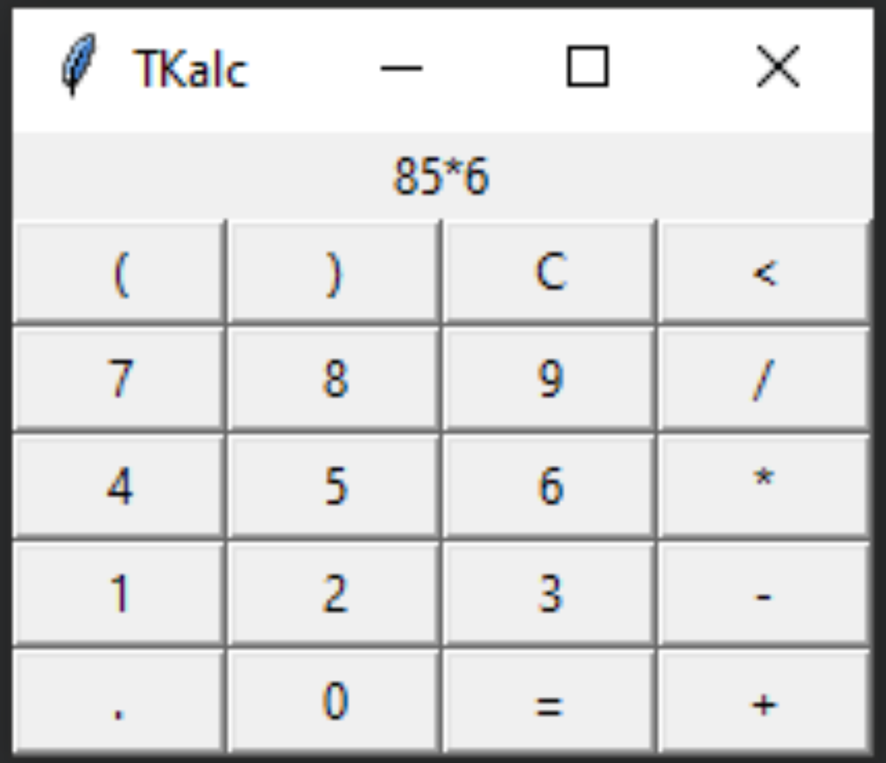
Elegant

(line of)

code ⚡

by Rodrigo Girão Serrão


TKalc
85*6
(
)
C
<
7
8
9
/
4
5
6
*
1
2
3
-
.
0
=
+

```python
from tkinter import *

def bt_draw(key, col, lin):
    bt = Button(window, text=key, command=lambda: bt_press(key))
    bt.grid(column=col+1, row=lin+1)
    return bt

def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

window = Tk()
window.title("TKalc")

disp = Label(window, text="")
print(disp.grid(column=0, row=0, columnspan=5))

keys = "()C<789/456*123-.0=+"
bt_list = [bt_draw(keys[n], n%4, n//4) for n in range(20)]

window.mainloop()
```

Remove lines

```
from tkinter import *

def bt_draw(key, col, lin):
    bt = Button(window, text=key, command=lambda: bt_press(key))
    bt.grid(column=col+1, row=lin+1)
    return bt

def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

window = Tk()
window.title("TKalc")

disp = Label(window, text="")
print(disp.grid(column=0, row=0, columnspan=5))

keys = "()C<789/456*123-.0=+"
bt_list = [bt_draw(keys[n], n%4, n//4) for n in range(20)]

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    bt = Button(window, text=key, command=lambda: bt_press(key))
    bt.grid(column=col+1, row=lin+1)

def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

window = Tk()
window.title("TKalc")

disp = Label(window, text="")
disp.grid(column=0, row=0, columnspan=5)

bt_list = [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")]

window.mainloop()
```

Statements, bad Expressions, good!

```
from tkinter import *

def bt_draw(key, col, lin):
    bt = Button(window, text=key, command=lambda: bt_press(key))
    bt.grid(column=col+1, row=lin+1)

def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

window = Tk()
window.title("TKalc")

disp = Label(window, text="")
disp.grid(column=0, row=0, columnspan=5)

bt_list = [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")]

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

Aux function to assign to disp

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

Use update

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    if   key == "C": disp["text"] = ""
    elif key == "<": disp["text"] = disp["text"][:-1]
    elif key == "=": disp["text"] = str(round(eval(disp["text"]), 6))
    else           : disp["text"] += key

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    if key == "C": update("")
    elif key == "<": update(disp["text"][:-1])
    elif key == "=": update(str(round(eval(disp["text"]), 6)))
    else: update(disp["text"] + key)

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

Call update once

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    if key == "C": update("")
    elif key == "<": update(disp["text"][:-1])
    elif key == "=": update(str(round(eval(disp["text"]), 6)))
    else: update(disp["text"] + key)

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    if key == "C": arg = ""
    elif key == "<": arg = disp["text"][:-1]
    elif key == "=": arg = str(round(eval(disp["text"]), 6))
    else: arg = disp["text"] + key
    update(arg)

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

Rewrite `arg` assignment w/ conditional expression

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    if key == "C": arg = ""
    elif key == "<": arg = disp["text"][:-1]
    elif key == "=": arg = str(round(eval(disp["text"]), 6))
    else: arg = disp["text"] + key
    update(arg)

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    update(
        "" if key == "C" else
        disp["text"][:-1] if key == "<" else
        str(round(eval(disp["text"]), 6)) if key == "=" else
        disp["text"] + key
    )

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])
```

Flatten out and turn into `lambda`

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
def bt_press(key):
    update(
        "" if key == "C" else
        disp["text"][:-1] if key == "<" else
        str(round(eval(disp["text"]), 6)) if key == "=" else
        disp["text"] + key
    )

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])
```

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if key == "<" else s

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/456*123-.0=+")])

window.mainloop()
```

```python
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if k

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/45

window.mainloop()
```

Get rid of import statement...

```
from tkinter import *

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if k

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/45

window.mainloop()
```

```
(tk := __import__("tkinter"))

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if k

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/45

window.mainloop()
```

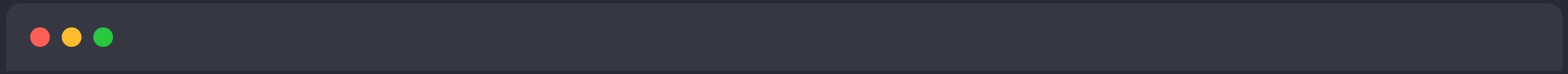
Simplify final function w/ long-circuiting

```
(tk := __import__("tkinter"))

def bt_draw(key, col, lin):
    (bt := Button(window, text=key, command=lambda: bt_press(key)))
    bt.grid(column=col+1, row=lin+1)

(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if k

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/45

window.mainloop()
```

```python
(tk := __import__("tkinter"))

def bt_draw(key, col, lin):
    (bt := tk.Button(window, text=key, command=lambda: bt_press(key))) and b

(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if k

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/45

window.mainloop()
```

```
(tk := __import__("tkinter"))

(bt_draw := lambda key, col, lin: (bt := tk.Button(window, text=key, comman
(update := lambda t: disp.config(text=t))
(bt_press := lambda key: update("" if key == "C" else disp["text"][:-1] if k

(window := Tk())
window.title("TKalc")

(disp := Label(window, text=""))
disp.grid(column=0, row=0, columnspan=5)

(bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789/45

window.mainloop()
```

```
((tk := __import__("tkinter"))

and (bt_draw := lambda key, col, lin: (bt := tk.Button(window, text=key, co

and (update := lambda t: disp.config(text=t))
and (bt_press := lambda key: update("" if key == "C" else disp["text"][:-1]

and (window := Tk())
and window.title("TKalc")

or (disp := Label(window, text=""))
and disp.grid(column=0, row=0, columnspan=5)

or (bt_list := [bt_draw(k, n%4 + 1, n//4 + 1) for n, k in enumerate("()C<789

and window.mainloop())
```



```
# A calculator in a single expression of Python
(tk := __import__("tkinter")) and (bt_draw := lambda key, col, lin:
(bt := tk.Button(window, text=key, command=lambda: bt_press(key)))
and bt.grid(column=col + 1, row=lin + 1)) and (update := lambda t:
disp.config(text=t)) and (bt_press := lambda key: update("" if key
== "C" else disp["text"][:-1] if key == "<" else str(round(eval(
disp["text"]), 6)) if key == "=" else disp["text"] + key)) and (
window := tk.Tk()) and window.title("TKalc") or (disp := tk.Label(
window, text="")) and disp.grid(column=0, row=0, columnspan=5) or (
bt_list := [bt_draw(k, n % 4 + 1, n // 4 + 1) for n, k in enumerate
("()C<789/456*123-.0=+")]) and window.mainloop()
```