

Esame Modulo JavaScript

 Subject

Esercizio 1: Variabili e Stringhe (10 punti)

Crea un programma JavaScript che:

1. Chieda all'utente il suo nome e lo memorizzi in una variabile. **(2.5 punti)**
2. Chieda all'utente il suo anno di nascita e lo memorizzi in un'altra variabile. **(2.5 punti)**
3. Visualizzi un messaggio che saluta l'utente con il suo nome e gli dice se è nato in un anno bisestile. **(5 punti)**



Un anno è bisestile se:

- è divisibile per 4 e allo stesso tempo non è divisibile per 100;
- oppure
- se l'anno è divisibile per 400;

Esercizio 2: Cicli e Condizioni (15 punti)

Scrivi una funzione JavaScript che:

1. Accetta un array di stringhe come argomento.
2. Calcola la concatenazione di tutte le stringhe che iniziano con la lettera "A" o "a". **(10 punti)**
3. Se la lunghezza della stringa concatenata supera i 10 caratteri, troncala a 10 caratteri e restituiscila. **(5 punti)**
4. Se il ciclo si completa senza che la stringa superi i 10 caratteri, restituisci la stringa concatenata.

Esercizio 3: Array e Oggetti (20 punti)

Dato il seguente oggetto che rappresenta un libro, esegui i compiti sottostanti:

```
const book = {
  title: "Il Grande Gatsby",
  author: "F. Scott Fitzgerald",
  year: 1925,
  rating: 8.5
};
```

1. Aggiungi una nuova proprietà `rating` all'oggetto libro, che rappresenta il rating del libro, e inizializzalo a `6`.
2. Aggiungi un metodo `increaseRating` che accetta come argomento un valore numerico e che incrementa il rating del libro del valore passato alla funzione. **(10 punti)**
Il nuovo rating risultante, non può superare 10. **(2.5 punti)**
3. Rimuovi la proprietà `year` dall'oggetto libro. **(2.5 punti)**
4. Stampa in console ogni proprietà dell'oggetto libro in questo formato: `"[chiave]: [valore]"`. **(5 punti)**

Esercizio 4: DOM e Eventi (20 punti)

Crea una pagina HTML con un `<input>` di testo, un elemento `<button>`, un elemento `<p>` con id `"errorMessage"` e un elemento `` con id `"tasks"`.

Scrivi codice JavaScript che aggiunge un ascoltatore di eventi al `<button>` che, quando cliccato, se l'utente non ha scritto nulla nell' `<input>`:

- inserisce all'interno del `<p>` `"errorMessage"` un messaggio di errore che dice "Per favore, inserisci del testo."; **(5 punti)**

Altrimenti, se c'è del testo all'interno dell' `<input>`:

- svuota il contenuto del `<p>` `"errorMessage"`; **(5 punti)**
- inserisce all'interno del `` `"tasks"` un nuovo `` che ha come contenuto ciò che è stato inserito dall'utente nell' `<input>`; **(10 punti)**

Esercizio 5: Error Handling e Parsing (20 punti)

Scrivi una funzione `power` per elevare un numero ad una potenza data:

1. Prende due numeri come parametri (`base` e `exponent`).
2. Controlla che effettivamente siano stati passati due numeri. In caso contrario, lancia un errore che mostra nel messaggio quale dei parametri non è un

numero. Se entrambi non sono dei numeri, il messaggio d'errore li mostra entrambi. **(15 punti)**

3. Nel caso in cui sia avvenuto un errore, la funzione ritorna `null`, altrimenti ritorna `base` elevato alla `exponent`. **(2.5 punti)**
4. Indipendentemente dal successo o dal fallimento, stampa in console un messaggio che indica che l'operazione è completa. **(2.5 punti)**

Esercizio 6: Clonazione di Oggetti (15 punti)

Dato il seguente oggetto annidato che rappresenta un team di sviluppatori, esegui i compiti sottostanti:

```
const devTeam = {
  lead: {
    name: "Alice",
    skills: ["JavaScript", "React", "Node.js"]
  },
  testers: [
    { name: "Bob", skills: ["HTML", "CSS", "JavaScript"] },
    { name: "Charlie", skills: ["Python", "Django", "PostgreSQL"] }
  ],
  projectName: "WebApp"
};
```

1. In un commento, scrivi quanti oggetti l'espressione soprastante ha inserito in memoria ed elencali. **(2.5 punti)**
2. Crea una variabile chiamata `teamLead` che faccia riferimento alla proprietà `lead` dell'oggetto `devTeam`. Aggiungi "GraphQL" alle `skills` di `teamLead`.
3. Crea una copia dell'oggetto `devTeam` e memorizzala in una nuova variabile chiamata `devTeamCopy`. Non puoi usare la funzione integrata `structuredClone`. Poi, cambia la proprietà `projectName` di `devTeam2` in "MobileApp". **(2.5 punti)**
`devTeamCopy`
4. Costruisci una funzione che crea e restituisce un oggetto `tester` a partire dal parametro `name` (stringa). Durante la creazione dell'oggetto, la funzione dovrebbe anche assegnare un array `skills` con tre stringhe casuali tra "JavaScript", "React", "Node.js", "HTML", "CSS", "Python", "Django", "PostgreSQL". **(5 punti)**
Usa questa funzione per aggiungere un nuovo tester a `devTeamCopy`. **(2.5 punti)**
5. In un commento, scrivi quanti oggetti sono stati creati in memoria oltre a quelli iniziali, e giustifica la tua risposta. **(2.5 punti)**

