Quiz 1: Dart Fundamentals

1. What are "Types"?
    a. **Data types of values you work with in your code**
    b. The name types of variables in your code.
    c. Function arguments used in your code

2. What's the job of "variables"?
    a. Variables allow you to execute code whenever you want.
    b. Variables describe which type of data you're working with.
    c. Variables hold data with which you can work in your code.

3. What are "functions"?
    a. Functions can be instantiated to create objects.
    b. Functions store data so that you can use it later.
    c. Functions hold code which you can call as often and whenever you want.

4. What's a "class"?
    a. A class is a data container for data with which you can work at a later point of time.
    b. A class is a blueprint for objects which you can create based on it.
    c. A class is basically the same as a Flutter widget.

5. What's a "property"?
    a. A variable inside a class.
    b. A function inside a class.
    c. A class inside a variable.

6. What's a "string"?
    a. A number value without decimal places.
    b. A number or text value
    c. A text value

7. What's NOT true about the below function

```
int calculateRoundedAge(int birthYear, int currentYear) {
  return currentYear - birthYear;
}

print(calculateRoundedAge(1989, 2020));
```

a. It takes two arguments which are both numbers.
b. It takes two arguments which match the return type of the function.
c. It returns a number with decimal places.
d. It's named correctly - describes what it does and uses camelCase.

Quiz 2: Flutter App Basics

1. What's the role of "runApp()?
    a. This function loads all required Flutter widgets.
    b. This function builds the main widget and calls build() on it.
    c. This function is the first function Dart executes when the file is loaded.

2. What's the job of the " build() " method?
    a. The build() method returns the widgets ("widget tree") which should be rendered onto the screen.
    b. The build() method creates a new variable.
    c. The build() method returns the widgets you may use to build your app.

3. What's a "Widget"?
    a. It's an alternative name for a Flutter app
    b. Widgets are the core building blocks of Flutter user interfaces
    c. Widgets are a core Dart feature

Quiz 3: Widget Basics

1. What's the role of a Widget?
    a. It draws something visible onto the screen.
    b. You build your app's user interface from Widgets.
    c. A Widget is the result of your compiled Flutter code.

2. What's true about a Flutter Widget?
    a. Every Widget is a Dart object.
    b. Every Widget is compiled to a platform-specific equivalent UI element.
    c. Every Widget is visible on the screen.

3. What does the term "Widget tree" describe?
    a. "Widget tree" means that Widgets are typically pretty complex objects.
    b. "Widget tree" means that you build your UI with a combination of (nested) Widgets.
    c. "Widget tree" means that you have to use a certain amount of Widgets to see something on the screen.

4. Why do we need Widgets?

a. Widgets are the only type of Dart objects you can use in Flutter apps.
b. Widgets are the only way of running your own code in a Flutter app.
c. Widgets are the core building block with which you compose rich user interfaces in Flutter.

5. How do you combine Widgets?
    a. You simply return multiple Widgets in your build() method - i.e. you have multiple return statements.
    b. You pass Widgets into special functions provided by Flutter.
    c. You pass Widgets into the constructors of other Widgets to fill certain roles.

6. What makes a Dart object a Widget?
    a. Extending StatelessWidget / StatefulWidget.
    b. Extending StatelessWidget / StatefulWidget and implementing a build() method.
    c. Extending StatelessWidget / StatefulWidget, implementing a build() method and having at least two constructors.

Quiz 4: Stateful & Stateless Widgets

1. What's the core difference between StatelessWidget and StatefulWidget?
    a. StatelessWidget can't contain and change properties.
    b. StatelessWidget can't re-run build() when its properties change.
    c. StatelessWidgets can never show updated content.

2. What's the difference between "Input Data" (also: "External Data) and "Internal Data" in a Widget?
    a. Input Data is the data received via the constructor of a Widget. If that changes, the Widget is rebuilt.
    b. Input Data is the data received from the user of your app.

3. Why do you need to call setState(() { ... }) in a StatefulWidget (when changing some internal data)?
    a. Without setState(), the Widget does not change its property value and hence it will not re-run the build() method. Hence updates aren't reflected on the screen.
    b. Without setState(), the Widget does change its property value but it will not re-run the build() method. Hence updates aren't reflected on the screen.
    c. setState() has no direct impact but is a good convention as it makes it really clear that you want to actively change some data.

Quiz 5: Advanced Flutter & Dart Basics

1. What's special about this class? class _Person { final _name = 'Max'; }

a. It's just a special way of naming the class and property that does not have any meaning.
b. It's a private class (with a private property) which can be only be used in your project.
c. It's a private class (with a private property) which can only be used in its own library / file.

2. What's the idea behind an "enum"?
   a. An enum is a value where you can choose from multiple possible choices. You choose by selecting a human-readable label, behind the scenes, the labels are mapped to integers (0, 1, ...).
   b. An enum is basically a number that has a human readable label.
   c. An enum is a value where you can choose from multiple possible choices. You choose by selecting a human-readable label, this label is then used as a string value in your code.

3. What can EdgeInsets.all(...) be used for?
   a. Create an object with a special configuration / special default property values.
   b. Create a function with a special configuration / special argument values.

4. What's the difference between a List ([]) and a Map ({}) in Dart / Flutter?
   a. There is no difference
   b. Lists give you an ordered list of single values, identified by an index. Maps use key-value pairs where each pair has its own index.
   c. Lists give you an ordered list of single values, identified by an index. Maps use key-value pairs where you identify values by their key.

5. What does the following code snippet produce? final names = ['Max', 'Manu', 'Julie']; final result = names.map((name) => Text(name)).toList();
   a. result is a list of Text() widgets where each widget holds a different name from names.
   b. names is a list of Text() widgets where each widget holds a different name from names.
   c. This will throw an error

6. What's true about final ?
   a. Properties or variables marked as final can't change at compile-time.
   b. Properties or variables marked as final can't change at runtime.
   c. Properties or variables marked as final can't change at all.