Simulazione Test Javascript

Docente: Riccardo Degni

Nota 1: attenersi alle tracce, e leggere attentamente i dettagli di ogni requisito
Nota 2: utilizzare unicamente i costrutti visti a lezione
Nota 3: è possibile consultare ogni fonte di documentazione online

Level: EASY (1pt per esercizio)

Esercizio 1: Letterali oggetti

Scrivi una funzione che prenda in input un array di 6 oggetti rappresentanti persone (nome, cognome, età) e restituisca un nuovo array contenente solo le persone con un'età superiore a 30 anni. Stampare le informazioni di tutte le persone del nuovo array.

Esercizio 2: Map

Scrivi una funzione chiamata moltiplicaArray che prenda in input un array di numeri e un numero intero, che funge da "fattore di moltiplicazione". La funzione deve restituire un nuovo array in cui ogni numero dell'array originale è moltiplicato per il fattore di moltiplicazione dato.
Chiamare la funzione 3 volte, con 3 array e 3 fattori di moltiplicazione differenti per stampare in output i 3 array prodotti.

Esercizio 3: Set

Scrivi una funzione chiamata unisciSet che prenda in input due set composti da 5 numeri ciascuno, e restituisca un nuovo set contenente tutti gli elementi del primo che NON sono presenti nel secondo.
Stampare in output il set risultante.

Esercizio 4: Funzioni

Crea una funzione chiamata salutaPersona che prende in ingresso un oggetto.
Se l'oggetto passato come parametro NON ha una proprietà chiamata "rate", la funzione stampa il messaggio "nessun rate". In caso contrario, se il valore della proprietà rate è compreso tra 0 e 10, stampa "rate basso", se è compreso tra 11 e 50 stampa "rate medio", se è compreso tra 51 e 100 stampa "rate alto".

es
salutaPersona({nome: mario rossi}) -> nessun rate
salutaPersona(rate: 95) -> rate alto

Level: MEDIUM (2pt per esercizio)

Esercizio 1: Date

Scrivi una funzione chiamata calcolaGiorni che prenda in input due oggetti Date rappresentanti due date diverse e restituisca il numero di giorni trascorsi tra le due date.
Chiamare la funzione per stampare in output il risultato.
La funzione deve prendere in considerazione questa caratteristica: se la seconda data è un giorno cronologicamente pari o successivo alla prima data, si procede con la normale esecuzione sopra elencata.
Se la seconda data è un giorno cronologicamente precedente alla prima data, viene prodotto manualmente un errore che blocca l'applicazione.

Esercizio 2: Letterali oggetti

Scrivi una funzione confrontaOggetti che prenda in input due oggetti con caratteristiche (proprietà e valori) scelti a piacere e restituisca true se hanno le stesse proprietà e se tutti i valori delle loro proprietà corrispondenti sono uguali, altrimenti false.

es
o1 = {nome: mario, cognome = rossi}
o2 = {nome: mario, cognome = rossi}
confrontaOggetti(o1, o2) da true

o1 = {nome: mario, cognome = rossi}
o2 = {x: 10, y = 20}
confrontaOggetti(o1, o2) da false

Esercizio 3: Funzioni come parametri e metodo map

Scrivi una funzione chiamata mappaNumeri, che prende come primo parametro un array di numeri, e come secondo parametro una funzione che internamente a mappaNumeri sarà utilizzata come parametro per il metodo map, chiamato per creare un nuovo array con il criterio utilizzato dalla funzione passata. Questo array verrà stampato internamente a mappaNumeri.
Chiamare la funzione mappaNumeri 2 volte, sia per stampare il doppio dei numeri dell'array passato come primo parametro, sia per stampare i suoi numeri aumentati di 10.

es
mappaNumeri([10, 20, 30], ?fn?) stampa 20 40 60
mappaNumeri([10, 20, 30], ?fn?) stampa 20 30 40

dove ?fn? saranno le apposite funzioni passate come parametro create da voi

Level: HARD (4pt per esercizio)

Esercizio 1

Dato il seguente JSON:

convertirlo in una controparte Javascript da utilizzare nel programma.
Creare una funzione chiamata dammiPersonaggio, che prende in ingresso come primo parametro un array di personaggi come quello prodotto dal JSON.
Come secondo parametro, prende una stringa.
Se la stringa è "cicla", la funzione stampa le informazioni di tutti i personaggi.
Se la stringa è "giovane", la funzione stampa le informazioni del personaggio più giovane.
Se la stringa è "migliore", la funzione stampa le informazioni del personaggio con la media rate più alta.
Se la stringa è "peggiore", la funzione stampa le informazioni del personaggio con la media rate più bassa.
Se il secondo parametro non è una stringa, generare un errore manualmente.

Chiamare la funzione dammiPersonaggio con tutti i test case, impedendo che l'applicazioni si blocchi se non viene passata una stringa (hint: usare un try/catch).

Quando si stampano le informazioni di un personaggio, non bisogna stampare la sua data di nascita, ma la sua età (es 30 anni).
Solo per i rate tenere in considerazione eventuali "parità" e stampare tutti i personaggi con il rate migliore o peggiore in caso di concomitanze.