Quiz

This quiz will test what you have learned about creating types in TypeScript.

Creating types

1 What is the correct type annotation for creating an array of dates?

2 What will TypeScript infer the type of the `people` variable to be in the declaration below?

```
let people = ["Paula", "Bob", "Sam"];
```

3

We have a function below that takes in a varying number of message parameters and outputs them to the console. How can we strongly-type the `messages` rest parameter so that only `string` types can be passed?

```
function outputMessages(...messages) {
    messages.forEach(message => console.log(message));
}
```

4

We have declared a `point` variable as follows:

```
const point = { x: 32, y: 77 };
```

Will the following assignment generate any type errors?

```
point.x = 40;
point.y = 80;
point.z = 10;
```

5 We have declared a `person` variable as an object with `firstName` and `level` properties as follows:

```
let person: { firstName: string, level: "high" | "Medium" |
"low" };
```

We want to use this type in several places in our codebase. What’s the best way to do this?

6

Are the `Dog` and `Apple` types structurally equivalent?

```
type Animal = {
    name: string;
}
type Dog = Animal & { size: "small" | "large" }

interface Fruit {
    name: string;
}
interface Apple extends Fruit {
    size: "small" | "large"
}
```

7

A type error is generated on the last line of the code below:

```
type Action = {
    readonly type: string;
}

const action: Action = { type: "loading" };
action.type = "loaded";
```

Why is this?

8

Look at the code below:

```
type Loaded = {
  readonly type: "loaded",
  readonly data: string[]
}
const loadedAction: Loaded = {
  type: "loaded",
  data: ["Fred", "Bob"]
};
loadedAction.data.push("Jane");
```

Will the line `loadedAction.data.push("Jane");` generate a type error?

9

Are the following two types both valid and structurally equivalent?

```
type Person = {
    name: string;

    dob?: Date;
}
```

```
interface Person {
    name: string;
    dob?: Date;
}
```

10

We need to create a tuple type to hold x and y coordinates. What is the correct type definition for this?

Check Answers

Well done, we are really getting to grips with TypeScript now!

In the next chapter, we will learn how to make our TypeScript types more reusable.