```python
#Notepad
#Icon http://www.doublejdesign.co.uk
import tkinter
from PIL import ImageTk, Image
from tkinter import StringVar, IntVar, scrolledtext, END, messagebox, filedialog

#Define window
root = tkinter.Tk()
root.title('Notepad')
root.iconbitmap('pad.ico')
root.geometry('600x600')
root.resizable(0,0)

#Define fonts and colors
text_color = "#fffacd"
menu_color = "#dbd9db"
root_color = "#6c809a"
root.config(bg=root_color)

#Define functions
def change_font(event):
    """Change the given font based off dropbox options."""
    if font_option.get() == 'none':
        my_font = (font_family.get(), font_size.get())
    else:
        my_font = (font_family.get(), font_size.get(), font_option.get())

    #Change the font style
    input_text.config(font=my_font)


def new_note():
    """Create a new Note which essentially clears the screen."""
    #Use a messagebox to ask for a new note
    question = messagebox.askyesno("New Note", "Are you sure you want to start a new note?")
    if question == 1:
        #ScrolledText widgets starting index is 1.0 not 0.
        input_text.delete("1.0", END)


def close_note():
    """Closes the note which essentially quits the program."""
    #Use a messagebox to ask to close
    question = messagebox.askyesno("Close Note", "Are you sure you want to close your note?")
    if question == 1:
        root.destroy()


def save_note():
    """Save the given note.  First three lines are saved as font family, font size, and font option."""
    #Use filedialog to get location and name of where/what to save the file as.
    save_name = filedialog.asksaveasfilename(initialdir="./", title="Save Note", filetypes=(("Text Files", "*.txt"), ("All Files", "*.*")))
    with open(save_name, 'w') as f:
        #First three lines of save file are font_family, font_size, and font_options.  Font_size must be a string noot int.
        f.write(font_family.get() + "\n")
        f.write(str(font_size.get()) + "\n")
        f.write(font_option.get() + "\n")

        #write remaining text in field to the file
        f.write(input_text.get("1.0", END))


```

```
63  def open_note():
64      """Open a previously saved note.  First three lines of note are font family, font
        size, and font option.  First set the font, then load the text."""
65      #Use filedialog to get location and directory of note file
66      open_name = filedialog.askopenfilename(initialdir="./", title='Open Note',
        filetypes=(("Text Files", "*.txt"), ("All Files", "*.*")))
67      with open(open_name, 'r') as f:
68          #Clear the current text
69          input_text.delete("1.0", END)
70
71          #First three lines are font_faimly, font_size, and font_option...You must strip
            the new line char at the end of each line!
72          font_family.set(f.readline().strip())
73          font_size.set(int(f.readline().strip()))
74          font_option.set(f.readline().strip())
75
76          #Call the change font for these .set() and pass an arbitrary value
77          change_font(1)
78
79          #Read the rest of the file and insert it into the text field
80          text = f.read()
81          input_text.insert("1.0", text)
82
83
84  #Define Layout
85  #Define frames
86  menu_frame = tkinter.Frame(root, bg=menu_color)
87  text_frame = tkinter.Frame(root, bg=text_color)
88  menu_frame.pack(padx=5, pady=5)
89  text_frame.pack(padx=5, pady=5)
90
91  #Layout for menu frame
92  #Create the menu:  new, open, save, close, font family, font size, font option
93  new_image = ImageTk.PhotoImage(Image.open('new.png'))
94  new_button = tkinter.Button(menu_frame, image=new_image, command=new_note)
95  new_button.grid(row=0, column=0, padx=5, pady=5)
96
97  open_image = ImageTk.PhotoImage(Image.open('open.png'))
98  open_button = tkinter.Button(menu_frame, image=open_image, command=open_note)
99  open_button.grid(row=0, column=1, padx=5, pady=5)
100
101 save_image = ImageTk.PhotoImage(Image.open('save.png'))
102 save_button = tkinter.Button(menu_frame, image=save_image, command=save_note)
103 save_button.grid(row=0, column=2, padx=5, pady=5)
104
105 close_image = ImageTk.PhotoImage(Image.open('close.png'))
106 close_button = tkinter.Button(menu_frame, image=close_image, command=close_note)
107 close_button.grid(row=0, column=3, padx=5, pady=5)
108
109 #Create a list of fonts to use
110 families = ['Terminal', 'Modern', 'Script', 'Courier', 'Arial', 'Calibri', 'Cambria',
    'Georgia', 'MS Gothic', 'SimSun', 'Tahoma', 'Times New Roman', 'Verdana', 'Wingdings']
111 font_family = StringVar()
112 font_family_drop = tkinter.OptionMenu(menu_frame, font_family, *families,
    command=change_font)
113 font_family.set('Terminal')
114 #Set the width so it will fit "times new roman" and remain constant
115 font_family_drop.config(width=16)
116 font_family_drop.grid(row=0, column=4, padx=5, pady=5)
117
118 sizes = [8, 10, 12, 14, 16, 20, 24, 32, 48, 64, 72, 96]
119 font_size = IntVar()
120 font_size_drop = tkinter.OptionMenu(menu_frame, font_size, *sizes, command=change_font)
121 font_size.set(12)
122 #Set width to be constant even if its 8.
123 font_size_drop.config(width=2)
124 font_size_drop.grid(row=0, column=5, padx=5, pady=5)
```

```
125
126     options = ['none', 'bold', 'italic']
127     font_option = StringVar()
128     font_option_drop = tkinter.OptionMenu(menu_frame, font_option, *options,
        command=change_font)
129     font_option.set('none')
130     #Set the width to be constant
131     font_option_drop.config(width=5)
132     font_option_drop.grid(row=0, column=6, padx=5, pady=5)
133
134     #Layout for the text frame
135     my_font = (font_family.get(), font_size.get())
136
137     #Create input_text as a scrolltext so you can scroll through the text field.
138     #Set default width and height to be more than the window size so that on the smallest
        text size, the text field size is constant.
139     input_text = tkinter.scrolledtext.ScrolledText(text_frame, width=1000, height=100,
        bg=text_color, font=my_font)
140     input_text.pack()
141
142     #Run the root window's main loop
143     root.mainloop()
```