi
- Per gestire gli eventi, basta usare la direttiva x-on
- Gli eventi permettono di eseguire azioni quali inviare un modulo, effettuare chiamate a una API e tanto altro...

# Demo time!

---



# Breve riepilogo

---

## Direttive

- x-data
- x-on
- x-text
- x-show
- x-model
- x-for

Altre qui:

<https://alpinejs.dev/start-here> 

## Eventi

- @event (x-on:event)
- @event.modifier

## Funzioni

- \$store
- \$watch
- \$dispatch



...e Delphi? 

# Delphi per il backend

---

- Delphi è perfetto per il backend della tua applicazione
- Usa il framework Web che preferisci!
  - RAD Server (incluso in Delphi)
  - WiRL (demo a breve!)
  - DelphiMVCFramework
  - MARS
  - Horse
  - mORMot
  - WebBroker
- Ricordati di abilitare il supporto ai file statici

# Demo time!



# Conclusioni

# Alpine.js in breve

---

- Alpine.js è un framework divertente e facile da imparare
- Può aiutarti ad aggiungere interattività alla tua applicazione Web
- Richiede solo un minimo sforzo per risultati più che dignitosi
- E' facilmente integrabile nei progetti esistenti
- Non richiede tooling JavaScript e quindi è compatibile con tutte le principali librerie Delphi per lo sviluppo Web
- Non necessita di creare progetti separati semplificando la manutenzione
- Si integra perfettamente all'applicazione Web server senza altri package
- Semplifica la configurazione del progetto in generale (es. non serve CORS)



# Ma c'è di più!

---

## → Alpine UI Components

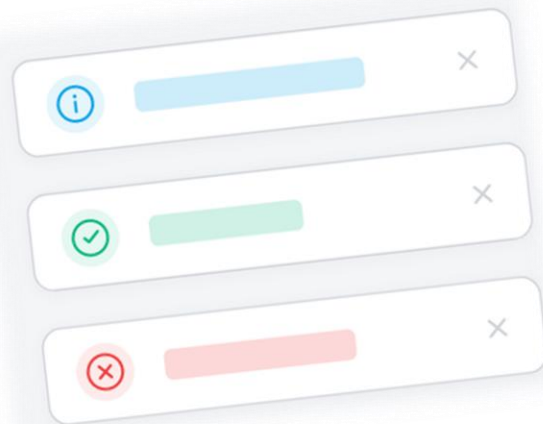
- Molti siti richiedono componenti visuali per una UI più articolata
  - Finestre modali, dropdown, schede, ecc.
  - Costruirli a mano è tutto fuorché semplice
- Questa libreria implementa tutti i principali widget
  - Usabili con copia/incolla e modifiche successive
  - Spendi meno tempo a «reinventare la ruota»

## → Risorse sul sito ufficiale

- Documentazione ufficiale completa
- Screencast
- Integrazioni di terze parti
- Supporto tecnico

## → Alternative interessanti con finalità simili

- HTMX (se proprio vuoi ridurre al minimo JavaScript)



# Domande? 🤔

---





THANK YOU