|  |  |  |  |
| --- | --- | --- | --- |
| [Esempio01](#esempio01) |  |  |  |
| [Esempio01bis](#esempio01bis) |  |  |  |
| [Esempio02](#esempio02) |  |  |  |
| [Esempio03](#esempio03) |  |  |  |
| [Oggetti Letterali](#oggettiLetterali) |  |  |  |
| [Oggetti con diverse proprietà](#oggettiProprietà) |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Quando si crea un oggetto si crea un **riferimento all’oggetto**.

Il riferimento all’oggetto si può anche memorizzare in una variabile.

Si dice anche che una variabile punta ad un oggetto.

**sintassi della creazione di un oggetto in JavaScript utilizzando new**

***let nomeOggetto = new oggetto(parametriCostruttore);***

[Inizio](#Inizio) Esempio 01:

let dataCorrente = new Date();

La variabile dataCorrente contiene il riferimento all’istanza Date(), quindi la possiamo utilizzare all’interno del nostro programma con i metodi propri dell’oggetto Date.

let annoCorrente = dataCorrente.getFullYear();

utilizziamo il metodo getFullYear() per ottenere l’anno corrente.

document.write(annoCorrente);

[Inizio](#Inizio) Esempio 01bis:

let data = new Date('2023,12,31');

Qui al costruttore passiamo una stringa che indica una determinata data.

document.write(data);

Il metodo **toLocaleDateString()** in JavaScript

let data2 = new Date('2023,12,31'). toLocaleDateString();

Questo metodo restituisce un oggetto date come stringa usando le convenzioni locali.

document.write(data2);

[Inizio](#Inizio) Esempio 02:

let stringa = new String('Ciao a tutti da Classe IV B');

La variabile stringa contiene il riferimento all’istanza String(), dove passiamo come parametro una stringa qualunque.

document.write(stringa);

document.write("<br>");

document.write("La lunghezza di questa stringa è di ");

document.write(stringa.length);

L’oggetto String ha la proprietà length che restituisce il numero di caratteri contenuti in una stringa.

document.write(" caratteri");

document.write("<br>");

document.write("Ho estratto il seguente carattere da questa stringa: ");

Estrarre un carattere da una stringa

document.write(stringa.charAt(3));

document.write("<br>");

document.write("Sostituzione del testo con creazione di una nuova stringa: ");

document.write(stringa.replace("da", "dalla"));

Sostituire i caratteri in una stringa

document.write("<br>");

document.write("Modifica della stringa in maiuscolo: ");

document.write(stringa.toUpperCase());

Utilizzato il metodo toUpperCase() della classe String() per trasformare il testo in maiuscolo.

const info = "INFORMAZIONE IMPORTANTE";

document.write("Modifica della stringa in MINUSCOLO: ");

document.write(info.toLowerCase());

Utilizzato il metodo toLowerCase() della classe String() per trasformare il testo in maiuscolo.

document.write("Estrarre una sottostringa con substr: ");

document.write(stringa.substr(17, 4));

La forma della funzione substr è la seguente:

 stringa.substr(inizio, numero\_caratteri)

document.write("<br>");

document.write("Estrarre una sottostringa con substring: ");

document.write(stringa.substring(17, 21));

La definizione di substring è invece la seguente:

 stringa.substring(inizio, fine)

document.write("<br>");

let pensiero = 'Ciao a tutti da Classe IV B';

let pensieroSplit = pensiero.split(' ');

Il metodo split in JavaScript consente di dividere una stringa in base ad un separatore, salvare le sotto stringhe ottenute in un array e restituire quindi l’array così formato.

La sintassi di questo metodo è la seguente: **stringa.split(separatore, limite)**.

Dove stringa rappresenta dunque la stringa da suddividere. Il parametro separatore è obbligatorio ed indica quale elemento utilizzare per effettuare la divisione (ad esempio uno spazio vuoto, una virgola, un trattino). Il parametro limite è invece opzionale e rappresenta il numero di suddivisione da fare. Quindi gli elementi che vanno oltre questo limite non verranno inclusi nella suddivisione.

document.write(pensieroSplit);

console.log(pensieroSplit);

Nella console del browser visualizzeremo un array di 7 elementi, come mostrato sotto:

Array(7) [ “Programmare”, “aiuta”, “lo”, “sviluppo”, “del”, “pensiero”, “logico” ]

document.write("<br>");

let pensiero2 = 'Ciao a tutti da Classe IV B';

let pensieroSplit2 = pensiero2.split('');

Se non inseriamo lo spazio tra gli apici avremmo ottenuto la suddivisione per lettera e quindi nel nostro caso un array di 27 elementi

document.write(pensieroSplit2);

console.log(pensieroSplit2);

[Inizio](#Inizio) Esempio 03:

costruzione di una stringa utilizzando il backtick (Alt 96)

let nominativo ="Bruce Springsteen";

let progressivo = 1;

let soprannome = "The Boss";

voglio ottenere la seguente stringa:
 Il cantante n. '1' Bruce Springsteen è soprannominato: "The Boss"

let cantante = `Il cantante n. '${progressivo}' ${nominativo} è soprannominato: "${soprannome}"`;

document.write("<br>"+cantante);

Metodo .slice

Il metodo slice() del linguaggio Javascript mi permette di estrarre una sequenza di elementi di un array

**.slice(inizio, fine)**

Il metodo ha due parametri obbligatori

Il primo parametro è la posizione dell'elemento iniziale (**incluso**) nell'indice dell'array
Il secondo parametro è la posizione dell'elemento finale (**escluso**) nell'indice dell'array

let numeri = [5,4,3,2,6,9,62];

let risultato = numeri.slice(2,4);

document.write (risultato);

console.log (risultato);

secondo esempio metodo slice

Selezioniamo tutti gli elementi di un array di stringhe partendo dal secondo.

Supponiamo di avere questo array di stringhe:

let oggettiScolastici = ['matita','gomma','temperino','quaderno','matita'];

Dopo utilizziamo il metodo slice specificando solo il primo parametro.

let risultato2 = oggettiScolastici.slice(2);

document.write(risultato2);

console.log (risultato2);

Nella console vedremo tutti gli elementi a partire dal secondo: Array(3) [ “temperino”, “quaderno”, “matita” ].

Esercizio **1**/7 di Stringhe

Scrivi un programma che prenda una stringa numerica e la converta in un numero, stampando il risultato in console

const str = "123";

const num = parseInt(str);

console.log("La stringa convertita in numero è: " + num);

Esercizio **2**/7 di Stringhe

Scrivi un programma che prenda una stringa, la converta in maiuscolo e stampi il risultato sulla console.

const saluto = "ciao";

const stringaMaiuscola = saluto.toUpperCase();

console.log("La stringa in maiuscolo è: " + stringaMaiuscola);

Esercizio **3**/7 di Stringhe

Scrivi un programma che prenda queste due stringhe (“La mia classe preferita è: ” “IVB”), le concateni e stampi il risultato e la lunghezza della stringa concatenata sulla pagina web.

const stringa1 = "La mia classe preferita è: ";

const stringa2 = "IVB ";

const risultato = stringa1 + " " + stringa2;

document.write("La stringa concatenata è: " + risultato);

document.write("<br>");

document.write("La lunghezza della stringa concatenata è: " + risultato.length);

Esercizio **4**/7 di Stringhe

Scrivi un programma che prenda una stringa maiuscola, la converta in minuscolo e stampi il risultato sulla console.

const maiusc = "CIAO";

const stringaMinuscola = maiusc.toLowerCase();

console.log("La stringa in minuscolo è: " + stringaMinuscola);

Esercizio **5**/7 di Stringhe

Scrivi un programma che prenda una stringa, una sottostringa da cercare e una sottostringa da sostituire, utilizzando il metodo replace() e stampi il risultato sulla pagina web.

const frase = "La mia casa è blu";

const sottostringaDaCercare = "blu";

const sottostringaDaSostituire = "rossa";

const nuovaStringa = frase.replace(

sottostringaDaCercare,

sottostringaDaSostituire

);

document.write("La nuova frase è: " + nuovaStringa);

Esercizio **6**/7 di Stringhe

Scrivi un programma che prenda una stringa, la divida in un array di sottostringhe e stampi il risultato sulla console.

const testo = "Ciao,mondo,come,stai";

const arraySottostringhe = testo.split(",");

console.log("Array di sottostringhe: ", arraySottostringhe);

Esercizio **7**/7 di Stringhe

Scrivi un programma che prenda una stringa e utilizzando il metodo slice() estragga una parte della stringa in base agli indici di inizio e fine e stampi il risultato sulla console e sulla pagina web.

const saluta = "Buongiorno";

const parteStringa = saluta.slice(3, 7);

console.log("La parte estratta della stringa è: " + parteStringa);

document.write("La parte estratta della stringa è: " + parteStringa);

[Inizio](#Inizio) Oggetti letterali:

creare un oggetto in JavaScript utilizzando gli Oggetti letterali

let persona = {

 nome:'Gianluca',

 cognome:'Mondelli',

 anni:17

 };

Elencando le proprietà con i rispettivi valori, mediante una sequenza di coppie nome:valore separate da virgole racchiuse tra parentesi graffe.

document.write("Il nome del socio è: ");

document.write(persona.nome);

Dopo aver creato l’oggetto possiamo accedere alle sue proprietà utilizzando la notazione puntata (dot notation).

**Tutte le proprietà di un oggetto sono pubbliche e accessibili.**

Quindi, ogni proprietà ha un nome seguito dai due punti : e un valore iniziale.

document.write("<br>");

document.write("Il cognome del socio è: ");

document.write(persona['cognome']);

Oppure si può usare la notazione con le parentesi quadre

[Inizio](#Inizio) Oggetti con diverse proprietà:

Gli oggetti nascono proprio per raggruppare in modo arbitrario una collezione di proprietà che possiamo definire a piacimento:

 abbiamo attinto a svariati tipi di proprietà per rappresentare una ideale partita di calcetto

let partita = {

 giorno: 'lunedì',

una proprietà che indica il giorno della settimana (come stringa)

 giocatori: ["luca", "gabriella", "matteo", "rosa"],

un’altra proprietà giocatori con un array con i nomi dei giocatori

 partitaPersa: false,

un Boolean con lo stato della vittoria

 golFatti: 2,

 golSubiti: 0

due interi con gol fatti e subiti

 };

Quello visto in questo esempio, riguarda la creazione di un oggetto letterale, cioè quando lo definiamo mentre lo creiamo

alert("Abbiamo perso la partita? " + (partita.partitaPersa ? 'Si' : 'No'));

alert("Risultato finale: " + partita.golFatti + " a " + partita.golSubiti);

**Operatore ternario:**

La sintassi dell'operatore ternario è la seguente:

**condizione ? valore\_se\_vero : valore\_se\_falso;**

**Esempio 1 di utilizzo di Operatore ternario:**

Valutare il voto dell’esame di maturità, ottenendo la stringa “Promosso” se risulta essere >=60 altrimenti “Bocciato”

let voto = 75;

let risultato = (voto >= 60) ? "Promosso" : "Bocciato";

document.write(risultato);

**Esempio 2 di utilizzo di Operatore ternario:**

Se il voto è maggiore o uguale a 90, viene assegnata la stringa "Eccellente". Se il voto è compreso tra 80 e 89, viene assegnata la stringa "Buono". Se il voto è compreso tra 60 e 79, viene assegnata la stringa "Sufficiente". Infine, se il voto è inferiore a 60, viene assegnata la stringa "Insufficiente".

let voto2 = 75;

let risultato2 = (voto2 >= 90) ? "Eccellente" : (voto2 >= 80) ? "Buono" : (voto2 >= 60) ? "Sufficiente" : "Insufficiente";

document.write(risultato2);

*Esercizio* ***1****/5 di utilizzo di Operatore ternario:*

Scrivi un codice che prenda un numero come input e verifichi se è pari o dispari utilizzando l'operatore ternario.

const numero = 10;

const risultato3 = numero % 2 === 0 ? "pari" : "dispari";

document.write("Il numero è:", risultato3);

Esercizio **2**/5 di utilizzo di Operatore ternario:

Scrivi un codice che prenda una stringa come input e verifichi se è vuota o non vuota utilizzando l'operatore ternario.

const stringa = "";

const risultato4 = stringa.length === 0 ? "vuota" : "non vuota";

document.write("La stringa è:", risultato4);

Differenza tra **==** e **===**

Il doppio uguale (==) verifica soltanto l'uguaglianza dei valori.

Il (===) verifica se le variabili confrontate hanno lo stesso valore e sono dello stesso tipo.

*Quando dovresti usare == e quando ===?*

Se hai dubbi, utilizza **===**. Ti eviterà un sacco di potenziali bug.

Esercizio **3**/5 di utilizzo di Operatore ternario:

Scrivi un codice che prenda un numero come input e verifichi se è maggiore, minore o uguale a zero utilizzando l'operatore ternario.

const numero2 = 5;

const risultato5 =

numero2 > 0

? "maggiore di zero"

: numero2 < 0

? "minore di zero"

: "uguale a zero";

document.write("Il numero è:", risultato5);

Esercizio **4**/5 di utilizzo di Operatore ternario:

Scrivi un codice che prenda due numeri come input e verifichi se il primo numero è divisibile per il secondo numero utilizzando l'operatore ternario.

const num1 = 10;

const num2 = 2;

const risultato6 = num1 % num2 === 0 ? "divisibile" : "non divisibile";

document.write("Il primo numero è:", risultato6);

Esercizio **5**/5 di utilizzo di Operatore ternario:

Scrivi un codice che prenda un numero e un intervallo come input e verifichi se il numero è compreso nell'intervallo utilizzando l'operatore ternario.

const numero3 = 15;

const intervalloMinimo = 10;

const intervalloMassimo = 20;

const risultato7 =

numero >= intervalloMinimo && numero3 <= intervalloMassimo

? "compreso"

: "non compreso";

document.write("Il numero è:", risultato7);

https://codegrind.it/esercizi/javascript/lavorare-con-stringhe