🚀 Next.js Interview Questions and Answers (1–50)

1. What is Next.js?

Answer: Next.js is a React-based framework for building server-side rendered (SSR) and statically generated (SSG) web applications.

2. What are the key features of Next.js?

Answer:

- Server-Side Rendering (SSR)
- Static Site Generation (SSG)
- File-based routing
- API routes
- Image optimization
- Built-in CSS support

3. What is the difference between React and Next.js?

Answer: React is a UI library. Next.js is a full-fledged framework built on top of React with routing, SSR, and SSG.

4. How do you create a new Next.js app?

Answer:

```
npx create-next-app my-app
```

5. What is file-based routing in Next.js?

Answer: Pages are automatically routed based on their file path in the `pages` directory.

6. What is SSR in Next.js?

Answer: Server-Side Rendering — the HTML is generated on each request at runtime.

7. What is SSG in Next.js?

Answer: Static Site Generation — the HTML is generated at build time and reused for each request.

8. What is CSR in Next.js?

Answer: Client-Side Rendering — the page is rendered in the browser after fetching necessary data.

9. How do you implement SSR in Next.js?

Answer: Using `getServerSideProps()` in a page component.

10. How do you implement SSG in Next.js?

Answer: Using `getStaticProps()` and optionally `getStaticPaths()`.

11. What is `getStaticProps()`?

Answer: A function to fetch data at build time for static generation.

12. What is `getServerSideProps()`?

Answer: A function to fetch data on every request (SSR).

13. What is `getStaticPaths()` used for?

Answer: To generate dynamic routes for SSG when using `getStaticProps()`.

14. What is the `pages` directory in Next.js?

Answer: Contains the route-based components — each file is a route.

15. How do you create a dynamic route in Next.js?

Answer: Use square brackets: `pages/blog/[id].js`

16. How do you access route parameters?

Answer: Via `useRouter()` hook or as a parameter in data fetching functions.

17. How do you add navigation in Next.js?

Answer: Using `<Link>` from `next/link`.

18. What is prefetching in Next.js?

Answer: Next.js automatically prefetches pages linked with `<Link>` when in the viewport.

19. How do you create API routes in Next.js?

Answer: Inside the `pages/api` directory, each file exports a handler.

20. What is the `public` directory in Next.js?

Answer: Static assets like images can be placed here and accessed directly.

21. How do you import images in Next.js?

Answer: Using the `<Image />` component from `next/image`.

22. What are the benefits of using `next/image`?

Answer: Optimized image loading, lazy loading, and automatic resizing.

23. What is Incremental Static Regeneration (ISR)?

Answer: Rebuilds static pages in the background after deployment using `revalidate` in `getStaticProps`.

24. How do you enable ISR?

Answer:

```
export async function getStaticProps() {
  return {
    props: {},
    revalidate: 60, // seconds
  };
}
```

25. What is middleware in Next.js?

Answer: Runs before request processing — for redirects, auth, etc.

26. What is `_app.js`?

Answer: A custom App component to initialize pages and add global styles.

27. What is `_document.js`?

Answer: Customizes the HTML structure rendered by Next.js (not for UI logic).

28. What is `next.config.js`?

Answer: Configuration file for customizing build behavior and environment.

29. How do you add global CSS in Next.js?

Answer: Import CSS in `_app.js`:

```
import '../styles/globals.css';
```

30. How do you use SCSS or Sass in Next.js?

Answer: Install `sass` and use `.scss` files.

31. What is the default port of Next.js dev server?

Answer: Port 3000.

32. How do you fetch data in Next.js client-side?

Answer: Use `useEffect` with `fetch` or `axios`.

33. What is the difference between `_app.js` and `_document.js`?

Answer:

- `_app.js`: Controls page initialization and state
- `_document.js`: Customizes HTML structure

34. How do you redirect in Next.js?

Answer: Use `redirect` in `getServerSideProps` or middleware.

35. How do you handle 404 pages?

Answer: Create `pages/404.js`.

36. How do you handle 500 errors?

Answer: Create `pages/500.js`.

37. What are environment variables in Next.js?

Answer: Stored in `.env.local` and accessed via `process.env.VAR_NAME`.

38. How do you deploy a Next.js app?

Answer: Use Vercel (recommended), Netlify, or custom servers.

39. What is Vercel?

Answer: The deployment platform for Next.js maintained by its creators.

40. How do you export a static site?

Answer: Run `next export` to generate a static HTML site.

41. What is `next/head` used for?

Answer: To manage `<head>` elements like title and meta tags.

42. What is the benefit of SSR?

Answer: Better SEO, faster first paint for dynamic content.

43. What is the benefit of SSG?

Answer: Super fast page loads, can be served via CDN.

44. What are the drawbacks of SSR?

Answer: Slower response time than SSG; more server load.

45. How to use cookies or auth in Next.js?

Answer: Use API routes or middleware to read/write cookies.

46. How does Next.js support TypeScript?

Answer: Out of the box — run `npx create-next-app --typescript`.

47. What is `fallback: true` in `getStaticPaths`?

Answer: Allows new paths to be generated on the fly when not pre-rendered.

48. What is client-side navigation?

Answer: Navigating between pages without full-page reload using Next.js' built-in router.

49. Can you use Redux with Next.js?

Answer: Yes, with proper setup using providers in `_app.js`.

50. What is the recommended folder structure in Next.js?

Answer:

- `/pages`
- `/components`
- `/public`
- `/styles`
- `/lib` or `/utils` for reusable logic