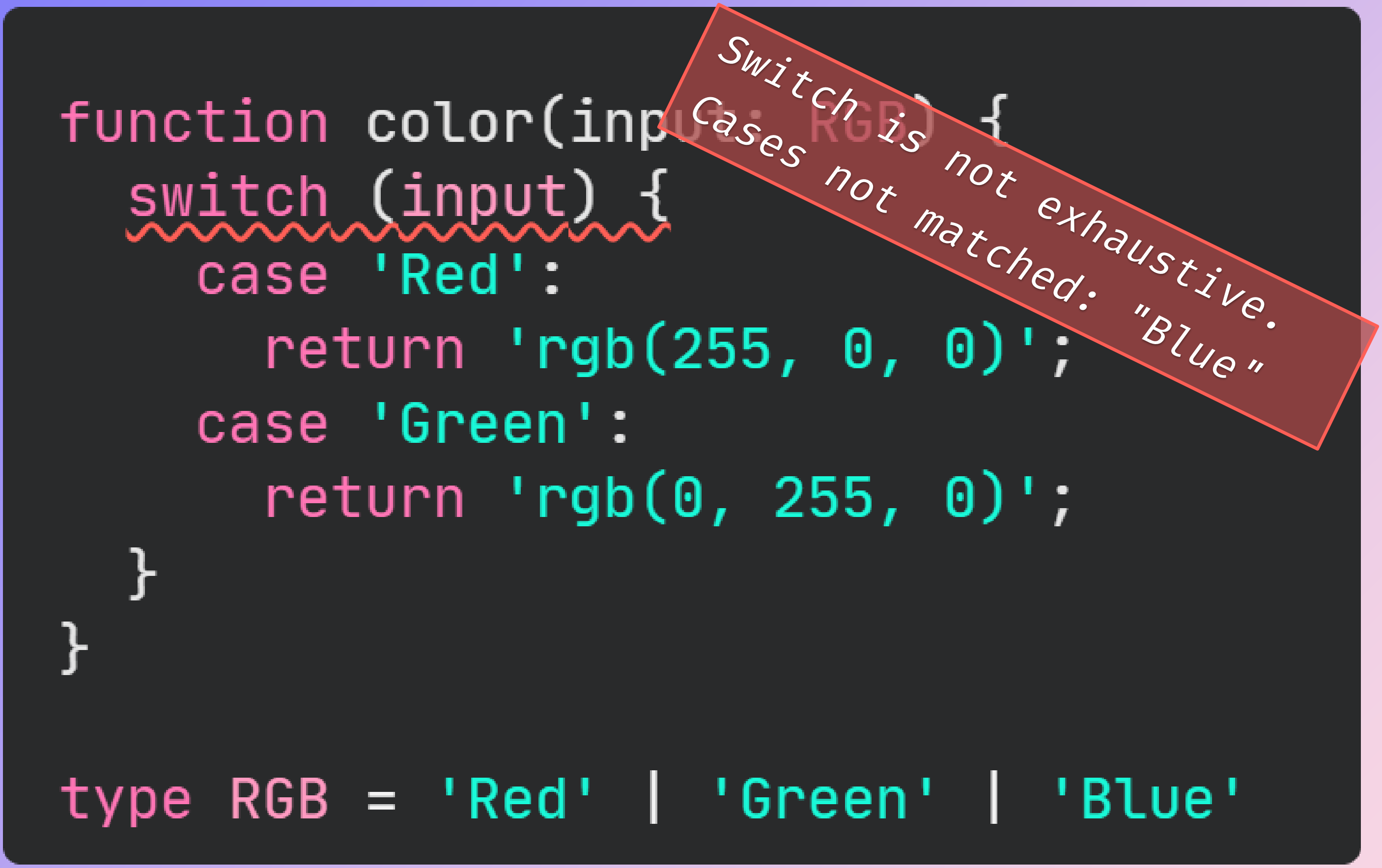
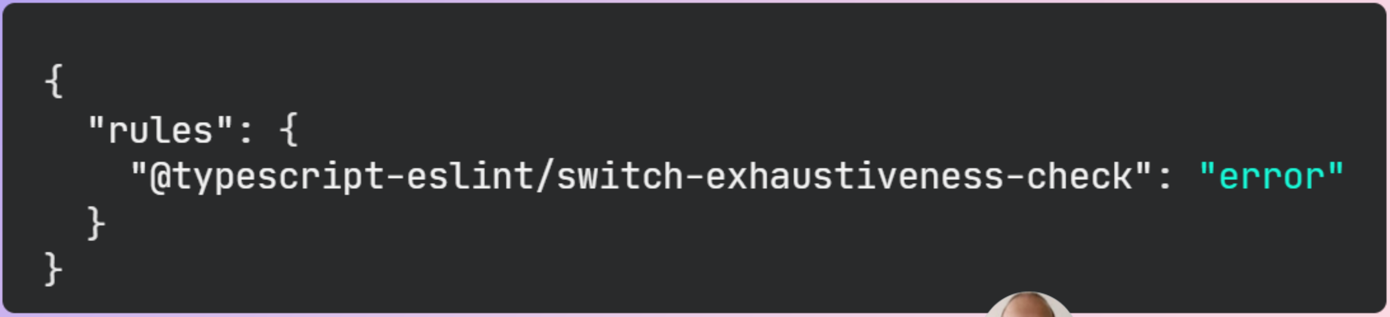
Switch Exhaustiveness in TypeScript



Switch is not exhaustive.
Cases not matched: "Blue"

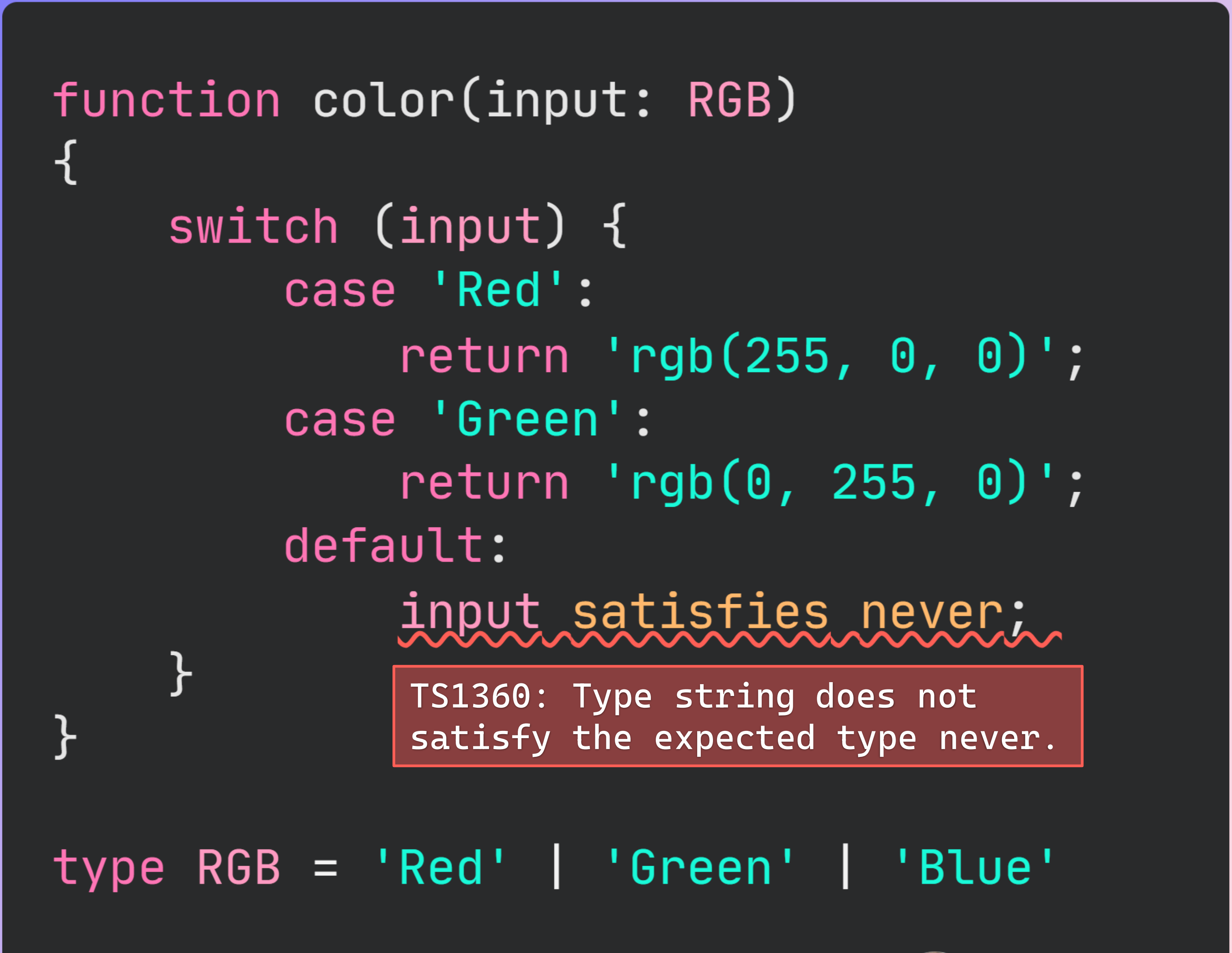
```
function color(input: RGB) {
  switch (input) {
    case 'Red':
      return 'rgb(255, 0, 0)';
    case 'Green':
      return 'rgb(0, 255, 0)';
  }
}

type RGB = 'Red' | 'Green' | 'Blue'
```



```
{
  "rules": {
    "@typescript-eslint/switch-exhaustiveness-check": "error"
  }
}
```

timdeschryver.dev/bits

Switch Exhaustiveness
in TypeScript



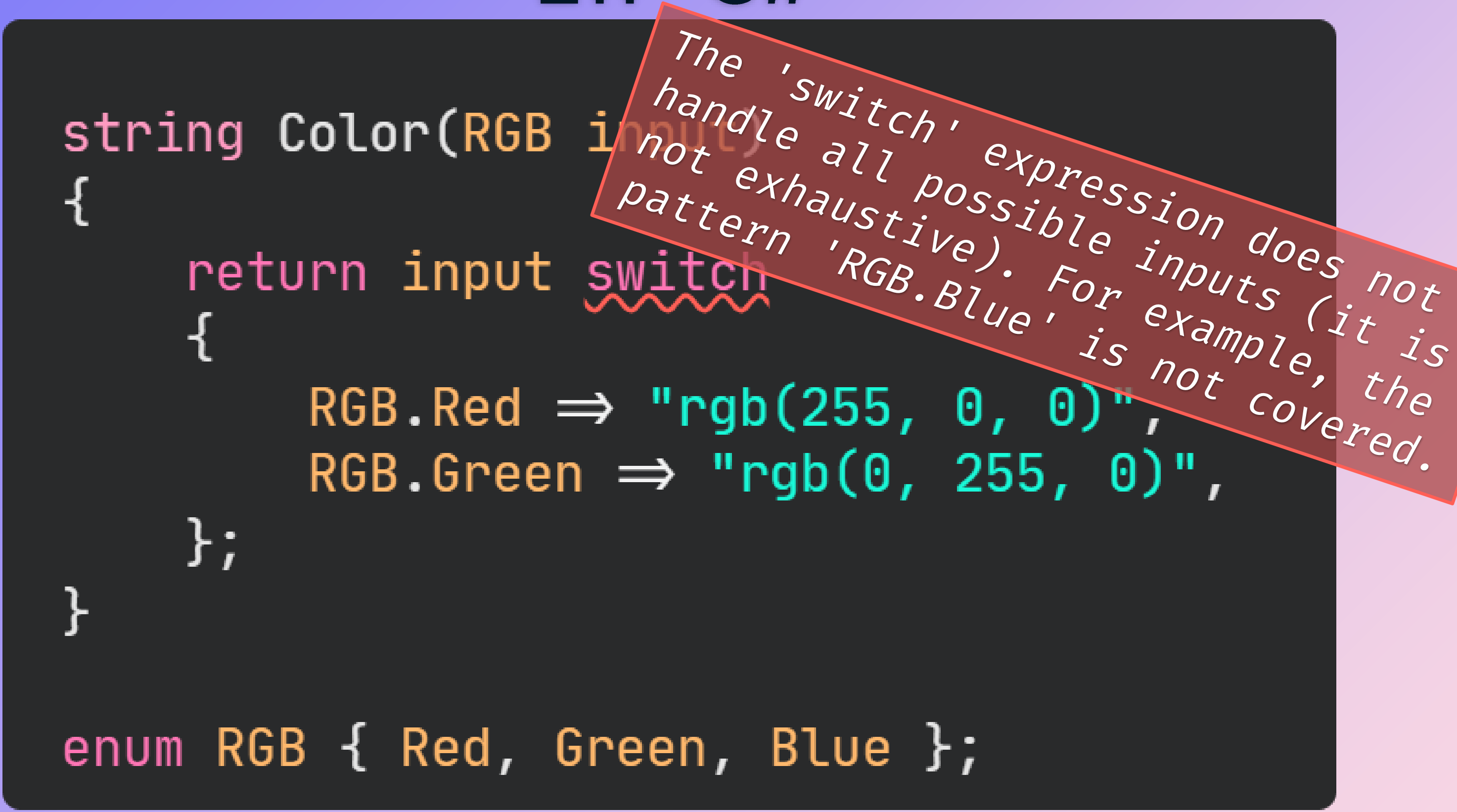
```typescript
function color(input: RGB)
{
    switch (input) {
        case 'Red':
            return 'rgb(255, 0, 0)';
        case 'Green':
            return 'rgb(0, 255, 0)';
        default:
            input satisfies never;
    }
}

type RGB = 'Red' | 'Green' | 'Blue'
```

TS1360: Type string does not satisfy the expected type never.

timdeschryver.dev/bits

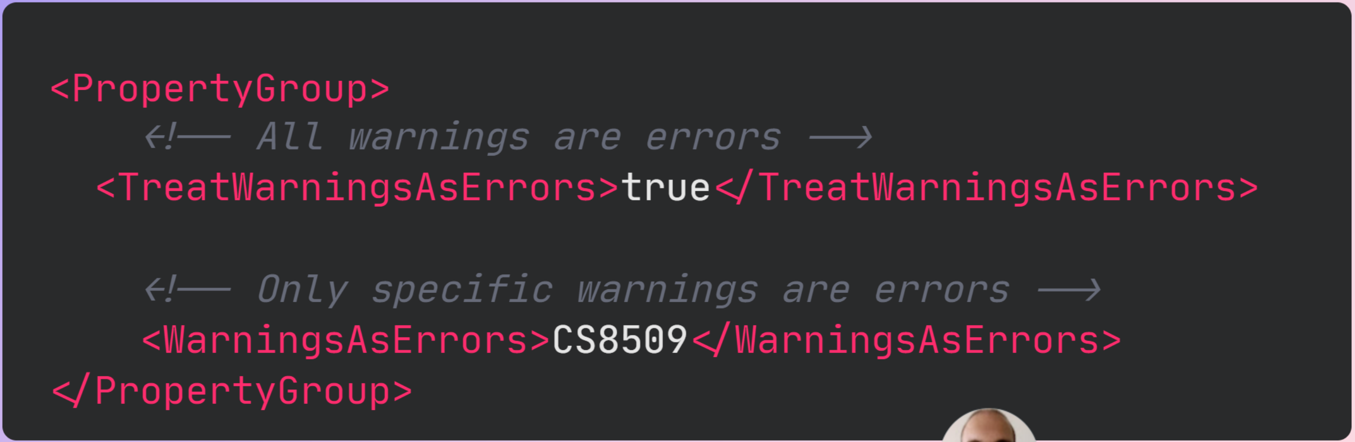
Switch Exhaustiveness

in C#



```csharp
string Color(RGB input)
{
    return input switch
    {
        RGB.Red => "rgb(255, 0, 0)",
        RGB.Green => "rgb(0, 255, 0)",
    };
}

enum RGB { Red, Green, Blue };
```

The 'switch' expression does not handle all possible inputs (it is not exhaustive). For example, the pattern 'RGB.Blue' is not covered.



```xml
<PropertyGroup>
    <!-- All warnings are errors -->
  <TreatWarningsAsErrors>true</TreatWarningsAsErrors>

    <!-- Only specific warnings are errors -->
    <WarningsAsErrors>CS8509</WarningsAsErrors>
</PropertyGroup>
```

timdeschryver.dev/bits