

HTML (HYPERTEXT MARK UP LANGUAGE)

CONSENTE DI DEFINIRE LA STRUTTURA (LAYOUT) DI PAGINE WEB
OGNI PAGINA WEB STRUTTURATA CON UTILIZZO DI LINGUAGGIO HTML PUO' ESSERE VISTA COME UN DOM (DOCUMENT OBJECT MODEL), OVVERO COME UNA STRUTTURA AD ALBERO CON NODI PADRE E NODI FIGLIO.

LIMITE : LINGUAGGIO STATICO (NON CONSENTE DI EFFETTUARE VARIAZIONI DEL DOM IN BASE AL VERIFICARSI DI EVENTI CONDIZIONATI DA AZIONI CHE VENGONO ESEGUITE DA UN UTILIZZATORE DELLA PAGINA).

JAVASCRIPT = LINGUAGGIO ADERENTE A DUE PARADIGMI (OOP e FUNZIONALE), EVENT DRIVEN (IN GRADO DI MODIFICARE DINAMICAMENTE IL DOM ORIGINARIO STRUTTURATO DA HTML), UNTYPED (NON OCCORRE ASSOCIARE A VARIABILI O FUNZIONI O METODI TIPI IN FASE DI SCRITTURA DEL CODICE, MA I TIPI VERRANNO ASSEGNATI A RUNTIME DAL RUNTIME ENVIRONMENT), NON COMPILATO MA SOLO INTERPRETATO.

RUNTIME ENVIRONMENT => JAVASCRIPT ENGINE

REACT JS = TECNOLOGIA FONDATA SULL'UTILIZZO DEL LINGUAGGIO JAVASCRIPT E CHE IMPLEMENTA IL MECCANISMO DEL VIRTUAL DOM, CONSENTENDO DI AGGIORNARE LO STATO DI UNA PAGINA WEB PARZIALMENTE E SPECIFICATAMENTE RISPETTO ALLE ESIGENZE

REACT TS = TECNOLOGIA FONDATA SULL'UTILIZZO DEL LINGUAGGIO TYPESCRIPT E CHE IMPLEMENTA IL MECCANISMO DEL VIRTUAL DOM, CONSENTENDO DI AGGIORNARE LO STATO DI UNA PAGINA WEB PARZIALMENTE E SPECIFICATAMENTE RISPETTO ALLE ESIGENZE

ANGULAR = FRAMEWORK GOOGLE/MICROSOFT FONDATA SULL'UTILIZZO DEL LINGUAGGIO TYPESCRIPT

CSS E BOOTSTRAP = TECNOLOGIE PER LO STYLING DELLE PAGINE WEB

ECMA SCRIPT 6	
2015	
OLD JAVASCRIPT	MODERN JAVASCRIPT
LIVE OBJECTS	CLASS OBJECTS
FUNCTIONS DECLARATION API	CUSTOM METHODS DECLARATION
METHODS (NO CUSTOM)	let(variable declaration)
var (variable declaration)	YES SPREAD E REST OPERATOR
NO SPREAD e REST OPERATOR	YES ARROW AND ANONYMOUS FUNCTIONS
ONLY TRADITIONAL FUNCTIONS	

ESISTONO TRE MODALITA' PER LA CREAZIONE DI UN LIVE OBJECT

1. Inizializzare una reference che referencia un Oggetto con le proprietà valorizzate all'interno di square brackets {}
BEST PRACTICE : utilizzare la parola chiave `const` per la reference in maniera tale che la reference sia immutabile, ovvero non possa essere riassegnata ad un altro Oggetto. In ogni caso pur usando `const` è possibile modificare lo stato dell'oggetto.
2. Inizializzare una reference che referencia un Oggetto inizialmente vuoto senza proprietà tramite parola chiave `new` e classe `Object`. In tal caso le proprietà dell'Oggetto devono esser inserite successivamente
3. Dichiarare una funzione che setti delle variabili al valore dei parametri in input dopodichè occorre dichiarare una reference che referencia un Oggetto tramite parola chiave `new` e nome della funzione

LA PRIMA E LA TERZA SOLUZIONE SONO LE MIGLIORI, IN QUANTO COINVOLGONO UN BASSO NUMERO DI OPERAZIONI COMPUTAZIONALI RISPETTO ALLA SECONDA.

NB:

UNA FUNZIONE JAVASCRIPT PUO' ESSERE INVOCATA:

- IN BASE AD UN EVENTO SULLA PAGINA (AD ESEMPIO L'AZIONE DI UN CLICK SU UNA PAGINA)
- TRAMITE KEYWORD `new` per la generazione di un live object

IL Javascript Engine casta automaticamente i JSON in Oggetti Javascript

Se dobbiamo convertire un Oggetto Javascript in JSON occorre scrivere del codice custom per chiedere la conversione al Javascript Engine (FUNZIONE Javascript stringify)

ESISTONO DUE TIPOLOGIE DI JSON

- SIMPLE JSON (in cui ogni proprietà ha un solo valore)
- COMPLEX (NESTED) JSON (in cui almeno una proprietà ha più di un valore)

STRUTTURE DATI JAVASCRIPT (TUTTE DINAMICHE) :

ARRAY

- DINAMICA
- INDICIZZATA E ORDINATA
- SIA OMOGENEA CHE ETEREOGENEA
- AMMETTE DUPLICATI

MAP

- DINAMICA
- INDICIZZATA E ORDINATA
- SIA OMOGENEA CHE ETEREOGENA
- AMMETTE DUPLICATI

SET

- DINAMICA
- NON INDICIZZATA MA ORDINATA
- SIA OMOGENEA CHE ETEREOGENEA
- NON AMMETTE DUPLICATI

IN JAVASCRIPT ESISTONO DUE TIPI DI OPERATORI DI UGUAGLIANZA

- ==
- === (STRICT EQUAL)

== confronta i valori ma non i tipi a runtime; se i valori sono uguali restituisce true, anche se i tipi sono differenti

=== confronta valori e tipi a runtime; restituisce true solo se tipi e valori a runtime coincidono

DIFFERENZA TRA var e let:

var e let hanno entrambi global scope (scope di script) se dichiarate esternamente ad un blocco di codice

se dichiarate internamente ad un blocco di codice (come un if o un for) let ha scope di blocco di codice (cioè non viene vista fuori dal blocco), mentre var ha sempre global scope, quindi viene vista anche al di fuori del blocco di codice rispetto al quale è stata dichiarata

IN JAVASCRIPT NON ESISTONO INTERFACCE MA SOLO CLASSI

NON E' POSSIBILE IMPLEMENTARE POLIMORFISMO DI OGGETTI IN JAVASCRIPT, IN QUANTO NON SI POSSONO ASSEGNARE IN FASE DI SCRITTURA DEL CODICE TIPI

IN JAVASCRIPT NON E' POSSIBILE IMPLEMENTARE PATTERN LOOSE COUPLING

TALI FRAGILITA' OOP SONO STATE SUPERATE DAL LINGUAGGIO TYPESCRIPT

UNA APPLICAZIONE SINCRONA E' UNA APPLICAZIONE NELLA QUALE LA COMUNICAZIONE TRA CLIENT E SERVER AVVIENE TRAMITE COPPIE RICHIESTA-RISPOSTA, OVVERO IL CLIENT NON PUO' EFFETTUARE UNA SUCCESSIVA RICHIESTA PRIMA DI AVER RICEVUTO LA RISPOSTA DAL SERVER ALLA PRECEDENTE RICHIESTA

I SISTEMI SINCRONI VENGONO DENOMINATI ANCHE BLOCKING SYSTEMS

APPLICAZIONE ASINCRONA = IL CLIENT PUO' EFFETTUARE N RICHIESTE PRIMA DI RICEVERE LE RISPOSTE ALLE SUE RICHIESTE, SENZA ATTENDERE LA RISPOSTA PER EFFETTUARE ALTRE RICHIESTE, ATTENDENDO IN BACKGROUND LE RISPOSTE DA PARTE DEL SERVER

UNA FUNZIONE DI CALLBACK E' UNA FUNZIONE PASSATA COME ARGOMENTO NELLA LISTA DEI PARAMETRI DI UN METODO O DI UN'ALTRA FUNZIONE

PUO' ESSERE UTILE PER IMPLEMENTARE MECCANISMI ASINCRONI
