* **Explain how the Navigator widget works in Flutter**.
* **Navigator manages a stack of routes in Flutter. Each screen is represented by a Route**. **When you move to a new screen, the route is pushed** **onto the stack, and when you go back, the current route is popped off the stack. The most common route type is MaterialPageRoute, which handles screen transition animations. Navigation methods require the current BuildContext to locate the nearest Navigator.**
* **Push:** **Opens a new screen by adding a route on top of the stack.**
* **Pop: Closes the current screen and returns to the previous route.**
* **Push Replacement:** **Replaces the current route with a new route.**
* **PushAndRemoveUntil:** **Pushes a new route and removes routes until a certain condition is met.**
* **Describe the concept of named routes and their advantages over direct route navigation**

**Named routes: Named routes allow you to define routes with a** **unique string identifier** **(name) in your app, instead of directly using a** **MaterialPageRoute every time. You register these routes in the MaterialApp widget using the routes property:**

**Advantage: -**

* **Centralized route management: All routes are defined in one place, making the code cleaner and easier to maintain.**
* **Reusability: The same named route can be used anywhere in the app without rewriting MaterialPageRoute.**
* **Easier to read & understand: Navigation calls are shorter and more semantic ('/profile' instead of creating a new route object each time).**
* **Supports dynamic arguments: You can pass data via arguments without modifying the route creation logic.**
* **Explain how data can be passed between screens using route arguments.**
* **In Flutter, data can be passed between screens using** **route arguments** **by utilizing the arguments property of the Navigator.pushNamed() method. When navigating to a new route, the required data is sent as an argument.**
* **In the destination screen, the data can be accessed using the ModalRoute.of(context)!.settings.Arguments property. This approach allows information to be transferred between screens in a clean and organized way while keeping navigation well-structured, especially when using** **named routes**.