Important Methods of Dart Data Types – Quick Guide

This short guide covers the most commonly used **methods of Dart data types** with examples, followed by **brainstorming** and **guess-the-output** questions.

---

1. String Methods

Strings are used to store text.

Common Methods

```dart
String text = "Dart Programming";

text.length;            // Number of characters
text.toUpperCase();    // Convert to uppercase
text.toLowerCase();    // Convert to lowercase
text.contains("Dart"); // Check if substring exists
text.replaceAll("Dart", "Flutter");
text.trim();           // Remove extra spaces
```

String Interpolation (Very Important)

String interpolation allows you to **insert variables directly inside a string** using `$` or `${}`.

```dart
String name = "Alex";
int age = 20;

print("My name is $name");
print("I am $age years old");
print("Next year I will be ${age + 1}");
```

It makes strings **cleaner, more readable, and dynamic**.

---

2. int and double Methods

Used for numbers.

Common Methods

```
int a = 10;
double b = 3.7;

a.isEven;     // true or false
a.isOdd;      // true or false

a.abs();      // Absolute value

b.round();    // Round to nearest integer
b.floor();    // Round down
b.ceil();     // Round up
```

---

3. List Methods

Lists store multiple values.

Common Methods

```
List<int> numbers = [1, 2, 3];

numbers.add(4);
numbers.remove(2);
numbers.length;
numbers.contains(3);
numbers.first;
numbers.last;
```

---

4. Map Methods

Maps store key–value pairs.

Common Methods

```
Map<String, int> marks = {
  "Math": 90,
  "Science": 85
};

marks.keys;
marks.values;
marks.containsKey("Math");
marks.containsValue(100);
marks.remove("Science");
```

---

5. bool Methods

Booleans store true or false.

```
bool isLoggedIn = true;

isLoggedIn.toString();
```

Brainstorming Questions

1. Why do we need methods for data types instead of writing logic again and again?
2. How do string methods make text handling easier in real apps?
3. Why is `round()` useful in shopping or billing apps?
4. What would happen if lists did not have built-in methods like `add()` or `remove()`?
5. In which real-life scenarios are maps more useful than lists?

Guess the Output (Slightly More Challenging)

1. Guess the output:

   ```
   String name = " dart ";
   print(name.trim().toUpperCase());
   ```

2. Guess the output:

   ```
   int x = 7;
   print(x.isEven || x.isOdd);
   ```

3. Guess the output:

```
double num = 4.6;
print(num.round() + num.floor());
```

4. Guess the output:

```
List<int> nums = [1, 2, 3];
nums.add(4);
nums.remove(2);
print(nums.length);
```

5. Guess the output:

```
List<String> items = ["apple", "banana", "mango"];
print(items.first.toUpperCase());
```

6. Guess the output:

```
Map<String, int> data = {"A": 1, "B": 2};
data["C"] = 3;
data.remove("B");
print(data.keys);
```

7. Guess the output:

```
String text = "Flutter";
print(text.contains("Dart"));
```

8. Guess the output (String Interpolation):

```
String name = "Sam";
int marks = 85;
print("$name scored ${marks + 5}");
```

9. Guess the output (String Interpolation):

```
int a = 5;
int b = 3;
print("Sum = ${a + b}");
```

Practical Thinking

1. Which string method would you use to validate email input?
2. Which number method is best for rounding product prices?
3. How do list methods help in building a to-do app?
4. Why are map methods useful for storing user profiles?
5. What problems could occur if data types had no built-in methods?

Key Takeaway

Methods make data types **powerful, reusable, and easy to work with**. Learning common methods will make your Dart programs **shorter, cleaner, and more efficient**.