Chapter 6 – Advanced Object Types

1. Introduction
   1. One of the challenges of writing TypeScript is knowing how to apply types in every situation we’ll encounter within our code
   2. we’ll learn how to use types with object-oriented programming patterns, how to use types together to create combined types
2. Interfaces and Types
   1. there’s another way to define types with the interface keyword.
   2. The syntaxes for type and interface are slightly different, since interface does not require an equals sign (=) before the typed object

Example

|  |
| --- |
| type Mail = {  postagePrice: number;  address: string;  }    const catalog: Mail = ... |

Example 2

|  |
| --- |
| interface Mail {  postagePrice: number;  address: string;  }    const catalog: Mail = ... |

Exercise

|  |
| --- |
| // Write an interface here  interface Run {  miles: number  }  function updateRunGoal(run: Run) {  console.log(`  Miles left: ${50 - run.miles}  Percent of goal: ${(run.miles / 50) \* 100}% complete  `)  }  updateRunGoal({  miles: 5,  }) |

1. Interfaces and Classes
   1. The interface keyword in TypeScript is especially good for adding types to a class
      1. interface is constrained to typed objects
      2. class is a way to program with objects

Example

|  |
| --- |
| interface Robot {  identify: (id: number) => void;  }    class OneSeries implements Robot {  identify(id: number) {  console.log(`beep! I'm ${id.toFixed(2)}.`);  }    answerQuestion() {  console.log('42!');  }  } |