

| വർക്ക് ഷീറ്റ് 1   |   |
|---|---|
| പ്രവർത്തനത്തിന്റെ പേര്  | പൈത്തൺ ഉപയോഗിച്ചുള്ള ഗണിത ക്രിയകൾ                                 |
| പ്രവർത്തന ഉദ്ദേശ്യങ്ങൾ  | അടിസ്ഥാന ക്രിയകൾ ചെയ്യുന്നതിനുള്ള പൈത്തൺ രീതി മനസ്സിലാക്കുന്നതിന് |
| ഉപയോഗിക്കേണ്ട സോഫ്റ്റ് വെയർ   | Python - Idle   |
| പ്രവർത്തനങ്ങൾ   |   |
| സോഫ്റ്റ് വെയർ തുറക്കുന്നു   | Application – programming – idle – file - new                     |
| രണ്ട് സംഖ്യകളുടെ തുക കാണുന്നത്  | Print 45+34 ( space after print)                                  |
| ഒരു സംഖ്യയിൽ നിന്ന് മറ്റൊരു സംഖ്യ കുറയ്ക്കുന്നത്  | Print 45-34 ( space after print)                                  |
| രണ്ട് സംഖ്യകളുടെ ഗുണന ഫലം കാണുന്നത്   | Print 45*34 ( space after print)                                  |
| കൃതി കാണുന്നത്  | Print 8**3 ( space after print)                                   |
| ഹരണ ക്രിയയിൽ ശിഷ്ടം കാണുന്നത്   | Print 25%4 ( space after print)                                   |
| <p>ഈ പ്രവർത്തനങ്ങൾ പൈത്തൺ ഷെല്ലിലാണ് പ്രവർത്തിപ്പിക്കേണ്ടത്(&gt;&gt;&gt; ഈ ചരത്തിന് ശേഷം commands type ചെയ്ത് enter അടിച്ചാൽ മതി. Save ചെയ്യണമെന്നില്ല.) ഉത്തരം ഗ്രാഫിക്സ് ആണെങ്കിൽ turtle graphics ജാലകത്തിലും അല്ലെങ്കിൽ python shell ജാലകത്തിലും പ്രത്യക്ഷപ്പെടും.</p> |   |

Prepared by Howlath.k  
CKHS Manimooly

വർക്ക് ഷീറ്റ് 2

പൈത്തൺ പ്രോഗ്രാമിൽ range എന്ന സങ്കേതത്തിന്റെ പ്രവർത്തനം

പൈത്തണിൽ സംഖ്യകളുടെ സമാന്തര ശ്രേണികൾ നിർമ്മിക്കാനുള്ള ഉപാധിയാണ് range

മാതൃകാ പ്രോഗ്രാമുകൾ

|  |   |
|--|---|
| Range (10) ( space after range)<br>range (30) ( small letter keep) | Output [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]<br>[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29] |
| Range (1,10)<br>range (1,10,2)                                     | [1, 2, 3, 4, 5, 6, 7, 8, 9]<br>[1, 3, 5, 7, 9]  |
| Range (17)<br>range (0,-10,-1)<br>range (0,65,7)                   | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]<br>[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]<br>[0, 7, 14, 21, 28, 35, 42, 49, 56, 63]         |

ലക്ഷ്യം - സമാന്തര ശ്രേണികൾ മനസ്സിലാക്കുക

For എന്ന നിർദ്ദേശത്തിന്റെ പ്രവർത്തനം {small letter keep}

```
from turtle import*
clear()
pencolor("blue")
pensize(8)
for i in range(8):
    rt(45)
    circle(50)
    circle(60)
    circle(70)
```

for നിർദ്ദേശത്തിനകത്ത് വീണ്ടും for നിർദ്ദേശം നൽകുന്ന പ്രവർത്തനം

Page no 39

```
from turtle import*
clear()
pencolor("green")
pensize("7")
for i in range(10):
    circle(30)
    circle(40)
    circle(50)
    circle(60)
    circle(70) circle(80)
```

For ന്റെ പകരം while ഉപയോഗിച്ചാൽ {small letter keep}

```
from turtle import*
clear()
pencolor("black")
pensize("6")
i=0
while(i<8):
    circle(50)
    i=i+1
```

```
from turtle import*
clear()
pencolor("orange")
pensize("5")
for i in range(8):
    rt(50)
    for j in range(30,91,10):
        circle(j)
```

ഒരേ പ്രോഗ്രാം നിശ്ചിത തവണ ആവർത്തിച്ചു പ്രവർത്തിക്കാൻ നൽകുന്ന നിർദ്ദേശമാണ് for.

While നിർദ്ദേശം നൽകിയിരിക്കുന്ന പ്രോഗ്രാമിൽ i=i+1 എന്ന command നൽകിയില്ലെങ്കിൽ

സർക്കിൾകറങ്ങി കൊണ്ടേയിരിക്കും. i=0 സർക്കിൾ കറങ്ങാനുള്ള നിർദ്ദേശം നൽകുന്നു.

{ വർക്ക് ഷീറ്റ് 3 മുതലുള്ള പ്രവർത്തനങ്ങൾ പൈത്തണിന്റെ പുതിയ ജാലകത്തിൽ ചെയ്ത് സേവ് ചെയ്യേണ്ടതാണ് }

വർക്ക് ഷീറ്റ് 4

ിരണ്ട് സംഖ്യകളുടെ തുക കാണുന്നതിനുള്ള ഫങ്ങഷനുകളു്

Application – programming – idle - file -new window എന്ന ക്രമത്തിൽ സോഫ്റ്റ് വെയർ തുറന്ന് താഴെ പറയുന്ന പ്രവർത്തനങ്ങൾ ഓരോന്നായി ടൈപ്പ് ചെയ്യുക

Page no 40 ലെ പ്രവർത്തനങ്ങൾ

```
def sum(a,b):
    c=a+b
    return c
```

(file – save- run – run - module / f5 – തുറന്ന് വരുന്ന പൈത്തൺ ഷെല്ലിൽ വില sum(15,16 )നൽകി enter അടിക്കുക)

```
def sub(a,b):
    s=a-b
    return s
```

(file – save- run – run - module / f5 – തുറന്ന് വരുന്ന പൈത്തൺ ഷെല്ലിൽ വില sub(15,16 )നൽകി enter അടിക്കുക ) വീണ്ടും സേവ് ചെയ്യാൻ save option click.

```
def pro(a,b):
    p=a*b
    return p {ഗുണന ഫലം}
```

(file – save- run – run - module / f5 – തുറന്ന് വരുന്ന പൈത്തൺ ഷെല്ലിൽ വില pro(15,16 )നൽകി enter അടിക്കുക ) വീണ്ടും സേവ് ചെയ്യാൻ save option click.

```
def power(a,b):
    q=a**b
    return q (കൃതി)
```

(file – save- run – run - module / f5 – തുറന്ന് വരുന്ന പൈത്തൺ ഷെല്ലിൽ വില power(15,16 )നൽകി enter അടിക്കുക )വീണ്ടും സേവ് ചെയ്യാൻ save option click.

വർക്ക് ഷീറ്റ് 5

സ്മിത്ത് പിരമിഡുകളുടെ പ്രവർത്തനം Page no 41,42 ലെ പ്രവർത്തനങ്ങൾ

ഒരക്ഷരം നിശ്ചിത തവണ പിരമിഡ് രൂപത്തിൽ for സങ്കേതം ഉപയോഗിച്ച് പ്രവർത്തിപ്പിക്കുന്ന പ്രോഗ്രാം  
 Application – programming – idle - file -new window എന്ന ക്രമത്തിൽ സോഫ്റ്റ് വെയർ തുറന്ന് താഴെ പറയുന്ന പ്രവർത്തനങ്ങൾ ഓരോന്നായി ടൈപ്പ് ചെയ്യുക  
 a="A"  
 for i in range(1,15):  
     print i\*a  
 (file – save- run – run - module / f5 പൈത്തൺ ഷെല്ലിൽ ഔട്ട് പൂട്ട് കാണാം)

A  
 AA  
 AAA  
 AAAA  
 AAAAA  
 AAAAAA  
 AAAAAAA  
 AAAAAAAA  
 AAAAAAAAA  
 AAAAAAAAAA  
 AAAAAAAAAA  
 AAAAAAAAAA  
 AAAAAAAAAA  
 AAAAAAAAAA  
 AAAAAAAAAA  
 AAAAAAAAAA

നിങ്ങളുടെ വിദ്യാലയത്തിന്റെ പേര് ഇൻപുട്ടായി സ്വീകരിച്ച് ആദ്യാക്ഷരം മുതൽ അവസാനം വരെ എണ്ണി തിട്ടപ്പെടുത്തി പ്രവർത്തിപ്പിക്കുന്ന പ്രോഗ്രാം  
 a=("CKHSMANIMOOLY")  
 n=len(a)  
 for i in range(n+1):  
     print a[:i]

C  
 CK  
 CKH  
 CKHS  
 CKHSM  
 CKHSMA  
 CKHSMAN  
 CKHSMANI  
 CKHSMANIM  
 CKHSMANIMO  
 CKHSMANIMOO  
 CKHSMANIMOOL  
 CKHSMANIMOOLY

ഈ പ്രവർത്തനത്തിൽ while എന്ന സങ്കേതത്തിന്റെ ഉപയോഗം  
 a=("MERABHARATH")

M  
 ME  
 MER  
 MERA

```

n=len(a)
i=1
while(i<n+1):
    print a[:i]
    i=i+1

def sum(a,b):
    c=a+b
    return c
import myfunction
print myfunction.sum(3,4)
print myfunction.sum('3','4')
print myfunction.sum(30,40)
print myfunction.sum('CKHS','MANIMOOLY')

```

(page no 40,41)

```

MERAB
MERABH
MERABHA
MERABHAR
MERABHARA
MERABHARAT
MERABHARATH

```

(OUT PUT)  
7  
34  
70  
CKHSMANIMOOLY

```

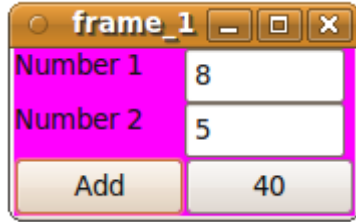
(def sum(a,b):
    c=a+b
    return c ഈ COMMANDS

```

PYTHON new window യിൽ ടൈപ്പ് ചെയ്ത് നിങ്ങളുടെ ഫോൾഡറിലേക്ക് myfunction എന്ന പേരിൽ സേവ് ചെയ്യുക.ശേഷം അതേ ജാലകത്തിൽ ബാക്കി commands type ചെയ്ത് Ctrl+s button click ചെയ്ത് run ചെയ്യിക്കുക)

വർക്ക ഷീറ്റ് 6. താഴെ കൊടുത്തിരിക്കുന്ന മാതൃക WxGlade soft wear ൽ തയ്യാറാക്കുക.

മാതൃക




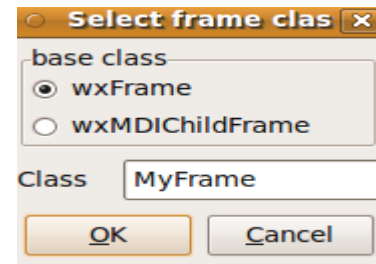
പ്രവർത്തനം - പൈത്തൺ കോഡ് ജനറേറ്റ് ചെയ്യൽ

സോഫ്റ്റ് വെയർ പ്രവർത്തന സജ്ജമാക്കുക

Application – programming – WxGlade എന്ന ക്രമത്തിൽ സോഫ്റ്റ് വെയർ തുറക്കുക. (tools window, properties window, Design Frame window എന്നിങ്ങനെ മൂന്ന് ജാലകങ്ങൾ ഡെസ്ക്ടോപ്പിൽ പ്രത്യക്ഷപ്പെടുന്നു )

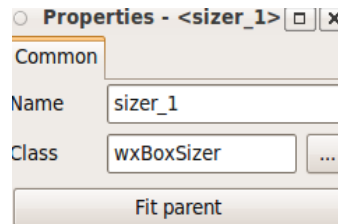
- ഹെഡിംഗ് നിർമ്മാണം

WxGlade ന്റെ ടൂൾ ജാലകത്തിൽ നിന്ന് Add a Frame button (  ) സെലക്ട് ചെയ്ത് തുറന്ന് വരുന്ന Frame Class ബോക്സിലെ wxFrame option ക്ലിക്ക് ചെയ്ത് ok നൽകുക



- വരിയും നിരയും ക്രമീകരിക്കാൻ

WxGlade :Tree ജാലകത്തിലെ SiZer സെലക്ട് ചെയ്ത് Properties ജാലകത്തിലെ class ന് നേരേയുള്ള ബട്ടണിൽ ക്ലിക്ക് ചെയ്യുക.



Select SiZer Type window യിൽ നിന്ന് WxFlexGridSizer എന്നത് സെലക്ട് ചെയ്ത് ok നൽകുക. തുടർന്ന് tree ജാലകത്തിലെ SiZer ൽ right button click ചെയ്ത് ആവശ്യമായ വരിയും നിരയും നൽകുക.

| <ul style="list-style-type: none"> <li>Font Size, Font colour എന്നിവ നൽകുന്നതിന്</li> </ul> | <p>WxGlade :Tree ജാലകത്തിലെFrame 1 സെലക്ട് ചെയ്ത് Properties ജാലകത്തിലെ Background, Foreground,Font എന്നിവ ടിക് മാർക്ക് നൽകി അനുയോജ്യമായ നിറവും ഫോണ്ടും നൽകുക</p>  |       |         |            |     |
|---|--|-------|---------|------------|-----|
| <ul style="list-style-type: none"> <li>ലേബൽ നിർമ്മിക്കുന്ന വിധം</li> </ul>                  | <p>WxGlade ന്റെ ടൂൾ ജാലകത്തിൽ നിന്ന് Add a static Text button ( <b>A</b> ) സെലക്ട് ചെയ്ത് ഡിസൈൻ ജാലകത്തിലെ ആദ്യത്തെ കളത്തിൽ ക്ലിക്ക് ചെയ്യുക. Properties ജാലകത്തിലെ Widget ടാബിൽ Label _1 എന്നത് Number 1 എന്നാക്കി ആദ്യത്തെ കളത്തിൽ വീണ്ടും ക്ലിക്ക് ചെയ്യുക. രണ്ടാമത്തെ കളവും ഇതേ ക്രമത്തിൽ മാറ്റുക.</p>   |       |         |            |     |
| <ul style="list-style-type: none"> <li>ടെക്സ്റ്റ് ബോക്സ് ചേർക്കുന്ന വിധം</li> </ul>         | <p>WxGlade ന്റെ ടൂൾ ജാലകത്തിൽ നിന്ന് Add a Text Ctrl button ( <b>[ab]</b> ) സെലക്ട് ചെയ്ത് ടെക്സ്റ്റ് ചേർക്കേണ്ട കളങ്ങളിൽ ക്ലിക്ക് ചെയ്യുക</p>   |       |         |            |     |
| <ul style="list-style-type: none"> <li>Add ബട്ടണുകൾ ചേർക്കാൻ</li> </ul>                     | <p>WxGlade ന്റെ ടൂൾ ജാലകത്തിൽ നിന്ന് Add a Button ( <b>OK</b> ) സെലക്ട് ചെയ്ത് ബട്ടൺ ചേർക്കേണ്ട കളങ്ങളിൽ ആദ്യത്തെ കളത്തിൽ ക്ലിക്ക് ചെയ്യുക തുടർന്ന് Properties ജാലകത്തിലെ Widget ടാബിൽ Button _1 എന്നത് Add/sum എന്നാക്കി ആദ്യത്തെ കളത്തിൽ വീണ്ടും ക്ലിക്ക് ചെയ്യുക. Widget ന് നേരേയുള്ള ആരോ കീ ക്ലിക്ക് ചെയ്ത് Handler എന്നിടത്ത് Add എന്ന് ടൈപ്പ് ചെയ്യുക.</p> <div data-bbox="1032 1013 1512 1252" data-label="Image"> <table border="1" data-bbox="1032 1013 1512 1252"> <thead> <tr> <th>Event</th> <th>Handler</th> </tr> </thead> <tbody> <tr> <td>EVT_BUTTON</td> <td>add</td> </tr> </tbody> </table> </div> <p>വീണ്ടും Add a Button സെലക്ട് ചെയ്ത് ഉത്തരം തെളിയേണ്ട കളത്തിൽ ക്ലിക്ക് ചെയ്ത് Widget ടാബിൽ Button _2 എന്നത് ഒഴിവാക്കുക</p> | Event | Handler | EVT_BUTTON | add |
| Event   | Handler  |       |         |            |     |
| EVT_BUTTON  | add  |       |         |            |     |
| <ul style="list-style-type: none"> <li>പൈത്തൺ കോഡ് സേവ് ചെയ്യുന്ന വിധം</li> </ul>           | <p>WxGlade :Tree ജാലകത്തിലെ Application ക്ലിക്ക് ചെയ്യുക. തുടർന്ന് Properties ജാലകത്തിലെ Output path ഒപ്ഷൻ നേരേയുള്ള ബട്ടണിൽ ക്ലിക്ക് ചെയ്ത്</p>   |       |         |            |     |



|   |   |
|---|---|
|   | <p>ആവശ്യമായ ഫോൾഡറിലേക്ക് .py എന്ന എക്സ്റ്റൻഷനോട് കൂടി ഫയൽ നാമം നൽകി സേവ് ചെയ്യുക. ശേഷം Properties ജാലകത്തിലെ Python എന്ന ഒപ്ഷൻ സെലക്ട് ചെയ്ത് Generate Code ബട്ടണിൽ അമർത്തുക. (<b>Generate Code നൽകുന്നതിന് മുൻപ് WxGlade ജാലകം ക്ലോസ് ചെയ്യരുത്</b>)</p>   |
| <ul style="list-style-type: none"> <li>പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്ന വിധം</li> </ul> | <p>സേവ് ചെയ്ത ഫയൽ ഓപ്പൺ ചെയ്യുക. Display ബട്ടൺ ക്ലിക്ക് ചെയ്താൽ ജനറേറ്റ് ചെയ്ത പൈത്തൺ കോഡുകൾ gedit ജാലകത്തിൽ തുറന്ന് വരും.</p> <p>ഇവിടെ <code>print "Event handler `add' not implemented!"</code> എന്ന ഭാഗം ഒഴിവാക്കി</p> <pre>c=int(self.text_ctrl_1.Value)*int(self.text_ctrl_2.Value) self.button_2.SetLabel(str(c))</pre> <p>എന്ന കമാന്റ് നൽകുക.</p> <p>Save ബട്ടൺ ക്ലിക്ക് ചെയ്ത് ജാലകം ക്ലോസ് ചെയ്യുക. തുടർന്ന് ഫയൽ Run in Terminal ൽ തുടർന്ന് പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുക. (Text box ൽ നമ്പറുകൾ ചേർത്ത് Add ബട്ടൺ ക്ലിക്ക് ചെയ്യുക.</p> |

Prepared by Howlath.k CKHS MANIMOOLY