**1. Czym jest useState, useEffect oraz useContext?**

**useState**

const [state, setState] = useState(initialState);

Zwraca zmienną przechowującą lokalny stan i funkcję do jego aktualizacji.

Podczas pierwszego renderowania zwrócona wartość stanu (state) jest taka sama, jak wartość przekazana jako pierwszy argument (initialState).

**useEffect**

Jeden z hooków, który pozwala na tworzenie stanowych komponentów funkcyjnych, a także na wykonywanie efektów ubocznych w funkcjach..

**useContext**

### useContext

const value = useContext(MyContext);

Przyjmuje obiekt kontekstu (wartość zwróconą przez React.createContext) i zwraca jego aktualną wartość. Wartość kontekstu jest określana przez właściwość (ang. prop) value najbliższego rodzica <MyContext.Provider> wywołującego komponentu.

**2. Krótko opisz czym są komponenty, podaj ich zalety i przykładowe zastosowanie**

Komponenty dzieli się na klasowe i funkcyjne.

Komponent funkcyjny składa się z javascriptowej funkcji przyjmującej reactowe argumenty.

function Welcome(parametr) {

return <h1>Cześć, {parametr.name}</h1>;

}

Komponenty klasowe – bazują na javascriptowej klasie odwołuje się do jej parametrów

class Welcome extends React.Component {

render() {

return <h1>Cześć, {this.props.name}</h1>;

}

}

Obydwa powyższe komponenty są równoważne z punktu widzenia Reacta.

**Zaletą komponentów jest** to, że można podzielić interfejs użytkownika na niezależne, pozwalające na ponowne użycie części i myśleć o każdej z nich osobno.

**3. Czym jest Server Side Rendering i czym różni się od Client Side Rendering?**

**Server Side Rendering**

Renderowanie po stronie serwera (SSR) to zdolność aplikacji do konwertowania plików HTML na serwerze na w pełni renderowaną stronę HTML dla klienta. Przeglądarka internetowa wysyła żądanie informacji z serwera, który natychmiast odpowiada wysyłając w pełni wyrenderowaną stronę do klienta.

**Client Side Rendering**

Renderowanie po stronie klienta pozwala programistom na pełne renderowanie ich stron internetowych w przeglądarce za pomocą JavaScript. Zamiast mieć inną stronę HTML na trasę, witryna renderowana po stronie klienta tworzy każdą trasę dynamicznie bezpośrednio w przeglądarce. To podejście rozprzestrzeniło się, gdy frameworki JS sprawiły, że stało się to łatwe.

**4. W jaki sposób można odczytać parametry przesłane przez metodę GET?**

**5. Podaj krótki przykład rekurencji oraz jak ją "opuścić".**

Funkcja rekurencyjna to funkcja, która wywołuje samą siebie, dopóki nie zostanie przerwana.

Funkcja rekurencyjna zawsze ma warunek, który uniemożliwia funkcji wywołanie samej siebie w pewnym momencie.

Pokazuje to funkcja zliczająca sumę liczb a następnie wyliczająca resztę z dzielenia przez 10.

Na przykład suma liczb 324.

Poniższy kod ilustruje funkcję rekurencyjną sumOfDigits():

Obraz zawierający tekst

Opis wygenerowany automatycznie

num%10 Zwraca resztę z dzielenia num przez 10, na przykład, w ten sposób  
324 / 10 = 32 reszta **4** 324 % 10 = **4**

W Math.floor(num/10)zawraca całkowita część num podzielona przez 10: przenoszenie jest częścią całkowitą. 324 / 10 = **32,4032**

**Jest if(num==0) to warunek, który przerywa wywołanie funkcji.**

**6. Czym jest pamięć lokalna strony i w jaki sposób można ją wykorzystać.**

Z lokalnej pamięci masowej, aplikacje internetowe mogą przechowywać dane lokalnie w przeglądarce użytkownika.

Przed HTML5, dane aplikacji musiały być przechowywane w cookies , zawarte w każde żądanie serwera. Przechowywanie lokalne jest bardziej bezpieczny, a duże ilości danych mogą być przechowywane lokalnie, bez wpływu na wydajność strony internetowej.

W przeciwieństwie do cookies , limit pamięci jest znacznie większa (at least 5MB) i informacje nigdy nie są przesyłane do serwera.

Przechowywanie lokalne jest za pochodzenie (per domain and protocol) . Wszystkie strony, z jednego pochodzenia, mogą przechowywać i dostęp do tych samych danych.

**7. Krótko opisz różnice pomiędzy flex i grid.**

**CSS Grid Layout** jest najlepszą metodą aby podzielić strone na części lub definiowanie relacji takich jak rozmiar, pozycja i warstwa, pomiędzy podstawowymi elementami HTML.

Podobnie do tabel (ang. "tables"), rozmieszczenie grid (ang. "siatka") pozwala autorowi wyrównać elementy w kolumny i wiersze. Jednakowoż grid pozwala na wiele więcej ułożeń niż tabele. Na przykład kontener grid potrafi spozycjonować elementy wewnątrz siebie w taki sposób aby na siebie nachodziły oraz używały warstw, podobnie jak elementy pozycjonowane przy użyciu CSS.

**CSS Flexbox** wykorzystuje narzędzia, które pozwalają na szybkie tworzenie kompleksowego i elastycznego układu strony. Flexbox to jednowymiarowa metoda rozmieszczania elementów w wierszach lub kolumnach, przedmioty rozciągają się aby wypełnić dodatkową przestrzeń lub kurczą się aby dopasować się do mniejszych przestrzeni.