

- [Dart Var vs Dynamic - Difference | Flutter Dart Tutorial 🔥](#)
 - [1. Introduction to var and dynamic](#)
 - [2. Understanding var](#)
 - [3. Understanding dynamic](#)
 - [4. Comparison Between var and dynamic](#)
 - [5. Practical Example in Flutter](#)
 - [6. Key Takeaways](#)

Dart Var vs Dynamic - Difference | Flutter Dart Tutorial 🔥

In this tutorial, we'll explore the differences between the `var` and `dynamic` keywords in Dart. Understanding these differences is crucial for effective Flutter development, as it helps you manage variable declarations and data types efficiently. Let's break down the key points, supported by examples and code snippets.

1. Introduction to `var` and `dynamic`

- **Var** and **Dynamic** are two ways to declare variables in Dart.
- Both are used when you don't want to explicitly specify the type of a variable.
- The difference lies in how they infer and manage data types.

2. Understanding `var`

- **Implicit Type Inference:** When you declare a variable using `var`, Dart infers the type based on the first value assigned to the variable.
- **Type-Safe:** Once the type is inferred, the variable can only hold values of that type. If you try to assign a value of a different type, it will cause a compile-time error.

```
void main() {  
  var name = "Raman"; // Dart infers that 'name' is of type String  
  // name = 42;        // This will cause an error because 'name' is inferred as  
String  
  
  var rollNumber = 11; // Dart infers that 'rollNumber' is of type int
```

```
// rollNumber = "eleven"; // This will cause an error because 'rollNumber' is
inferred as int

print(name);           // Output: Raman
print(rollNumber);     // Output: 11
}
```

3. Understanding **dynamic**

- **No Type Safety:** A variable declared as **dynamic** can hold any type of value, and you can change the type of the value it holds at runtime without any errors.
- **Use Cases:** **dynamic** is useful when you're dealing with data whose type might change or is not known at compile time, such as data from an API or user input.

```
void main() {
    dynamic section = "C"; // Initially a String
    print(section);       // Output: C

    section = 42;         // Now an int
    print(section);       // Output: 42

    section = true;       // Now a bool
    print(section);       // Output: true
}
```

4. Comparison Between **var** and **dynamic**

- **Type Inference:**
 - **Var:** Type is inferred from the first assigned value and cannot change.
 - **Dynamic:** No type inference; the variable can hold values of any type and can change types at runtime.
- **Compile-Time Safety:**
 - **Var:** Type-safe, errors are caught at compile time.
 - **Dynamic:** No compile-time type checking, potential errors are caught at runtime.
- **Use Cases:**
 - **Var:** Use when you know the type of the variable and it won't change.

- **Dynamic:** Use when the type might change, or you don't know the type beforehand.

5. Practical Example in Flutter

Consider a scenario where you're fetching data from an API. You might not know the exact structure or types of the data, so you could use **dynamic** to handle various possible types.

```
void main() {  
  dynamic response = fetchApiResponse(); // Assume this function fetches data  
  from an API  
  
  if (response is String) {  
    print("Received a string: $response");  
  } else if (response is int) {  
    print("Received an integer: $response");  
  } else {  
    print("Received unknown type of data");  
  }  
}  
  
dynamic fetchApiResponse() {  
  // Simulate different types of responses  
  return "API response as a string";  
  // return 404; // Uncomment this to simulate an integer response  
}
```

In this example, the **response** variable can hold different types of data depending on what the API returns, making **dynamic** a suitable choice.

6. Key Takeaways

- **Var** is type-safe and should be used when the variable type is known and consistent.
- **Dynamic** is flexible and should be used when the variable type is unknown or can change.
- Use **Var** for predictable and consistent data types, and **Dynamic** for handling unpredictable or varying types.

By understanding and correctly using **var** and **dynamic**, you can write more robust and adaptable Dart code, particularly in Flutter development.