Final vs Const Keywords - Difference and How to Use? | Flutter Dart Tutorial

In this article, we'll dive into the difference between the `final` and `const` keywords in Dart, a language used heavily in Flutter development. Understanding these keywords is crucial for effective runtime programming and optimizing your Flutter projects.

1. Understanding `final` Keyword

Key Points:

- **Single Assignment**: A variable declared with `final` can only be assigned once.
- **Runtime Initialization**: The value of a `final` variable is determined at runtime, meaning it can be initialized later in the code, not necessarily at compile time.
- **Non-reassignable**: Once a `final` variable is assigned a value, it cannot be reassigned a new value.

Example:

```
void main() {
  final String name = 'Raman';

  // Trying to reassign a final variable will cause an error
  // name = 'Ramanujan';  // Error: 'name' can only be set once.

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

In this example, the `name` variable is declared as `final` and assigned the value 'Raman'. If you try to assign a new value to `name`, it will throw an error because `final` variables cannot be reassigned after their initial assignment.

2. Understanding `const` Keyword

Key Points:

- **Compile-time Constant**: A `const` variable is a compile-time constant. This means its value must be known at compile time and cannot change at runtime.
- **Mandatory Initialization**: When using `const`, the variable must be initialized with a value at the time of declaration.
- **Immutable**: Variables declared with `const` cannot be modified in any way once they are assigned.

Example:

```
void main() {
  const String name = 'Raman';

  // Trying to reassign or modify a const variable will cause an error
  // name = 'Ramanujan';  // Error: Constant variables can't be assigned a value.

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

Here, `name` is a `const` variable, meaning it is a constant value that cannot be changed after the initial assignment.

3. Differences Between `final` and `const`

Point-wise Differences:

- **Initialization Timing**:
  - `final`: Value is assigned once, and this assignment can occur at runtime.
  - `const`: Value is assigned at compile-time, and it must be initialized at the time of declaration.
- **Reassignment**:
  - `final`: The variable can only be assigned once, but the assignment can occur later in the code.
  - `const`: The variable is immutable and cannot be reassigned or modified after declaration.
- **Runtime Behavior**:
  - `final`: Used for values that are determined at runtime but should not change afterward.
  - `const`: Used for values that are known at compile-time and remain constant throughout the program.

4. Practical Example: List with `final` vs `const`

Using `final`:

```
void main() {
  final List<String> names = ['Raman', 'Ramanujan', 'Peter'];

  // You can modify the contents of the list
  names.add('Alice');

  // But you cannot reassign the entire list
  // names = ['New', 'List'];  // Error: Final variables cannot be reassigned.
```

```
  print(names);  // Output: [Raman, Ramanujan, Peter, Alice]
}
```

With `final`, while the variable `names` cannot be reassigned to a different list, you can still modify the contents of the list.

Using `const`:

```
void main() {
  const List<String> names = ['Raman', 'Ramanujan', 'Peter'];

  // Attempting to modify the contents of a const list will cause an error
  // names.add('Alice');  // Error: Cannot add to an unmodifiable list.

  print(names);  // Output: [Raman, Ramanujan, Peter]
}
```

With `const`, the entire list and its contents are immutable. You cannot add, remove, or change any elements once the list is created.

5. Conclusion

Understanding when to use `final` and `const` is essential for efficient Dart programming, especially in Flutter. Use `final` when you have a value that is determined at runtime and should not be reassigned. Use `const` for values that are known at compile-time and should remain constant throughout the life of the program.

In practice, you'll often use `final` in situations where the value is dynamic but immutable after initialization, and `const` for true constants, such as mathematical constants or fixed configuration values.

By correctly using these keywords, you can write more predictable and reliable Flutter code.