

HTML DOM DEFINISCE:

- ELEMENTI HTML COME OGGETTI
- PROPRIETÀ degli ELEMENTI HTML
- I METODI X ACCEDERE AGLI ELEMENTI HTML
- GLI EVENTI PER TUTTI gli ELEMENTI HTML

HTML DOM è UNO STANDARD CHE DEFINISCE COME OTTENERE, CAMBIARE, ~~AGGI~~ AGGIUNGERE O MODIFICARE ELEMENTI HTML.

WINDOW = RADICE. RAPPRESENTA LA FINESTRA CORRENTE DEL BROWSER. IN FIGURA SONO:

- NAVIGATOR = BROWSER IN QUANTO APPPLICAZIONE
- DOCUMENT = IL CONTENUTO DELLA FINESTRA
- LOCATION = INFO SU URL CORRENTE
- HISTORY = ELENCO degli URL VISITATI di RECENTE

TRAMITE IL MODELLO DOM JS PUO':

- ^{RIMUOVERE/AGGIUNGERE} CAMBIARE TUTTI gli ELEMENTI e ATRIBUTI HTML di UNA PAGINA
- CAMBIARE TUTTI gli STILI CSS
- REAGIRE a EVENTI nella PAGINA
- CREARE NUOVI ELEMENTI

WINDOW.DOCUMENT.GETELEMENTBYID("HEADER");

[Signature]

[Signature]

CRUD STA PER

- CREATE - READ - UPDATE - DELETE

SONO 4 OPERAZIONI DI DATABASE PRIMITIVE. QUESTE OP. SI POSSONO ASSOCIARE AI VERBI HTTP USATI PIÙ FREQUENTEMENTE IN REST:

- CREATE → POST = CREAZIONE DI UNA RISORSA ~~NUOVA~~ FIGLIO
- READ → GET = RECUPERA LA RAPP. DI UNA RISORSA.
- UPDATE → PUT = AGGIORNA / CREA UNA RISORSA.
- DELETE → DELETE = ELIMINA UNA RISORSA

PUT ⇒ CLIENT DECIDE IL NOME DELLA RISORSA

POST = SERVER DECIDE IL NOME DELLA RISORSA

PUT /ESAMI/ID

POST /ESAMI

ci sono 2 tipi di OFFLINE STORAGE:

- SESSION STORAGE = PERMETTE di MEMORIZZARE DATI SPECIFICI A UNA FINESTRA/TAB. PERMETTE di ISOLARE INFO in OGNI FINESTRA. NON È PERSISTENTE. VIENE MANTENUTO PER IL TEMPO in cui LA FINESTRA/TAB del BROWSER È APERTA.
- LOCAL STORAGE = PERMETTE di SALVARE i DATI PERSISTENTI nel PC dell'UTENTE, ATRAVERSO il BROWSER. OGNI DATO SALVATO nel LOCAL STORAGE PUÒ ESSERE RECUPERATO.

SIMILITUDINE con COOKIES: DATI MEMORIZZATI COME COPPIE CHIAVE-VALORE

DIFFERENZE con COOKIES:

- COOKIES SONO LETTI LATO SERVER, LOCAL STORAGE SONO DISPONIBILI SOLO LATO CLIENT

~~LOCAL STORAGE~~ SONO una compromissione

- I COOKIES INVIATI CON OGNI RICHIESTA AL SERVER PORTANO OVERHEAD SOSTANZIALI in TERMINI di BANDA USATA.

LOCAL STORAGE RISIEME NELL'HARD DISK dell'UTENTE e VIEN LETTO A LOGO ZERO.

- IL LOCAL STORAGE FORNISCE MOLTO PIÙ SPAZIO di MEMORIZZAZIONE RISPETTO AI COOKIES

HTML5

- È UNO STANDARD PER STRUTTURARE E PRESENTARE IL CONTENUTO del WWW.
- È PRAGMATICO
- FORNISCE SUPPORTO PER TUTTO CIÒ CHE È JS.
- AGGIUNGE SEMANTICA, MIGLIORE DEFINIZIONE delle COMPONENTI delle WEB PAGE, i TAG HTML FORNISCE INFO AGGIUNTIVE UTILI x i MOTORI di RICERCA

INTRODUCE MOLTE FUNZIONALITÀ RISPETTO AD HTML:

- NUOVI ELEMENTI STRUTTURALI: <HEADER>, <FOOTER>, <NAV> (in TOT 28 TAG)
- FORM 2.0
- SUPPORTO NATIVO del BROWSER <AUDIO> e <VIDEO>
- CANVAS
- WEB STORAGE = MEMO CHE SI TROVA DENTRO il BROWSER e PERMETTE di MEMORIZZARE i DATI COME COPPIA (CHIAVE, VALORE)
- DRAG e DROP

LA SINTASSI È:

- PIÙ PERMISSIVA
- CASE INSENSITIVE
- TAG di CHIUSURA non RICHIESTI
- VIRGOLETTE E VALORI x ATTRIBUTI OPZIONALI

~~SEMANTIC~~

SEMANTIC MARKUP = ASSOCIA UN SIGNIFICATO AL CONTENUTO

- È UNA SINTASSI CHE DA L'IDEA di COSA SI STA FACENDO SIA A UNA MACCHINA CHE A UN UOMO.
(BROWSER)
- PAGINE ADATTE x MOSTRARE il CONTENUTO in MANIERA COMPRESIBILE A UTENTI con DISABILITÀ

ci sono 3 modelli di servizio del cloud computing:

- SAAS (GOOGLEDOCS)

- UTENTE USA APPICAZIONI di UN CLOUD PROVIDER CHE SONO ESEGUITE SULL'INFRASTRUTTURA CLOUD
- IL CLIENTE non GESTISCE o CONTROLLA L'INFRASTRUTTURA CLOUD (INCLUSO RETE, SERVER, SO, STORAGE)
- IL CLIENTE CONTROLLA UN NUMERO LIMITATO di CONFIGURAZIONI SPECIFICHE PER gli UTENTI
- ESTERNALIZZA UN SAAS E L'UTENTE LO USA VIA RETE

- PAAS

- UTENTE INSTAUA e CREA APP TRAMITE SERVIZI SUPPORTATI dal PROVIDER
- UTENTE MANTIENE il CONTROLLO SULLA PIATTAFORMA INSTAUA.
- UTENTE non GESTISCE o CONTROLLA L'INFRASTRUTTURA SOTTOSTANTE (INCLUSO RETE, SERVER, SO, STORAGE)
- UTENTE MANTIENE CONTROLLO delle APP INSTAUATE ~~alla propria configurazione~~

- IAAS

- UTENTE GESTISCE CPU, MEMORIA, STORAGE, RETE e RISORSE di COMPUTAZIONE AGGIUNTIVE, SO, APP INSTAUATE
- UTENTE non GESTISCE o CONTROLLA SOLO L'INFRASTRUTTURA CLOUD
- UTENTE PUO' INSTAUARE ed ESEGUIRE CODICE GENERICO INCLUSO SO e APPICAZIONI
- ~~UTENTE~~ UTENTE HA CONTROLLO LIMITATO di COMPONENTI di RETE
- OFFRE RISORSE VIRTUAUZZATE ON DEMAND
- MONITOR + VISIONE ALL'HM

JSON è:

- UN FORMATO ADATTO ALL'INTERSCAMBIO DI DATI FRA APP. CLIENT-SERVER
- USATO QUANDO SI SERVONO APP BASATE IN JS.
- FACILE DA LEGGERE E SCRIVERE.
- FORMATO DI SCAMBIO BASATO IN TESTO E MOLTO LEGGERO
- SUPPORTA DUE STRUTTURE DATI < INSIEME DI COPPIE NOME/VALORE
LISTA ORDINATA DI VALORI

è POSSIBILE CONFRONTARE JSON E XML NELLA BASE DI:

- PARSING = IL METODO EVAL DI JS FA PARSING DI JSON, IN XML IL PARSING È + DIFFICILE
- USO DI ARRAY = XML USATO A DESCRIVERE DATI STRUTTURATI CHE NON INCLUDONO ARRAY. JSON INCLUDE ARRAY.
- VERBOSITÀ = XML È + VERBOSO DI JSON. È + FACILE E VELOCE JSON.

SOMIGLIANZE TRA JSON E XML:

- SONO SEMPLICI E APERTI
- SUPPORTO UNICODE
- RAPP. DATI AUTODESCRITTIVI.