**React Q&A**

**Cos’è ReactJS? (risposta aperta)**

*ReactJS è una libreria JavaScript per la creazione di interfacce utente interattive e dinamiche.*

*È ampiamente utilizzato nello sviluppo web per la creazione di applicazioni a pagina singola (Single-Page Applications) e componenti riutilizzabili.*

*ReactJS è focalizzato sulla creazione di UI modulari e ben organizzate, dove i componenti possono essere composti insieme per formare l'intera interfaccia.*

*Utilizza un approccio chiamato "DOM virtuale" per ottimizzare le prestazioni e rendere l'aggiornamento degli elementi dell'interfaccia più efficiente.*

*ReactJS è stato sviluppato da Facebook ed è ora supportato da una vasta comunità di sviluppatori.*

**ReactJS è un framework, non una libreria. (v/f)**

**FALSO**

**Il file package.json contiene…**

* Informazioni non essenziali, può essere tranquillamente eliminato
* Solamente il nome e la versione dell’applicazione
* Molte informazioni utili, come ad esempio l’elenco di tutte le dipendenze richieste dall’applicazione
* Il primo componente React caricato all’avvio

**create-react-app è l’unico modo possibile per creare un’applicazione**

**React (v/f)**

**VERO**

**Qual è il comando da lanciare nel terminale per creare una nuova create-react-app con nome “test”?**

* npx create-react-app test
* npm create-react-app test
* npx create-react-app-test
* npx create-new-react-app test

**Un componente React può venire creato in tre modi: come funzione,**

**classe o interfaccia. (v/f)**

**FALSO**

**Cos’è un componente React?**

* Una libreria che è necessario importare dentro l’applicazione
* Una pagina ben specifica all’interno dell’applicazione
* Un blocco di logica/contenuto riutilizzabile all’interno dell’applicazione
* Una landing page per l’applicazione

**Qual è la differenza tra componenti creati come funzione e componenti**

**creati come classe?**

* I componenti creati come funzione sono più semplici da scrivere ma con meno funzionalità, mentre i componenti creati come classe necessitano in genere di un maggior numero di linee di codice ma possiedono più funzionalità.
* I componenti creati come classe sono più semplici da scrivere ma con meno funzionalità, mentre i componenti creati come funzione necessitano in genere di un maggior numero di linee di codice ma possiedono più funzionalità.

**Le props sono frammenti di informazione assegnati all’invocazione di un componente React, utili al fine di rendere il componente dinamico e più riutilizzabile. (v/f)**  
 **VERO**

**Le props possono essere passate solamente da un componente genitore ad un componente figlio, non è possibile fare il contrario (v/f)**

**VERO**

**Da dove possono venire recuperate le props all’interno di un componente React creato come classe?**

* Sono proprietà contenute in un oggetto accessibile attraverso i parametri della funzione
* **Possono essere recuperate all’interno dello oggetto ‘this’, dentro un sotto-oggetto chiamato ‘props’**
* Vengono applicate come proprietà dell’oggetto globale ‘window’
* Vengono applicate come proprietà dell’oggetto ‘document’