ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Vol 10, Issue 11, November 2023

Research on the Use of Python in the Development of an Innovative Scientific Web Application with Developing Scientific Functionalities on the Adapted Scientific Calculator

[1] Vladimir Simovic, [2] Matija Varga, [3] Zdravko Kunic

[1] Algebra University College Zagreb & University of Applied Sciences Baltazar Zapresic (Croatia-EU)
[2] University North Koprivnica & University of Applied Sciences Baltazar Zapresic (Croatia-EU)
[3] Algebra University College Zagreb
Corresponding Author Email: [1] Vladimir.Simovic@algebra.hr, [2] mvarga@unin.hr, [3] Zdravko.Kunic@algebra.hr

Abstract*— This pilot research will present the survey results on the specific topic of "Research on the use of Python in the development of an innovative scientific web application with developing scientific functionalities on the adapted scientific calculator". Python is an object-oriented programming language that, in addition to being object-oriented, also allows for procedural programming methodology and an innovative data approach. Python applications operate in world-famous corporations like Google, NASA, IBM, and Autodesk. As a novelty of this research, we should investigate which innovative and scientific functionalities are used in similar application software solutions of the world, European, and nationally known corporations, and which is the subject of new scientific research in future. This research relates to a similar research named "Pilot research on the minimum required knowledge of web programming, development tools and programming languages". The purpose of this paper is to present first information about the research background of the usage of the Python programming language in the development of a particular part of innovative scientific web applications with the case of developing scientific functions on a tuned scientific calculator. The primary research methodologies used here are the method of scientific and professional text analysis, the survey method, the chi-square test, and the comparative method. This research will be used for the next, more significant, and similar research.

***Index Terms*—Python, web, innovative, scientific functionalities, calculator.**

I. INTRODUCTION

This research paper presents the results of a survey A lot of innovative authors deal with the functionalities of different scientific calculators that use the best performance of portable kernels in Python as noted in [1], [2] and [7] also enabling type checking on columns in different data frame libraries as it is noted in [11] and with various modern technology products like Blockchain-based learning solutions as it is noted in [3] and [6], etc. The article entitled "Research on the use of Python in the development of an innovative scientific web application with developing scientific functionalities on the adapted Scientific Calculator" presented in this paper includes a program created in Python that allows users to

(1) Apply basic arithmetic (calculation) operators for addition, subtraction, multiplication, division, exponentiation, etc.),
(2) Rooting a given number,
(3) Displaying the value of the variable PI and applying the value of PI,
(4) Applying trigonometric functions in the calculation,
(5) Applying logarithmic functions, etc.

The objectives of the work are to find answers to problems, such like:

(1) Will be all this functional and applicable in the so-called real (private or public) sector,
(2) Show how to apply all this for the program, program options, and select options,
(3) Show the logic of program development only in terms of program code and the use of the Python programming language in which the application is developing.

The Scientific Calculator app allows you to select two basic views of the calculator using the File drop-down menu. There are different views: (1) 'Standard' and (2) 'Scientific' while the third option (3) is the so-called 'Exit' as the exit from the "Scientific calculator" application. Also, in the application navigation bar, there is another drop-down 'menu' so-called 'Edit' that allows (buttons) to select options: (1) cutting, (2) copying, and (3) pasting. The program is in Python (programming language). We will show how Python is an object-oriented programming language and, in addition to being object-oriented, also allows procedural programming methodology as is noted [8]. Also, well-known corporations such as Google, NASE, IBM, Autodesk (etc.) are using Python as it is noted in [9].

The program has innovative solutions (in this paper) as program codes and signs. Each command used in the program is shown in detail, as loops and branches, to the level of programming code. The paper used the scientific method of analysis of scientific content and the analysis of an

ISSN (Online) 2456 -1304
International Journal of Science, Engineering and Management (IJSEM)
Vol 10, Issue 11, November 2023

innovative scientific calculator application made in Python.

In addition to the presented applications, the research will present the results of the survey and answers to the following questions: (1) did the respondents know that applications created with the Python programming language can also be used as web applications if they are placed on a web server, (2) do the respondents use Python programming language when creating web applications, (3) whether the respondents knew how the Python programming language is used to manipulate data in existing relationships within the database, (4) whether the respondents use the Python programming language to manipulate data in existing relationships within the database and (5) is there a better (and more necessary) scientific web calculator for the respondents and their needs, i.e. everyday use (in the scientific sense and with scientific functions (sin, cos, tan, pi, e, etc.) made in the Python programming language) or a standard calculator (in a standard view with standard functions (plus, minus, times, division, exponentiation, root, etc.) made in the Python programming language).

II. A SCIENTIFIC CALCULATOR APPLICATION CREATED WITH PYTHON - IN BRIEF

A. The program Scientific Calculator in Python – standard view

The scientific calculator application was created in Python (programming language) and is presented in this chapter using the final version of the program (i.e., the application) launched by the Run Module option in IDLE (Python's IDLE means abbreviation from Integrated Development and Learning Environment). This paper presents the Python programming language used during the development of specific scientific applications in which functions: branches & loops were applied.

Figure 1. The scientific calculator application in the standard view
Source: Authors.

Figure 1 as noted in [10] and [4] shows a scientific calculator application in standard view with arithmetic (calculation) operators (for addition, subtraction, multiplication, division, exponentiation, etc.). It is also possible to root the default number using some basic options. Figure 1 also shows a graphical interface created in the Python programming language. I.e., there are buttons for calculation operations and numerical parts (numbers) of the scientific calculator developed. In addition, the application has a "screen" to display the results of calculations from which it is possible to copy data, paste and cut. The app has buttons to minimize, maximize form and exit the app. The second part of the application created in Python shows the scientific calculator (Figure 2).

B. The program Scientific Calculator in Python – design view

This part presents the Scientific Calculator program through the Scientific option created in Python, while the module represents the so-called Scientific design view as it is noted in [10] and [4].

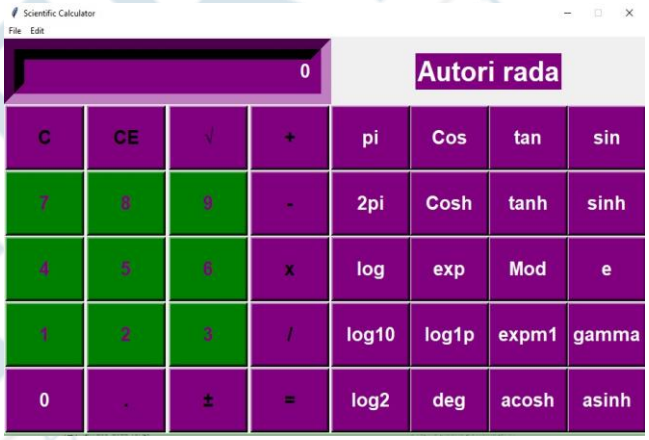

Figure 2. View of the scientific calculator application in scientific terms
Source: Authors (in Croatian: Autori rada).

Figure 2 as it is noted in [10] and [4] shows the Scientific calculator application in scientific terms. In addition to (1) the basic arithmetic (calculation) operators (for addition, subtraction, multiplication, division, exponentiation, etc.), the form shows the possibility of (2) rooting a given number, (3) displaying the value of the PI constant and applying the value of the PI constant… (Napier constants) Euler's number (4) application of trigonometric functions (sine, cosine, tangent, cotangent) in the calculation, (5) application of logarithmic functions, etc.

C. The program Scientific Calculator in Python – code view

This part shows the programming code in the Python programming language. The intention is to create a scientific calculator application. The code view is explained below (below) as it is noted in [10] and [4]:

```
from tkinter import *
import math
import tkinter.messagebox
```

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Vol 10, Issue 11, November 2023

The Python programming language allows you to import modules required in a program with the import command as it is noted in [10] and [4].

```
root = Tk()
root.title("Scientific Calculator")
root.configure(background = 'lightgreen')
root.resizable(width=False, height=False)
root.geometry("480x568+450+90")
calc = Frame(root)
calc.grid()

class Calc():
    def __init__(self):
        self.total=0
        self.current=''
        self.input_value=True
        self.check_sum=False
        self.op=''
        self.result=False
```

Branches are explored within the Calc () class. It is very often in life and computer programs. The action next in the text depends on the outcome of the question that starts with "if". That gives the possibility of branching into different types of action depending on some criteria as it is noted [8].

```
    def numberEnter(self, num):
        self.result=False
        firstnum=txtDisplay.get()
        secondnum=str(num)
        if self.input_value:
            self.current = secondnum
            self.input_value=False
        else:
            if secondnum == '.':
                if secondnum in firstnum:
                    return
            self.current = firstnum+secondnum
        self.display(self.current)

    def sum_of_total(self):
        self.result=True
        self.current=float(self.current)
        if self.check_sum==True:
            self.valid_function()
        else:
            self.total=float(txtDisplay.get())

    def display(self, value):
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, value)
```

In this part of the code (below) for the scientific calculator application are defined functions as it is noted in [10] and [4]:

```
    def valid_function(self):
        if self.op == "add":
            self.total += self.current
        if self.op == "sub":
            self.total -= self.current
        if self.op == "multi":
            self.total *= self.current
        if self.op == "divide":
            self.total /= self.current
        if self.op == "mod":
            self.total %= self.current
        self.input_value=True
        self.check_sum=False
        self.display(self.total)

    def operation(self, op):
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        elif not self.result:
            self.total=self.current
            self.input_value=True
        self.check_sum=True
        self.op=op
        self.result=False

    def Clear_Entry(self):
        self.result = False
        self.current = "0"
        self.display(0)
        self.input_value=True

    def All_Clear_Entry(self):
        self.Clear_Entry()
        self.total=0

    def pi(self):
        self.result =  False
        self.current = math.pi
        self.display(self.current)

    def tau(self):
        self.result =  False
        self.current = math.tau
        self.display(self.current)

    def e(self):
        self.result =  False
        self.current = math.e
        self.display(self.current)

    def mathPM(self):
        self.result = False
        self.current = -(float(txtDisplay.get()))
        self.display(self.current)

    def squared(self):
        self.result = False
        self.current = math.sqrt(float(txtDisplay.get()))
        self.display(self.current)

    def cos(self):
        self.result = False
```

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Vol 10, Issue 11, November 2023

```
        self.current = math.cos(math.radians(float(txtDisplay.get())))
        self.display(self.current)

    def cosh(self):
        self.result = False
        self.current = math.cosh(math.radians(float(txtDisplay.get())))
        self.display(self.current)

    def tan(self):
        self.result = False
        self.current = math.tan(math.radians(float(txtDisplay.get())))
        self.display(self.current)

    def tanh(self):
        self.result = False
        self.current = math.tanh(math.radians(float(txtDisplay.get())))
        self.display(self.current)

    def sin(self):
        self.result = False
        self.current = math.sin(math.radians(float(txtDisplay.get())))
        self.display(self.current)

    def sinh(self):
        self.result = False
        self.current = math.sinh(math.radians(float(txtDisplay.get())))
        self.display(self.current)

    def log(self):
        self.result = False
        self.current = math.log(float(txtDisplay.get()))
        self.display(self.current)

    def exp(self):
        self.result = False
        self.current = math.exp(float(txtDisplay.get()))
        self.display(self.current)

    def acosh(self):
        self.result = False
        self.current = math.acosh(float(txtDisplay.get()))
        self.display(self.current)

    def asinh(self):
        self.result = False
        self.current = math.asinh(float(txtDisplay.get()))
        self.display(self.current)

    def expm1(self):
        self.result = False
        self.current = math.expm1(float(txtDisplay.get()))
        self.display(self.current)

    def lgamma(self):
        self.result = False
        self.current = math.lgamma(float(txtDisplay.get()))
        self.display(self.current)

    def degrees(self):
        self.result = False
        self.current = math.degrees(float(txtDisplay.get()))
        self.display(self.current)

    def log2(self):
        self.result = False
        self.current = math.log2(float(txtDisplay.get()))
        self.display(self.current)

    def log10(self):
        self.result = False
        self.current = math.log10(float(txtDisplay.get()))
        self.display(self.current)

    def log1p(self):
        self.result = False
        self.current = math.log1p(float(txtDisplay.get()))
        self.display(self.current)

added_value = Calc()

txtDisplay = Entry(calc, font=('Helvetica',20,'bold'),
                   bg='purple',fg='white',
                   bd=30,width=28,justify=RIGHT)
txtDisplay.grid(row=0,column=0, columnspan=4, pady=1)
txtDisplay.insert(0,"0")

numberpad = "789456123"
i=0
btn = []
for j in range(2,5):
    for k in range(3):
        btn.append(Button(calc, width=6, height=2,
                          bg='green',fg='purple',
                          font=('Helvetica',20,'bold'),
                          bd=4,text=numberpad[i]))
        btn[i].grid(row=j, column= k, pady = 1)
        btn[i]["command"]=lambda x=numberpad[i]:added_value.numberEnter(x)
        i+=1
```

In the continuation of the program (below), the calculator buttons (btn) are defined and each button's height, width, background colour, font, font size, font style, etc. as noted in [10] and [4]:

```
btnClear = Button(calc, text=chr(67),width=6,
                  height=2,bg='purple',
                  font=('Helvetica',20,'bold')
                  ,bd=4, command=added_value.Clear_Entry
                 ).grid(row=1, column= 0, pady = 1)

btnAllClear = Button(calc, text=chr(67)+chr(69),
                     width=6, height=2,
                     bg='purple',
```

```
                font=('Helvetica',20,'bold'),
                bd=4,
                command=added_value.All_Clear_Entry
               ).grid(row=1, column= 1, pady = 1)

    btnsq = Button(calc, text="\u221A",width=6, height=2,
                bg='purple', font=('Helvetica',
                                   20,'bold'),
               bd=4,command=added_value.squared
              ).grid(row=1, column= 2, pady = 1)

    btnAdd = Button(calc, text="+",width=6, height=2,
                bg='purple',
                font=('Helvetica',20,'bold'),

bd=4,command=lambda:added_value.operation("add")
                ).grid(row=1, column= 3, pady = 1)

    btnSub = Button(calc, text="-",width=6,
                height=2,bg='purple',
                font=('Helvetica',20,'bold'),

bd=4,command=lambda:added_value.operation("sub")
                ).grid(row=2, column= 3, pady = 1)

    btnMul = Button(calc, text="x",width=6,
                height=2,bg='purple',
                font=('Helvetica',20,'bold'),

bd=4,command=lambda:added_value.operation("multi")
                ).grid(row=3, column= 3, pady = 1)

    btnDiv = Button(calc, text="/",width=6,
                height=2,bg='purple',
                font=('Helvetica',20,'bold'),

bd=4,command=lambda:added_value.operation("divide")
                ).grid(row=4, column= 3, pady = 1)

    btnZero = Button(calc, text="0",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),

bd=4,command=lambda:added_value.numberEnter(0)
                 ).grid(row=5, column= 0, pady = 1)

    btnDot = Button(calc, text=".",width=6,
                height=2,bg='purple',
                font=('Helvetica',20,'bold'),

bd=4,command=lambda:added_value.numberEnter(".")
                ).grid(row=5, column= 1, pady = 1)
    btnPM = Button(calc, text=chr(177),width=6,
               height=2,bg='purple',
font=('Helvetica',20,'bold'),
               bd=4,command=added_value.mathPM
              ).grid(row=5, column= 2, pady = 1)

    btnEquals = Button(calc, text="=",width=6,
                height=2,bg='purple',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.sum_of_total
               ).grid(row=5, column= 3, pady = 1)
# ROW 1 :
btnPi = Button(calc, text="pi",width=6,
               height=2,bg='purple',fg='white',
               font=('Helvetica',20,'bold'),
               bd=4,command=added_value.pi
              ).grid(row=1, column= 4, pady = 1)

btnCos = Button(calc, text="Cos",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.cos
               ).grid(row=1, column= 5, pady = 1)

btntan = Button(calc, text="tan",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.tan
               ).grid(row=1, column= 6, pady = 1)

btnsin = Button(calc, text="sin",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.sin
               ).grid(row=1, column= 7, pady = 1)

# ROW 2 :
btn2Pi = Button(calc, text="2pi",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.tau
               ).grid(row=2, column= 4, pady = 1)

btnCosh = Button(calc, text="Cosh",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.cosh
                ).grid(row=2, column= 5, pady = 1)

btntanh = Button(calc, text="tanh",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.tanh
                ).grid(row=2, column= 6, pady = 1)

btnsinh = Button(calc, text="sinh",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.sinh
                ).grid(row=2, column= 7, pady = 1)

# ROW 3 :
btnlog = Button(calc, text="log",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.log
```

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Vol 10, Issue 11, November 2023

```
        ).grid(row=3, column= 4, pady = 1)

    btnExp = Button(calc, text="exp",width=6, height=2,
                bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.exp
               ).grid(row=3, column= 5, pady = 1)

    btnMod = Button(calc, text="Mod",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),


bd=4,command=lambda:added_value.operation("mod")
               ).grid(row=3, column= 6, pady = 1)

    btnE   = Button(calc, text="e",width=6,
                height=2,bg='purple',fg='white',
                font=('Helvetica',20,'bold'),
                bd=4,command=added_value.e
               ).grid(row=3, column= 7, pady = 1)

    # ROW 4 :
    btnlog10 = Button(calc, text="log10",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.log10
                ).grid(row=4, column= 4, pady = 1)

    btncos   = Button(calc, text="log1p",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.log1p
                ).grid(row=4, column= 5, pady = 1)

    btnexpm1 = Button(calc, text="expm1",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd = 4,command=added_value.expm1
                ).grid(row=4, column= 6, pady = 1)

    btngamma = Button(calc, text="gamma",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.lgamma
                ).grid(row=4, column= 7, pady = 1)
    # ROW 5 :
    btnlog2 = Button(calc, text="log2",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.log2
                ).grid(row=5, column= 4, pady = 1)

    btndeg = Button(calc, text="deg",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.degrees
                ).grid(row=5, column= 5, pady = 1)

    btnacosh = Button(calc, text="acosh",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.acosh
                ).grid(row=5, column= 6, pady = 1)

    btnasinh = Button(calc, text="asinh",width=6,
                 height=2,bg='purple',fg='white',
                 font=('Helvetica',20,'bold'),
                 bd=4,command=added_value.asinh
                ).grid(row=5, column= 7, pady = 1)

    lblDisplay = Label(calc, text = "Autori rada",
                  font=('Helvetica',30,'bold'),
                  bg='purple',fg='white',justify=CENTER)

    lblDisplay.grid(row=0, column= 4,columnspan=4)

    def iExit():
        iExit = tkinter.messagebox.askyesno("Scientific
Calculator",
                                    "Do you want to exit ?")
        if iExit>0:
            root.destroy()
            return

    def Scientific():
        root.resizable(width=False, height=False)
        root.geometry("944x568+0+0")

    def Standard():
        root.resizable(width=False, height=False)
        root.geometry("480x568+0+0")
```

In the next part of the code (below) a menu (menubar) is defined which consists of two parts: (1) File and (2) Edit as it is noted in [4].

```
    menubar = Menu(calc)

    # ManuBar 1 :
    filemenu = Menu(menubar, tearoff = 0)
    menubar.add_cascade(label = 'File', menu = filemenu)
    filemenu.add_command(label = "Standard", command =
Standard)
    filemenu.add_command(label = "Scientific", command =
Scientific)
    filemenu.add_separator()
    filemenu.add_command(label = "Exit", command = iExit)

    # ManuBar 2 :
    editmenu = Menu(menubar, tearoff = 0)
    menubar.add_cascade(label = 'Edit', menu = editmenu)
    editmenu.add_command(label = "Cut")
    editmenu.add_command(label = "Copy")
    editmenu.add_separator()
    editmenu.add_command(label = "Paste")

    root.config(menu=menubar)
    root.mainloop()
```

Python is used in this manner, but its unique features offer an environment that makes it a better choice for scientists and

ISSN (Online) 2456 -1304
International Journal of Science, Engineering and Management (IJSEM)
Vol 10, Issue 11, November 2023

engineers looking for a high-level language for writing scientific applications. Python stands out as a platform for scientific computing as it is noted [9]. For this reason, we also decided to apply a scientific calculator as it is noted in [10] and [4].

III. RESULTS OF THE SURVEY ON THE TOPIC OF THE SURVEY: "PYTHON PROGRAMMING LANGUAGE WHEN CREATING WEB APPLICATIONS"

In this chapter, the research results obtained based on the scientific method of surveying are presented. The research sample is N=43. The selected sample is intentional because they wanted to collect relevant data based on the research plan, i.e., data from experts and scientists who deal with programming and web programming and the creation of web applications [5].

Figure 3 shows the result on whether respondents understand how applications created in the Python programming language can also be used as web applications (if they are placed on a web server). Most respondents, 83.7%, knew that applications created with the Python programming language can also be used as web applications if they are placed on a web server, while 14% of respondents did not know about it. One respondent declared that she did not think about it.

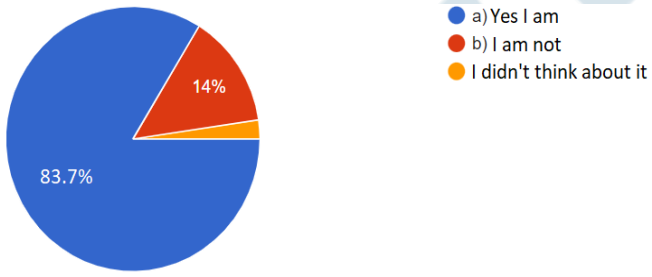

Figure 3. Results on whether respondents understand how applications created in the Python programming language can also be used as web applications (if they are placed on a web server)
Source: Authors.

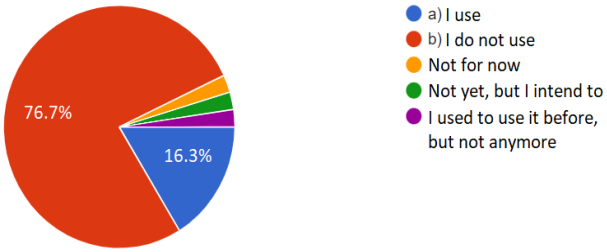

Figure 4. Results on the respondents' use of the Python programming language when creating web applications
Source: Authors.

Figure 4 shows the result on whether respondents use the Python programming language when creating web applications. The largest number of respondents, 76.7%, chose not to use the Python programming language when creating web applications, while 16.3% of respondents chose the option to use the Python programming language when creating web applications. Also, the other answers were like: "Not for now", "Not yet, but I intend to" and "I used to use it, now I don't anymore".

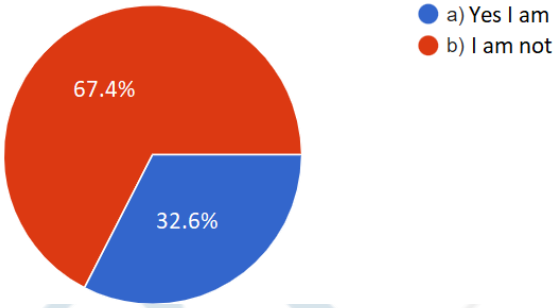

Figure 5. Results on whether the respondents knew how the Python programming language is used to manipulate data in existing relations within the database
Source: Authors.

Figure 5 shows the result on whether the respondents knew how the Python programming language is used to manipulate data in existing relations within the database. The largest number of respondents, 67.4%, knew that the Python programming language is used to manipulate data in existing relationships within the database, while 32.6% of respondents did not know how the Python programming language is used to manipulate data in existing relationships within the database.

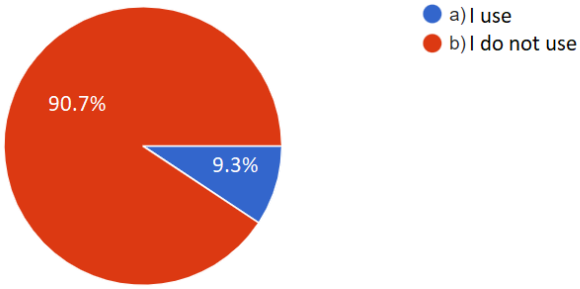

Figure 6. Results on the use of the Python programming language for data manipulation in existing relations within the database
Source: Authors.

Figure 6 shows the result on whether respondents use the Python programming language to manipulate data in existing relations within the database. Most respondents 90.7% do not use the Python programming language to manipulate data in existing relations within the database, while 9.3% of respondents use the Python programming language to manipulate data in existing relations within the database.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Vol 10, Issue 11, November 2023

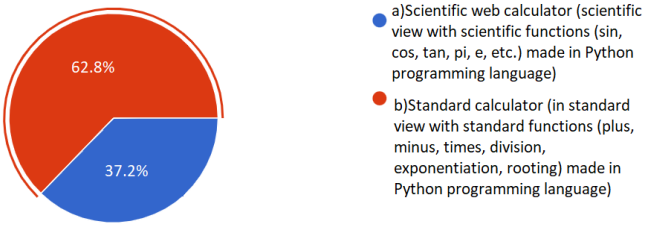

Figure 7. Results, whether a scientific web calculator made in Python programming language or a standard calculator is better for the respondents and their needs and daily use
Source: Authors.

Figure 7 shows that most respondents, 62.8%, chose a standard calculator (in the standard view with standard functions (plus, minus, times, division, exponentiation, rooting, etc.) made in the Python programming language) as better for their needs and daily use, while 37.2% of respondents chose a scientific web calculator.

IV. ANALYSIS OF RESULTS AND CHI-SQUARE TEST

In this chapter, we will look at the analysis of survey results and we will perform a procedure called the chi-square test, which is used in most cases if it is about qualitative data or if the data distribution deviates significantly from normal. At the very beginning, it should be emphasized that the chi-square test is calculated with frequencies, and it is not allowed to enter measurement units in the calculation. The basic research data can also be measured values, but only their frequencies are entered into the chi-square.

Table 1 shows the results of the chi-square tests for the answers to the 1st, 2nd, 3rd, 4th, and 5th questions of the online survey.

1st question: Did you know that applications created with the Python programming language can also be used as web applications if they are placed on a web server? 2nd question: Do you use the Python programming language when creating web applications? 3rd question: Did you know how the Python programming language is used to manipulate data in existing relations within the database? 4th question: Do you use the Python programming language to manipulate data in existing relationships within the database? 5th question: Is a web calculator made (in the Python programming language) better (and more necessary) for your needs and daily use: (a) Scientific, with functions sin, cos, tan, pi, e, etc.; (b) Standard, with functions plus, minus, times, divide, exponentiate, root, etc. (?)

Table 1. The results of the chi-square test, respectively, for the answers to the 1st, 2nd, 3rd, 4th, and 5th questions of the online survey

	Yes	No	Other	Total:	Divided:
f0	37	8		45	22.5000
ft	22.5	22.5		45	
	f0	ft	f0-ft	(f0-ft)2	Chi-square
	37	22.5000	14.5000	210.2500	9.3444
	8	22.5000	-14.5000	210.2500	9.3444
				Chi-square	18.6888888888889
	I don't use	I use	Other	Total:	Divided:
f0	37	9		46	23.0000
ft	23	23		46	
	f0	ft	f0-ft	(f0-ft)2	Chi-square
	37	23.0000	14.0000	196.0000	8.5217
	9	23.0000	-14.0000	196.0000	8.5217
				Chi-square	17.0434782608696
	Yes	No	Other	Total:	Divided:
f0	14	32		46	23.0000
ft	23	23		46	
	f0	ft	f0-ft	(f0-ft)2	Chi-square
	14	23.0000	-9.0000	81.0000	3.5217
	32	23.0000	9.0000	81.0000	3.5217
				Chi-square	7.04347826086957
	I don't use	I use	Other	Total:	Divided:
f0	42	4		46	23.0000
ft	23	23		46	
	f0	ft	f0-ft	(f0-ft)2	Chi-square
	42	23.0000	19.0000	361.0000	15.6957
	4	23.0000	-19.0000	361.0000	15.6957
				Chi-square	31.3913043478261
	a) Scientific Python web calculator (in the scientific sense with functions like sin, cos, tan, pi, e, etc.)	b) Standard Python web calculator (in the standard sense with functions of the type plus, minus, times, division, exponentiation, rooting, etc.)	Other	Total:	Divided:
f0	16	30		46	23.0000
ft	23	23		46	
	f0	ft	f0-ft	(f0-ft)2	Chi-square
	16	23.0000	-7.0000	49.0000	2.1304
	30	23.0000	7.0000	49.0000	2.1304
				Chi-square	4.26086956521739

Source: Authors, based on the Google Forms tool.

Based on the analysis of the chi-square test, there is a significant difference in the answers to the 1st, 2nd, 3rd, 4th, and 5th questions of the online survey between the answers obtained and the randomly distributed answers.

V. CONCLUSION

The reason for writing the paper was to present first information about an innovative software development approach to scientific functions for the application Tuned scientific calculator in Python. The research methodology was an innovative software development approach to scientific functions in the Python programming language. Based on the research in this paper on the topic: “Research on the use of Python in the development of an innovative scientific web application with developing scientific functionalities on the adapted scientific calculator”, we conclude that the work (application) is extremely important to support the calculation of more complex mathematical expressions. The form (original) for the scientific part of the calculator is especially interesting. Regardless of the company activity, they are engaged in, it is known that programming as a process of writing commands in a

particular programming language is used in almost all fields of profession and all fields of science especially in mathematics to develop applications that will improve processes or individual modules or companies (private or public corporations). The contribution of the paper can be seen by presenting the final application of the scientific calculator. Also, novelty, as the continuation of this research, we should investigate which innovative and scientific functionalities are used in similar application software solutions of the world, European, and nationally known corporations, which is the subject of future scientific research. Also, based on the results of the survey, we came to the conclusion that: (1) the majority of respondents, 83.7%, knew that applications created with the Python programming language can also be used as web applications if they are placed on a web server, (2) the largest number of respondents, 76.7%, is choosing not to use the Python programming language when creating web applications, (3) the largest number of respondents (67.4%) knew that the Python programming language is used to manipulate data in existing relations within the database, (4) the majority of respondents 90.7% did not use the programming language the Python language for manipulating data in existing relations within the database and (5) that the majority of respondents (62.8%) chose a standard calculator (in the standard view with standard functions (plus, minus, times, division, exponentiation, rooting, etc.) made by Python programming language) as better for their needs and daily use. The main research methodologies used here are the method of scientific and professional text analysis, the survey method, the chi-square test, and the comparative method. The pilot research presented here was accepted, based on the chi-squared tests, in all five cases (questions). This research will serve for the next, larger, and similar research.

REFERENCES

[1] 5th high school, "Introduction to Python (motivation and installation)", Fifth high school & EU, Zagreb, 2014 (in Croatian: 5. Gimnazija, "Uvod u Python (motivacija i instalacija)", Peta gimnazija & EU, Zagreb, 2014), http://ipaq.petagimnazija.hr/wp-content/uploads/2014/12/Uvod_u_Python.pdf

[2] Al Awar, N., Mehta, N., Zhu, S., Biros, G., & Gligoric, M., Citation: PyKokkos: Performance Portable Kernels in Python, 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 164-167, DOI: 10.1109/ICSE-Companion55297.2022.9793761, https://ieeexplore.ieee.org/document/9793761/authors#authors

[3] Ayaz, F., Sheng, Z., Tian, D., & Guan, Y.L., A Blockchain Based Federated Learning for Message Dissemination in Vehicular Networks, 2022 IEEE Transactions on Vehicular Technology, 71(2), 1927-1940, DOI: 10.1109/TVT.2021.3132226, https://ieeexplore.ieee.org/document/9633160

[4] GeeksforGeeks. "A computer science portal for geeks", (2023), https://www.geeksforgeeks.org/. Noida, Uttar Pradesh: GeeksforGeeks

[5] GOOGLE, "Google Forms," https://www.google.com/intl/hr/forms/about/

[6] H.J. Mohammed, and K.H.A Faraj, "Python-WSGI and PHP-Apache Web Server Performance Analysis by Search Page Generator (SPG)," vol. 5, pp. 132-138e, January 2021, DOI 10.25079/ukhjse.v5n1y2021.pp132-138e.v5n1y2021.ppxx-xx

[7] Jurić-Kavelj, S., Marković, I., & Miklić, D., Introduction to programming in Python, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, 2012, https://www.fer.unizg.hr/_download/repository/p02-python.pdf

[8] Linge, S., Langtangen, H.P., Citation: Loops and Branching. In: Programming for Computations - Python: A Gentle Introduction to Numerical Simulations with Python 3.6. Texts in Computational Science and Engineering, 2020, 15, 59-77, Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-16877-3_3

[9] Oliphant, T.E., Citation: Python for Scientific Computing. Computing in Science & Engineering, 2007, 9(3), 10-20. IEEE Xplore. DOI: 10.1109/MCSE.2007.58, https://ieeexplore.ieee.org/document/4160250

[10] Varga, M., Programming in the Python and Visual Basic programming languages: a collection of tasks, Association of financial and IT experts in Međimurje, Mursko Središće, 2018 (in Croatian: Varga, M., Programiranje u programskim jezicima Python i Visual Basic: zbirka zadataka, Udruga financijsko -informatičkih stručnjaka Međimurja, Mursko Središće: 2018)

[11] Zhuang, Y., Lu, M.-Y., Citation: Enabling Type Checking on Columns in Data Frame Libraries by Abstract Interpretation, 2022 IEEE Access, 10, 14418-14428, DOI: 10.1109/ACCESS.2022.3146287, https://ieeexplore.ieee.org/document/9691374