```
def foo():
    return 0
```

```
string title = "This is a Unicode pi in the sky"
/*
Defined as $\pi = \lim_{n\to\infty} \frac{P_n}{d}$ where $P$ is the perimeter
of an $n$-sided regular polygon circumscribing a
circle of diameter $d$.
*/
const double pi = 3.1415926535
```

```
# $\sum_{i=1}^{\infty} i + 1$
def foo():
    return 0
```

```
def f(x):
    return 'Some text ' + str(x) +
    ' some more text ' + str(x) +
        ' even more text that goes on for a while and a while longer, and so on
forever and ever.'
```

```
def f(x):
    y = x **2
return y
```

```
    some_string = 'SomeTextThatGoesOnAndOnForSoLongThatItCouldNeverFitOnOneLine'
```

```
␣␣␣␣def␣boring(args␣=␣None):
␣␣␣␣␣␣␣␣pass
```

```
def␣boring(args␣=␣None):
␣␣␣␣pass
```

```
def␣boring(args␣=␣None):
␣␣␣␣pass
```

```
x=~/foo/
```

```
    def f(x):
        return x**2
```

```
    def func
        puts "message"
    end
    sike = "sike"

    def g(x):
        return 2*x
```

```
def  (x):
    return x**2
def  (x):
    return 2*x
```

```
def hello():
    print("Hello, world!")
```

```
def foo():
# This is a comment that contains math: $\sum_{i=1}^{n} i$.
   return 0
```

```
def all(iterable):
    for i in iterable:
        if not i:
            return False
    return True
```

```
public boolean isRowValid(TextField textField, int numRows) {
    try {
        int row = Integer.parseInt(textField.getText()) - 1;
        if (row >= 0 && row < numRows) {
            return false;
        } else {
            System.out.println("Invalid row selection. Please choose a valid
            ↪  row.");
            Platform.runLater(textField::clear);
            return true;
        }
    } catch (NumberFormatException e) {
        System.out.println("Invalid input. Please enter a valid integer.");
        Platform.runLater(textField::clear);
        return true;
    }
}
```

```
public static Matrix convertBackToOriginalForm(String[][] matrix) {
    String[][] originalFormMatrix = new String[matrix.length][];
```

```
        System.out.println("Matrix length: " + matrix.length);
//          System.out.println("OG Matrix: \n");
        printStringMatrix(matrix);
        for (int i = 0; i < matrix.length; i++) {
            originalFormMatrix[i] = new String[matrix[i].length];
            for (int j = 0; j < matrix[i].length; j++) {
                originalFormMatrix[i][j] = MatrixApp.isFractionMode() ?
                ↪  convertDecimalToFraction(matrix[i][j]) :
                ↪  convertFractionToDecimalString(matrix[i][j]);
            }
        }
        System.out.println("Original Form Matrix: \n");
        return new Matrix(originalFormMatrix);
    }
```

```
    public static void printStringMatrix(String[][] matrix) {
        for (String[] row : matrix) {
            System.out.println(Arrays.toString(row));
        }
    }
```

```
def helloworld():
    print("Hello, Dracula!")
```