Sub Procedures, Part I (Continue)

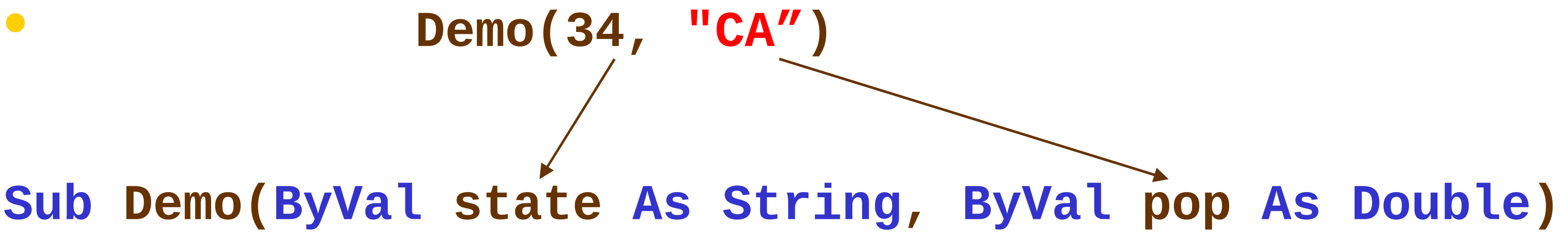
Variables and Expressions as Arguments

```
Dim s As String = "CA"
Dim p As Double = 17
Demo(s, 2 * p)


Sub Demo(ByVal state As String, ByVal pop As Double)
  txtBox.Text = state & " has population " & pop & _
                                         " million."
End Sub
```

- **Note**: The variable names in the arguments need not match the parameter names. For instance, *s* versus *state*..

Passing Strings and Numbers



- Demo(34, "CA”)

```
Sub Demo(ByVal state As String, ByVal pop As Double)
  txtBox.Text = state & " has population " & pop & _
                                            " million."
End Sub
```

- **Note**: The statement `Demo(34, "CA")` would not be valid. The types of the arguments must be in the same order as the types of the parameters.

Calling

A Sub procedure can call another Sub procedure.

```
Private Sub btnAdd_Click(...) Handles btnAdd.Click
  Sum(2, 3)
End Sub

Sub Sum(ByVal num1 As Double, ByVal num2 As Double)
  DisplayPurpose()
  lstBox.Items.Add("The sum of " & num1 & " and " _
           & num2 & " is " & (num1 + num2) & ".")
End Sub
```