================================================================================================

**Tema:** JavaScript

**Modulo:** 02

**Aula:** Online dal vivo

**Giorni:** 26/05/2025 a 30/05/2025

**Professor:** Marika Di Blasio

================================================================================================

================================================================================================

**JavaScript**

Creato in 1995 da Brendan Eich presso Netscape Communications. Originariamente chiamato Mocha, poi LiveScript, infine JavaScript, era progettato per rendere le pagine web più interattive.

================================================================================================

**PRIMO ASSAGGIO DI CODICE**

Il console.log() è una funzione per mostrare messaggi nella console del browser (strumenti per sviluppatori)

“Hello, World!” è una stringa, un testo da visualizzare.

Ogni istruzione JavaScript termina spesso con un punto e virgola ( ; ) anche se non sempre obbligatorio.

================================================================================================

**VARIABILI**

Contenitori per memorizzare dati. Puoi pensare a loro come etichetta che dai a un valore, che è utili per riutilizzare e modificare i dati nel tuo codice. Utili per riutilizzare e modificare i dati nel tuo codice. In JavaScript usiamo parole chiave per dichiarare le variabili:

* let: Per variabili il cui valore può cambiare;
* const: Per variabili il cui valore non dovrebbe cambiare (constanti);
* var: Dobbiamo evitare se possibile, ha un comportamento più complesso.

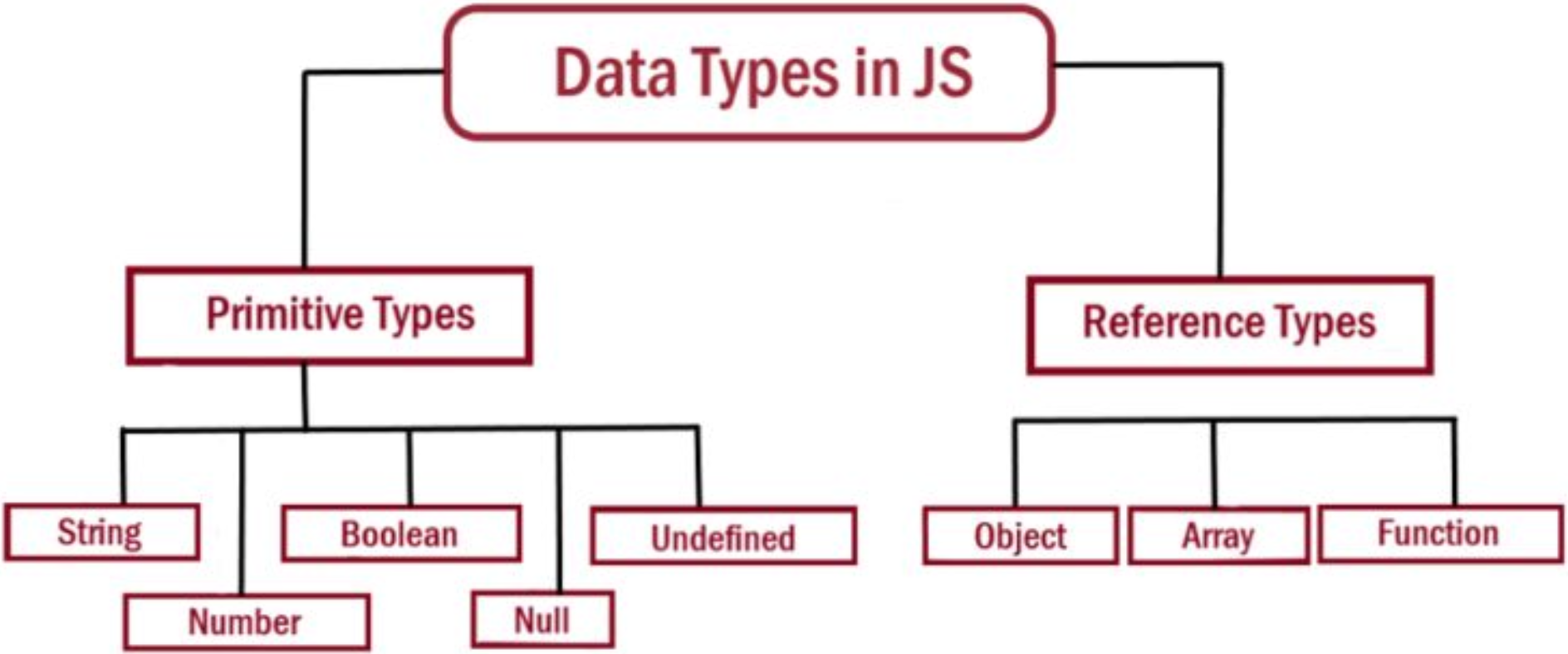
Usiamo l’operatore di assegnazione ( = ) per dare un valore a una variabili.

================================================================================================

**TIPI DI DATI FONDAMENTALI**

Abbiamo diversi tipi di dati:

* Stringa (String): Sequenza di caratteri (testo) racchiusa tra virgolette singole ( ‘ ) o doppie ( “ );
* Numero (Number): Rappresenta valori numerici (interi o decimali);
* Booleano (Boolean): Rappresenta um valore di verità: True / False;
* null: Rappresenta l’assenza intenzionale di un valore
* undefinied: Indica che una variabile è stata dichiarata ma non ha ancora ricevuto un valore;

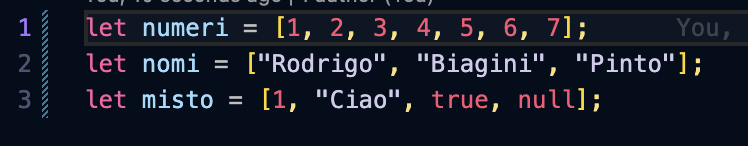


Base dei tipi di variabile.

================================================================================================

**ARRAY**

Una array è una lista ordinata di valori, dove ogni elemento ha un indice, che inizia da 0, può contenere elementi di qualsiasi tipo, anche altre array o oggetti, e si definiscono usando le parentesi quadre []. Se accede agli elementi utilizzando l’indice tra parentese quadri.



Dopo, abbiamo gli oggetti.

================================================================================================

**OGGETTI**

Un oggetto è una collezione di proprietà, dove ogni proprietà ha un nome (chiave) e un valore, dove le chiave sono generalmente stringhe (o simboli) ed i valori possono essere di qualsiasi tipo (primitivo o non primitivo), che si definiscono usando le parentese graffe { }

Se accede alle proprietà usando la notazione a punto ( . ) o la notazione a parentese quadre ( [ ] ).

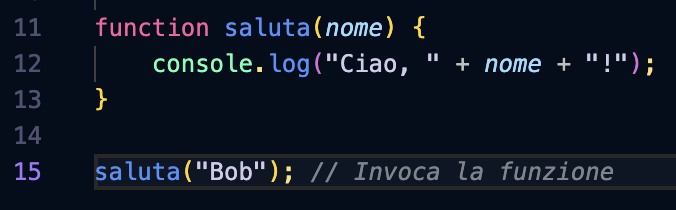


La notazione a parentesi quadre è utile quando il nome della proprietà è memorizzato in una variabile o contiene spazi o caratteri speciali.

================================================================================================

**FUNZIONI**

Anche le funzioni in JavaScript sono considerate oggetti di “prima classe”, rappresentando blocchi di codice che possono essere definiti e poi invocati (eseguiti) più volte. Possono ricevere parametri (input) e restituire un valore (output).

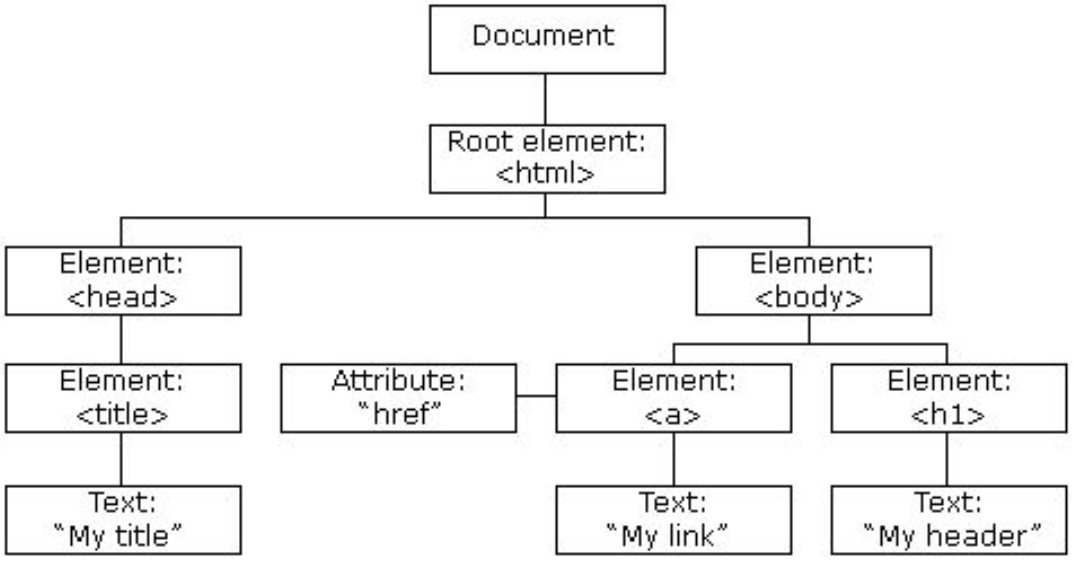


================================================================================================

**DOM DOCUMENT OBJECT MODELO**

Il DOM è la rappresentazione strutturata (ad albero) di un documento HTML;

Ogni elemento HTML, attributo e teste all’interno del documento è un nodo nell’albero del DOM. JavaScript utilizza il DOM per accedere e manipolare gli elementi della pagina web.

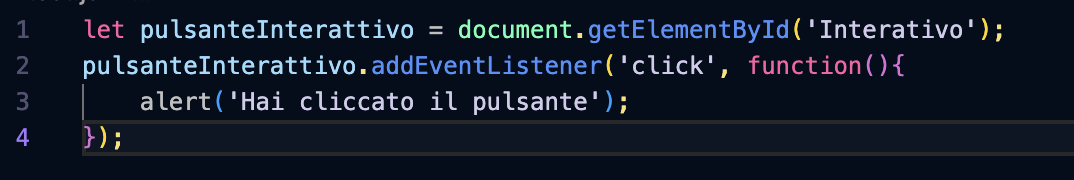


E per accedere agli elementi del DOM, possiamo usare:

* document.getElementByID(‘ID): Seleziona un elemento tramite il suo attributo id (unico);
* document.getElementsByClassName(‘class’): Seleziona tutti gli elementi con una determinata class (restituisce un HTMLCollection);
* document.getElementsByTagName(‘tag’): Seleziona tutti gli elementi con un determinato tag HTML (restituisce un HTMLCollection);
* document.querySelector(‘selettoreCSS’): Seleziona il primo elemento che corrisponde a un selettore CSS (come div, .classe, #id, p, span).
* Document.querySelectorAll(‘selettore CSS’): Seleziona tutti gli elementi che corrispondono a un selettore CSS (restituisce un NodeList).

Il DOM consente di reagire alle azioni dell’utente (click, mouseover, tasti premuti, ecc). Si utilizzano gli eventi listener per “ascoltare” specifici eventi su specifici elementi e eseguire codice in rIsposta.

elemento.addEventListener(‘tipoEvento’, funzioneCallback): Attacca um gestore di eventi all’elemento.



================================================================================================

**MODIFICARE GLI STILI CSS**

La proprietà “elemento.style” consente di accedere e modificare gli stili CSS inline di un elemento, la proprietà elemento.classList fornisce metodi utili per manipolare le classi CSS di un elemento:

* elemento.classList.add('nuovaClasse'): Aggiunge una o più classi.
* elemento.classList.remove('classeDaRimuovere'): Rimuove una o più classi.
* elemento.classList.toggle('classeDaAlternare'): Aggiunge la classe se non esiste, la rimuove se esiste.
* elemento.classList.contains('classeDaVerificare'): Restituisce true se l'elemento ha la classe specificata.

================================================================================================

**EVENTO**

Un evento è un’azione o un’occorrenza che succede nel browser (o all’interno della pagina web. Può essere causato dall’utente (es. clic del mouse, pressione di un tasto, movimento del mouse) o dal browser stesso (es. caricamento della pagina, ridimensionamento della finestra, errore):

Click: Clic su un elemento.

Mouseover: Il puntatore del mouse si sposta sopra un elemento.

Mouseout: Il puntatore del mouse esce da un elemento.

Keydown: Un tasto viene premuto.

Keyup: Un tasto viene rilasciato.

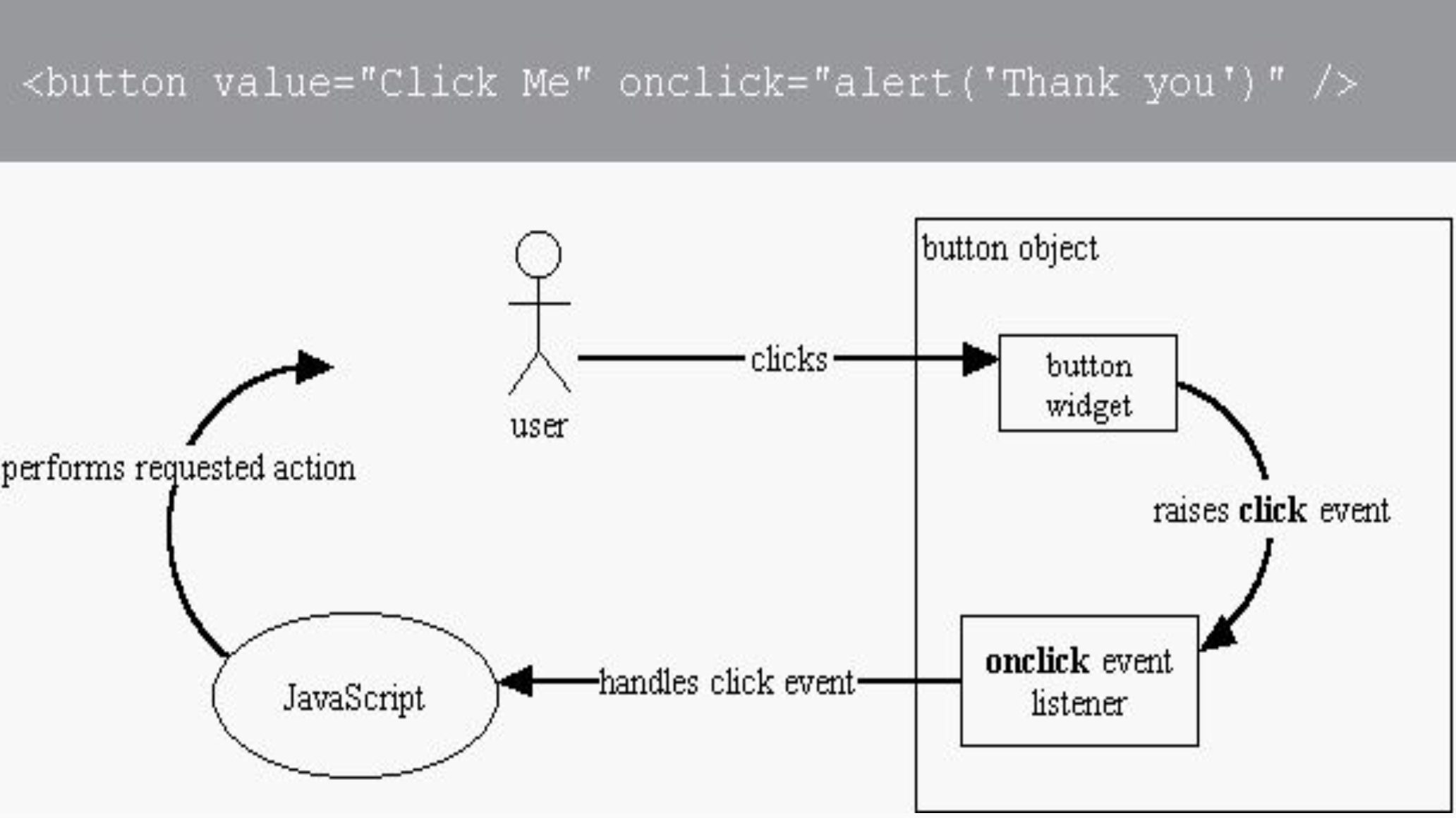
Submit: Un form viene inviato.

Change: Il valore di un elemento di input cambia.

Lood: La pagina o una risorsa (es. immagine) è stata caricata completamente.

Resize: La finestra del browser è stata ridimensionata.

Scroll: La barra di scorrimento è stata spostata.



================================================================================================

**EVENT LISTENER**

Un event listener è una funzione che “ascolta” un particolare evento su uno specifico elemento HTML. Quando l’evento si verifica, la funzione listener (chiamata anche callback dell’evento o gestore dall’evento= viene eseguita.

tipoEvento: Una stringa che specifica il tipo di evento da ascoltare (es. ‘Click’, ‘mouseover’, ‘keydown’)

funzioneCallback: La funzione JavaScript da eseguire quando l’evento si verifica

================================================================================================

**TITOLO**

Testo.

================================================================================================