

main.py archive

Importing necessary Libraries

```
from unicodedata import name  
from functionAnalyzer import main
```

Calling the main method so the program starts

```
if __name__ == "__main__":  
    main()
```

functionAnalyzer.py archive

```
# Importing necessary Libraries
from sympy import limit, Symbol
import matplotlib.pyplot as plt
import numpy as np

# Method wich graphs and displays
def drawer(function, xPoint=0):
    # Finding the function type with a custom method
    function_type = functionType(function)

    # The graphic X range *(From, To, Each)*
    x = np.arange(-2501, 2501, 1)

    # Getting X and Y values
    y = eval(function)
    yPoint = evaluatePoint(str(xPoint), function)

    # Setting the title of the drawer with the function type
    plt.title(f"{function_type}: {function}")

    # The graphic Y range *(From, To)*
    plt.ylim([-500, 500])

    # Display the axes themeselves
    plt.axhline(0, color="black")
    plt.axvline(0, color="black")

    # Drawing the function itself
    plt.grid()
    plt.plot(
        np.where(x == xPoint - 2501),
        [yPoint],
        marker=".",
        markersize=10,
        markeredgecolor="green",
        markerfacecolor="green",
    )
    plt.plot(x, y)
    plt.show()

# Method wich evaluates the function on a certain point
def evaluatePoint(xPoint, function):
    f_type = functionType(function)
    if f_type == "Irrational" and int(xPoint) < 0:
        yPoint = "Imaginary number"
    else:
        yPoint = eval(function.replace("x", xPoint))
    print(xPoint, yPoint)
    return yPoint
```

Method wich analyzes and returns the function type

```
def functionType(function):  
    if ")" / "(" or "/" in function:  
        function_type = "Rational"  
        return function_type  
    elif "**(0" in function or "**0" in function:  
        function_type = "Irrational"  
        return function_type  
    else:  
        function_type = "Polinomial"  
        return function_type
```

Method wich analyze and returns the function limits based on the point

```
def getFunctionLimits(function, point):  
    x = Symbol("x")  
    limits = limit(function, x, point), limit(function, x, point, "-")  
    return f"limit of {function} when x = {point} \n {limits}"
```

Method wich ejecutes the program

```
def main():  
    function = str(input("Enter the function: "))  
    point = int(input("Enter the point: "))  
    func_type = functionType(function)  
    fdex = evaluatePoint(str(point), function)  
    limits = getFunctionLimits(function, point)  
    drawer(function, point)  
  
    print(f"The function type is: {func_type}")  
    print(f"The function in that point is equal to: {fdex}")  
    print(f"Limits of the funciton: {limits}")
```