

technocamps

Arfer Gorau a Chanllawiau Python

ar gyfer TGAU Cyfrifiadureg



Llyfr Gwaith Ymarferwr



Llywodraeth Cymru
Welsh Government



Prifysgol
Abertawe
Swansea
University



Cardiff
Metropolitan
University

Prifysgol
Metropolitan
Caerdydd

University of
South Wales
Prifysgol
De Cymru



Y Comisiwn Addysg Drydyddol ac Ymchwil
Commission for Tertiary Education and Research



institute of
CODING
in wales

technocamps



PRIFYSGOL
ABERYSTWYTH
UNIVERSITY



Prifysgol Wrexham
Wrexham University



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Cyflwyniad

technocamps

Croeso i Lyfr Gwaith Ymarferwr Python! Bwriad y llyfr hwn yw bod yn llawlyfr defnyddiol i addysgu TGAU Cyfrifiadureg. Yn y llyfr hwn fe welwch 3 rhaglen; gêm Hangman gymharol syml i helpu i ymgysylltu'r dysgwyr wrth ddysgu codio, rhaglen siop syml sy'n dangos yr holl hanfodion, a rhaglen siop Tkinter fwy cymhleth sydd wedi torri ac y mae'n rhaid ei thrwsio. Trwy ddefnyddio'r 3 rhaglen hyn a mwy, rydym yn gobeithio y gall

pob dysgwr deimlo'n ddigon hyderus yn Python i drwsio rhaglen yn llwyddiannus yn ystod eu harholiad! Yn ogystal, mae tudalennau wedi'u neilltuo i arferion gorau a chonfensiynau o fewn Python - er efallai nad yw'r rhain yn gwbl ofynnol ar gyfer TGAU, trwy lynu wrthynt byddwch yn paratoi eich dysgwyr ar gyfer dyfodol mewn technoleg.

Deall y Cod

- Mae Cod Trwm yn god newydd i'w ychwanegu/cywiro.
- Cod Italig yw cod sydd yn gallu/dylai cael ei ysgrifennu ar un llinell - GALL cod rannu dros sawl linell cyn belled â bod y llinell yn gorffen gyda "chysylltydd" h.y. (neu , neu + ac ati.
- Mae'r tildau (~ ~ ~) yn dynodi lle mae mwy o god na chafodd ei argraffu
- Mae'r cod wedi'i liwio er mwyn ei ddarllenadwyedd ac mae ganddo fwy o liwiau nag yn IDLE.
- Mae mewnoliad cywir y cod yn angenrheidiol.

Deunyddiau Ar-lein

Mae'r deunyddiau i gyd ar gael ar-lein ar ein wefan! Sganiwch y cod QR neu ewch i tc1.me/educonf2025



Cysylltwch â ni!

Byddem yn hapus i drafod unrhyw gwestiynau neu ymholiadau ynglyn â chynnwys y llyfr gwaith hwn. Ebostiwch ni yn info@technocamps.com

Cynnwys

Arfer Gorau Python	3
Hangman	8
PyShop	24
Twyll-len PyShop	41
Trwsio tkinter	44
Twyll-len Tkinter	58

Arfer Gorau Python (Canllaw PEP8)

Arfer Enwi

Newidynnau:	<code>fy_newidyn_synhwyrol</code>
Cysonion:	<code>FY_NGHYSONYN_SYNHWYROL</code>
Ffwythiannau:	<code>fy_ffwythiant_synhwyrol</code>
Modiwlau a Phecynnau:	<code>fymodiwl_synhwyrol</code> neu <code>fy_modiwl_synhwyrol</code>
Dosbarthiadau:	<code>FyNosbarthSynhwyrol</code>

Gall Modiwlau
a Phecynnau
ddefnyddio tanlinell
os yw'n haws i
ddarllen - ond
dylent fod mewn
llythrennau bach yn
unig

Sut Dylai Edrych

Dim gofod gwyn diangen:

`adio(x, y)` **nid** `adio (x , y)`

Tabiau Nid Bylchau!

Mae hyn mewn gwirionedd y gwrthwyneb i'r arfer, ble dylai mewnnoliad fod yn union 4 bylchau bob amser, ond mae hynny'n anoddach i ddysgwyr ei ddeall!

Ni ddylai hyd y llinell fod yn fwy na 80 nod!

Mae hyn er mwyn gwneud eich cod yn ddarllenadwy ar sgriniau bach / mewn golwg ffenestr / sgriniau fertigol / pan gaiff ei fformatio mewn porwyr ac ati...

Dim Atalnodau

```
import math
import random

nid      import math, random

enw = "dan"
oed = 30

nid      enw, oed = "dan", 30
```

Gall Python wneud hyn ond mae'n cael ei ystyried yn arfer gwael! Mae amgylchiadau penodol lle mae'n cael ei dderbyn.

Dim Rhifau Hud

```
ENW_SYNHWYROL = 30
if fy_newidyn > ENW_SYNHWYROL:

nid      if fy_newidyn > 30:
```

Caniateir 0 ac 1 fel rhifau hud!

Dim Gwallau

Dylid ymdrin ag unrhyw wallau posibl bob amser!

```
try:
    canlyniad = 10 / 0
except ZeroDivisionError as e:
    print(f"Gwall: {e}")
```

Sylwadau

Mae defnydd da o sylwadau yn hanfodol!

Dylai hyn fod o fewn rheswm - nid oes angen sylw ar bob llinell o god, dylai'r cod ei hun fod yn amlwg. Fodd bynnag, efallai y bydd angen eglurhad ar gyfer iteriadau/dewisiadau cymhleth.

```
# Ailadrodd trwy'r rhestr ac yn gwneud rhywbeth
```

Defnyddio DocStrings!

Ar frig pob modiwl, ffwythiant, a dosbarth, disgrifiwch bwrpas y modiwl, ffwythiant neu'r dosbarth hwnnw! Ni ddylai hyn egluro'r cod, dim ond y syniad cyffredinol, pa bamedrau sydd rhaid darparu, a'r hyn y mae'n ei ddychwelyd!

```
"""
Mae'r ffwythiant hon yn gwneud y peth hwn.
Args: n (int): Mae'n ei wneud i'r cyfanrif n
Returns: m (int): Mae'n dychwelyd y peth hwnnw
"""
```

Terfynnau Bach

Dylai fod gan Ffwythiannau a Modiwlau gwmpas cyfyngedig bach!

Mae'n well cael llawer o ffwythiannau bach sy'n cyflawni tasgau syml ac y gellir eu haildefnyddio. Mae hyn hefyd yn wir am y modiwlau yn eich pecyn.

Peidiwch ag Ailadrodd Eich Hun, Mae Syml yn Well na Chymhleth,
Mae Darllenadwyedd yn Bwysig.

Strwythur Ffeilio Python

```

"""
docstring modiwl
"""

import llyfrgell_safonol
import llyfrgell_allanol
import modiwlau_y_prosiect

CYSONYN_LEFEL_MODIWL = 0

def pob_ffwythiant(paramedrau):

    """
    docstring ffwythiant
    """

    CYSONYN_LEFEL_FFWYTHIANT = 0

    newidyn_lefel_ffwythiant = 0

class pobDosbarth:

    """
    docstring dosbarth
    """

main()
```

Dyma'r drefn resymegol ac angenrheidiol o elfennau mewn ffeil Python yn unig.

Nid oes rhaid defnyddio pob elfen ym mhob ffeil - mae modiwlau bach y gellir eu haildefnyddio yn well!

Mae **Newidynnau Lefel Modiwl** (Eang) yn bosibl ond yn arfer gwael. Cynnwys Lefel A yw Dosbarthiadau.

Dylai **main()** bob amser alw'r prif ffwythiant sy'n cychwyn y rhaglen.

Mewn geiriau eraill, ni ddylai fod unrhyw god ar waelod y ffeil python - dylid diffinio'r cod sy'n cychwyn y rhaglen yn **main()**!

Strwythur Prosiect Python

```
fy_pecyn/  
├── fy_pecyn/  
│   ├── modiwl1.py  
│   ├── modiwl2.py  
│   └── is_becyn/  
│       ├── is_becyn_1.py  
│       └── is_becyn_2.py  
├── profion/  
│   ├── prawf_modiwl1.py  
│   └── prawf_modiwl2.py  
├── adnoddau/  
│   ├── data_mewnbwn.txt  
│   └── data_allbwn.csv  
├── README.txt  
└── gofynion.txt
```

Cyfeiriadur prif pecyn
Modiwl 1
Modiwl 2
Cyfeiriadur is_becyn

Cyfeiriadur profion

Cyfeiriadur adnoddau

Dogfennaeth
Rhestr dibyniaethau

Dim ond trosolwg sylfaenol yw hwn o'r strwythur ffeiliau y gallech ddod ar eu traws mewn prosiectau Python - o lyfrgelloedd rydych chi wedi'u lawrlwytho i gynrychioliadau ar-lein fel ar GitHub.

Mae llawer mwy o ffeiliau a chyfeiriaduron y gellir eu cynnwys.

Noder nad yw creu pecyn o'r cymhlethdod hwn, na chofio'r strwythur hwn, yn ofyniad ar gyfer cwrs TGAU Cyfrifiadureg.

Dim ond trosolwg a gwybodaeth gefndirol ddefnyddiol yw hwn.

PEP 8 Style Guide:
peps.python.org/pep-0008/



Hangman

technocamps

Cefndir

Bwriad y dasg hon yw bod yn brosiect hwyliog i ddysgwyr ddysgu Python a thyfu mewn hyder wrth ddefnyddio.

Dim ond i'r Shell y bydd y prosiect a ddangosir yma'n allbynnu, ond nid oes unrhyw reswm na ellid ail-greu hyn yn Tkinter (gyda rhyngwyneb graffigol), neu na ellid defnyddio llyfrgell Turtle i luniadu'r graffigwaith! Mae'r cod ar gyfer fersiwn o Tkinter wedi'i gynnwys i gyfeirio ato.

Deunyddiau

Mae'r deunyddiau ar gyfer y dasg hon - sy'n cynnwys y rhaglen wedi'i chwblhau i'r addysgwr gyfeirio ati a rhestr eiriau i'r dysgwyr, i'w cael yn y ddolen ganlynol: tc1.me/educonf2025



Nodiadau

Tasg 1 - Gair ar Hap

Disgrifiad Y Dasg

I ddechrau, rhaid i ddysgwyr greu ffeil Python newydd (h.y. `'hangman.py'`, dyma fydd yr unig fodiwl yn y gweithgaredd hwn - fodd bynnag, tasg bonws da fyddai gwahanu'r cod dros sawl modiwl!).

Y dasg gyntaf yw cynhyrchu gair ar hap, dyna bwynt Hangman wedi'r cyfan. Darperir rhestr eiriau fel nad oes rhaid i ddysgwyr ysgrifennu rhestr gynhwysfawr.

Côd Tasg 1

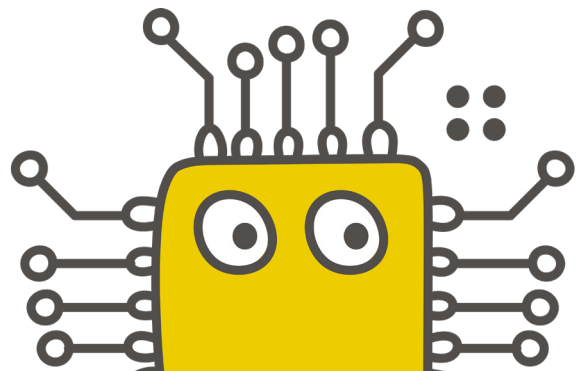
```
import os
from random import randint

def gair_ar_hap():

    ffeil_geiriau = open(os.path.dirname(
                        os.path.abspath(__file__)) +
                        "/wordlist.txt", "r")
    data_geiriau = ffeil_geiriau.read()
    rhestr_geiriau = data_geiriau.split("\n")

    rhif = randint(0, len(rhestr_geiriau))
    gair = rhestr_geiriau[rhif]

    return gair
```



Tasg 2 - Dechrau Gêm

Disgrifiad Y Dasg

Nawr bod gennym ni elfen sylfaenol Hangman, gallwn ni ddechrau meddwl drwy'r rhesymeg o weithredu. Mae angen i ni ddechrau'r gêm, ond er mwyn dangos unrhyw gynrychioliad i'r sgrin (**h.y.** `__`) bydd angen i ni wedi cynhyrchu'r gair. Ond efallai ein bod ni hefyd eisiau gallu ailchwarae'r gêm! Felly byddwn ni'n gwneud dwy ffwythiant, un o'r enw `main()` a'r llall o'r enw `dechrau_gem()`. Ar ddechrau `dechrau_gem()` byddwn ni'n datgan y rhestr wag o ddyfaliadau, y gair ar hap a'r cynrychioliad - dylai'r rhain ddigwydd hefyd pan fyddwn ni'n ailchwarae'r gêm.

Côd Tasg 2

```
def main():  
  
    print("Croeso i Hangman!")  
  
    dechrau_gem()  
  
def dechrau_gem():  
  
    dyfaliadau = []  
  
    gair = gair_ar_hap()  
  
    cynnydd = "_" * len(gair)
```

Tasg 3 - Argraffu

Disgrifiad Y Dasg

Er ein bod ni eisiau dechrau gweithredu rhesymeg y gêm ei hun, mae angen i ni hefyd ddangos pob cam i'r chwaraewr, neu ni fydd yn glir bod unrhyw beth wedi digwydd. Mae angen i ni argraffu cyflwr presennol y gêm ar bob cam. Hefyd, dylem argraffu hyn cyn dechrau'r gêm.

Côd Tasg 3

```
def dechrau_gem():  
    ~ ~ ~  
    argraffu(dyfaliadau, cynnydd)  
  
def argraffu(dyfaliadau, cynnydd):  
    print("\n" + cynnydd)  
  
    str_dyfaliadau = ""  
  
    for i in dyfaliadau:  
        str_dyfaliadau = str_dyfaliadau + i + " "  
  
    print("\nWedi Dyfalu:\n" + str_dyfaliadau)
```

Tasg 4 - Rhesymeg y Gêm

Disgrifiad y Dasg

Nawr daw rhesymeg y gêm. Mae angen i'r defnyddiwr ddyfalu llythyren, yna:

os yw'r llythyren honno yn y gair dylem ddisodli'r " " ar gyfer pob achos o'r llythyren mewn cynnydd gyda'r llythyren ei hun

Byddwn yn gweithredu gofynion mwy cymhleth unwaith y bydd y rhesymeg sylfaenol yn gweithio.

Côd Tasg 4

```
def gwirio(gair, cynnydd, dyfaliad):  
  
    for llythyren in gair:  
  
        if llythyren == dyfaliad:  
  
            indecs = word.index(letter)  
  
            i = indecs * 2  
  
            cynnydd = cynnydd[:i] + cynnydd[indecs]  
                        + cynnydd[i+1:]  
  
    return cynnydd
```

Nodwch fod yn rhaid i ni luosi `indecs` â 2 ar gyfer `cynnydd` oherwydd rydym yn argraffu 2 nod am bob llythyren (" ").

Tasg 5 - Dolen y Gêm

Disgrifiad Y Dasg

Rhaid i ni gael dolen gêm fel bod y defnyddiwr yn gallu dyfalu pob llythyren yn y gair. Rhaid i bob llythyren a nodir gan y defnyddiwr gael ei gwirio, ei hychwanegu at y rhestr o lythrennau a ddyfalwyd, a'i hargraffu, nes bod pob llythyren yn y gair wedi'i darganfod.

Côd Tasg 5

```
def dyfalu(gair, cynnydd, dyfaliadau):  
    while "_" in cynnydd:  
        dyfaliad = input("\nDyfalu Llythyren: ")  
        cynnydd = gwirio(gair, cynnydd, dyfaliad)  
        dyfaliadau.append(dyfaliad)  
        argraffu(dyfaliadau, cynnydd)
```

Gwirio'r Gêm

Gwiriwch y rhaglen i weld a yw'n gweithio ac oes ganddi unrhyw broblemau.

Tasg 6 - Gwallau

Disgrifiad Y Dasg

Drwy brofi eich gêm byddwch wedi sylwi ar ychydig o wallau, gan gynnwys:

- Gall defnyddwyr rhoi'r un llythyren fwy nag unwaith
- Gall defnyddwyr rhoi nodau anllythrennog, fwy nag un nod...
- Mae gan ddefnyddwyr "fywydau" anfeidrol

Byddwn yn datrys pob un o'r rhain yn eu tro.

Tasg 6 - Atal Llythrennau Ailadrodd

```
def gwirio(gair, cynnydd, dyfaliad):  
  
    for llythyren in gair:  
  
        if llythyren == dyfaliad:  
  
            ~ ~ ~  
  
            gair = gair[:indecs] + " " +  
                  gair[indecs+1:]  
  
    return cynnydd
```

Sylwch, drwy ychwanegu'r llinell hon yn `gwirio`, ein bod yn disodli'r holl lythrennau a ddyfalwyd yn gywir yn `gair` gyda `bylchau`, fel na fyddant byth yn paru â'r dyfaliad eto. Nid yw hynny'n atal y llythyren rhag cael ei dyfalu na'i hychwanegu at ddyfaliadau, ond byddwn yn codio hynny gyda'r cam nesaf.

Tasg 6 - Dilysiant

```
def dyfalu(gair, cynnydd, dyfaliadau):  
  
    global bywydau  
  
    while ("_" in cynnydd) and (bywydau > 0):  
  
        yn_dyfalu = True  
        while yn_dyfalu:  
  
            dyfaliad = input("\nDyfalu Llythyren: ")  
  
            if len(dyfaliad) != 1:  
                print("Dyfalu 1 llythyren!")  
  
            elif dyfaliad.isalpha() == False:  
                print("Rhaid dyfalu llythrennau!")  
  
            elif dyfaliad in dyfaliadau:  
                print("Barod wedi dyfalu hyn!")  
  
            else:  
                yn_dyfalu = False  
  
        cynnydd = gwirio(gair, cynnydd, dyfaliad)  
  
        dyfaliadau.append(dyfaliad)  
  
        argraffu(dyfaliadau, cynnydd)
```


Tasg 6 - Bywydau

Sylwch ein bod wedi ychwanegu gwiriad am fywydau, felly mae'r ddolen gêm ond yn rhedeg tra bod fywydau ar ôl - nawr mae rhaid i ni godio'r rhesymeg.

```
def dechrau_gem():  
  
    global bywydau  
    bywydau = 9  
  
    ~ ~ ~  
  
def gwirio(gair, dyfaliad, cynnydd):  
  
    global bywydau  
    darganfod_llythyren = False  
  
    for llythyren in gair:  
        if llythyren == dyfaliad:  
  
            darganfod_llythyren = True  
  
            ~ ~ ~  
  
    if darganfod_llythyren == False:  
        bywydau -= 1  
  
    return cynnydd
```

Tasg 7 - Estyniadau

Disgrifiad Y Tasg

Mae digon o estyniadau posibl ar gyfer y gêm hon, mae rhai wedi'u heithrio'n fwriadol o'r cod hwn gan eu bod yn syml iawn, mae eraill yn fwy cymhleth neu'n adeiladu ar wybodaeth sylfaenol.

Wedi'i Heithrio'n Fwriadol (Symb):

- Ychwanegu negeseuon llwyddiant a methiant ar gyfer pan fydd y defnyddiwr yn dyfalu'r gair / yn rhedeg allan o fywydau!
- Creu dolen i redeg yr holl gêm i alluogi'r defnyddiwr ddewis chwarae eto heb orfod ail-lansio'r rhaglen.

Safonol:

- Ychwanegu modd 2 chwaraewr a/neu fwrdd sgôr fel y gall chwaraewyr gweld eu buddugoliaethau.
- Ychwanegu opsiwn anhawster (dewis geiriau yn seiliedig ar hyd?)

Uwch:

- Defnyddiwch y llyfrgell Turtle (neu lyfrgell raffigol arall) i lunio'r crogwr wrth i'r defnyddiwr golli pob bywyd!
- Defnyddiwch y llyfrgell Tkinter i roi rhyngwyneb defnyddiwr graffigol i'r rhaglen!

Nodiadau

Tkinter a Canvas

```
import os
from random import randint
import tkinter as tk
from tkinter import messagebox

def main():
    global root, cynfas

    # Creu y prif ffenestr
    root = tk.Tk()
    root.title("Hangman")

    # Ymgychwyn gem
    dechrau_gem()
    root.mainloop()

def dechrau_gem():
    global bywydau, dyfaliadau, gair

    # Diffinio bywydau, rhestr gwag o ddyfaliadau,
    # a'r gair i ddyfalu
    bywydau = 6
    dyfaliadau = []
    gair = gair_ar_hap()

    dechrau_gui()
```

Tkinter a Canvas

```
def gair_ar_hap():  
  
    ffeil_geiriau = open(os.path.dirname(  
        os.path.abspath(__file__)) +  
        "/wordlist.txt", "r")  
    data_geiriau = ffeil_geiriau.read()  
    rhestr_geiriau = data_geiriau.split("\n")  
  
    rhif = randint(0, len(rhestr_geiriau))  
    gair = rhestr_geiriau[rhif]  
    return gair  
  
def dechrau_gui():  
    global root, bywydau, dyfaliadau, gair,  
        cynfas, label_gair, label_bywydau,  
        label_dyfaliadau, maes_dyfaliad  
  
    # Creu'r cynfas am lunio'r hangman  
    cynfas = tk.Canvas(root, width=300, height=300)  
    cynfas.grid(column=0, row=0)  
  
    # Creu botymau Ail-Geisio and Allan  
    botwm_ail = tk.Button(root,  
        text="Ail-Geisio",  
        command=lambda: ail_geisio())  
    botwm_ail.grid(column=0, row=5)  
  
    botwm_allan = tk.Button(root, text="Allan",  
        command=lambda: adael_gem())  
    botwm_allan.grid(column=0, row=6)
```

Tkinter a Canvas

```
# Alluogi'r defnyddiwr i ddefnyddio'r allwedd
# dychwelyd i ddyfalu
root.bind("<Return>", lambda event:gwirio())

# Lunio'r crocbren
cynfas.create_line(20, 280, 120, 280)
cynfas.create_line(70, 280, 70, 20)
cynfas.create_line(70, 20, 170, 20)
cynfas.create_line(170, 20, 170, 50)

# Creu label i ddangos y gair
label_gair = tk.Label(root,
    text=" ".join(["_" for llythyren in gair]))
label_gair.grid(column=0, row=1)

# Creu label i ddangos bywydau ar ôl
label_bywydau = tk.Label(root,
    text="Bywydau: {}".format(bywydau))
label_bywydau.grid(column=0, row=2)

# Creu label i ddangos y dyfaliadau
label_dyfaliadau = tk.Label(root,
    text="Wedi dyfalu: ")
label_dyfaliadau.grid(column=0, row=3)

# Creu maes i'r defnyddiwr dyfalu llythyren
maes_dyfaliad = tk.Entry(root)
maes_dyfaliad.grid(column=0, row=4)
```

Tkinter a Canvas

```
def gwirio():
    global root, bywydau, dyfaliadau, gair,
        label_gair, label_bywydau,
        label_dyfaliadau, maes_dyfaliad

    dyfaliad = maes_dyfaliad.get().lower()
    maes_dyfaliad.delete(0, tk.END)

    # Gwirio a yw'r dyfaliad yn un llythyren,
    # neu a yw'r dyfaliad eisoed wedi'i dyfalu
    if (len(dyfaliad) != 1) or (not dyfaliad.isalpha())
    or (dyfaliad in dyfaliadau):
        return

    # Ychwanegu dyfaliad i dyfaliadau
    dyfaliadau.append(dyfaliad)
    label_dyfaliadau.config(
        text="Wedi dyfalu: {}".format(
            " ".join(guessed)))

    # Gwirio fod y dyfaliad yn y gair
    if dyfaliad in gair:
        rhestr_llyth = list(label_gair["text"])

        for i in range(len(gair)):
            if gair[i] == dyfaliad:
                rhestr_llyth[2*i] = dyfaliad

    label_gair.config(text="".join(rhestr_llyth))
```

Tkinter a Canvas

```
# Gwiriwch a yw'r defnyddiwr wedi ennill
if "_" not in rhestr_llyth:
    messagebox.showinfo("Hangman", "Ennill!")
    maes_dyfaliad.config(state=tk.DISABLED)
    return

#Os nad yw dyfaliad yn y gair, lleihau bywydau
else:
    bywydau -= 1
    label_bywydau.config(
        text="Bywydau: {}".format(bywydau))

# Llunio'r hangman
if bywydau == 5:
    cynfas.create_oval(140, 50, 200, 110)
elif bywydau == 4:
    cynfas.create_line(170, 110, 170, 170)
elif bywydau == 3:
    cynfas.create_line(170, 130, 140, 140)
elif bywydau == 2:
    cynfas.create_line(170, 130, 200, 140)
elif bywydau == 1:
    cynfas.create_line(170, 170, 140, 190)
elif bywydau == 0:
    cynfas.create_line(170, 170, 200, 190)
    messagebox.showinfo("Hangman",
        "Methu! Roedd '{}' y gair!".format(gair))
    maes_dyfaliad.config(state=tk.DISABLED)
```

Tkinter a Canvas

```
def ail_geisio():
    global cynfas, label_dyfaliadau, label_gair,
        label_bywydau

    # Dileu'r cynfas a chlirio'r labelai
    cynfas.destroy()
    label_dyfaliadau.config(text="")
    label_gair.config(text="")
    label_bywydau.config(text="")

    #Ail-ddechrau Game
    dechrau_gem()

def adael_gem():
    global root

    #Dileu'r root (h.y. y GUI llawn)
    root.destroy()

main()
```

Sylwch fod y rhaglen hon yn defnyddio nifer gormodol o newidynnau eang, mae hyn yn arfer gwael a dylech osgoi! Fodd bynnag, mae'n gwneud Tkinter yn llawer symlach (ac yn symlach i'w ddilyn). Yn ddelfrydol, dylid pasio pob newidyn yn y rhaglen hon fel paramedrau.

Cefndir

Yn flaenorol bwriadwyd y dasg hon i ddangos holl ofynion gwaith cwrs TGAU **Cyfrifiadureg**, ond nid yw hyn yn bodoli rhagor.

Mae'r dasg yn dal yn werth chweil wrth ddeall sut i adeiladu rhaglen Python fawr, ac mae'n cwmpasu llawer o gysyniadau craidd. Yn ogystal, mae'r **Deunyddiau Enghreifftiol** ar gyfer **TGAU Cyfrifiadureg** newydd yn defnyddio cod tebyg i god y gwaith cwrs blaenorol, ac mae'n rhesymol i dybio y byddant yn parhau i ddefnyddio'r math hwn o system ar gyfer asesiadau yn y dyfodol.

Deunyddiau

Mae'r deunyddiau ar gyfer y dasg hon - sy'n cynnwys y rhaglen wedi'i chwblhau i'r addysgwr gyfeirio yn ogystal â chyflwyniad i gynorthwyo gyda chyflwyno cynnwys - ar gael yn y ddolen ganlynol: tc1.me/educonf2025



Nodiadau

Tasg 1 - Mewngofnodi

Disgrifiad Y Dasg

I ddechrau, rhaid i ddysgwyr greu ffeil Python newydd (h.y. **'pyshop.py'**, dyma fydd yr unig fodiwl yn y gweithgaredd hwn - fodd bynnag, tasg bonws da yw rhannu'r ffeil dros sawl modiwl!).

Y dasg gyntaf yw creu system fewngofnodi syml gan ddefnyddio'r derfynell (ni fyddwn yn defnyddio Tkinter yn y dasg hon). Rhaid i'r system fewngofnodi storio'r enw defnyddiwr a'r cyfrinair (fel newidynnau am y tro), derbyn mewnbwn gan y defnyddiwr ar gyfer yr enw defnyddiwr a'r cyfrinair a gwirio a ydynt yn cyfateb, ac yn olaf argraffu neges llwyddiant neu fethiant mewngofnodi ar gyfer y defnyddiwr.

Côd Tasg 1

```
enw_stor = "gweinyddwr"
cyfrinair_stor = "12345"

enw_mewn = input("Rhowch enw defnyddiwr: ")
cyfrinair_mewn = input("Rhowch cyfrinair: ")

if (enw_stor == enw_mewn) and
(cyfrinair_stor == cyfrinair_mewn):
    print("Croeso!")
else:
    print("Mewngofnodiad wedi Methu!")
```

Tasg 2 - Dewislen

Disgrifiad Y Dasg

Y cam nesaf yw codio prif **ddewislen** i'r defnyddiwr ar ôl iddynt fewngofnodi. Bydd hyn yn caniatáu i ddefnyddwyr sydd wedi mewngofnodi'n llwyddiannus ddefnyddio system y siop - felly dylai gynnwys ychydig o nodweddion.

Am y tro ni fyddwn yn ychwanegu'r nodweddion hyn, byddwn yn **stympio** - dyma pryd rydych chi'n gadael lle i nodwedd arfaethedig yn eich cod, gan ganiatáu i'r cod hwnnw dal cael ei redeg (gall naill ai beidio â gwneud unrhyw beth neu hysbysu ei fod wedi rhedeg). Dyma'r hyn a wnaethom o'r blaen gyda'r datganiadau argraffu yn dileu'r mewngofnodi, mae'n offeryn hynod ddefnyddiol i raglenwyr!

Nodiadau

Côd Tasg 2

~ ~ ~

```
if (enw_stor == enw_mewn) and
(cyfrinair_stor == cyfrinair_mewn):

    print("Croeso!")

    dewis = input("""
Dewis opsiwn:
1. Gweld
2. Ychwanegu
3. Dileu
""")

    try:
        dewisiad = int(dewis)
        if dewisiad == 1:
            print("Dewislen -> Gweld")
        elif dewisiad == 2:
            print("Dewislen -> Ychwanegu")
        elif dewisiad == 3:
            print("Dewislen -> Dileu")
        else:
            print("Dewisiad annilys!")

    except ValueError:
        print("Rhaid derbyn cyfanrif dilys!")

else:
    print("Mewngofnodiad wedi Methu!")
```

Task 3 - Ffwythiannau

Disgrifiad Y Dasg

Nawr bod y ddewislen yn gweithio, bydd angen i ni ddechrau disodli'r stympiâu gyda gweithrediadau. Ond bydd hyn yn arwain at un rhaglen fawr iawn, anodd ei dilyn, ac anhrefnus!

I osgoi hyn, gallwn ddechrau gwahanu ein cod mewn i sawl ffwythiant. Yn ddelfrydol, dim ond un pwrpas fydd gan bob ffwythiant a bydd wedi'i henwi'n briodol; bydd hyn yn helpu i gadw ein cod yn ddarllenadwy ac yn hawdd ei aildddefnyddio (os oes angen).

Gadewch i ni ddechrau trwy rannu'r mewngofnodi a'r ddewislen mewn i ddau ffwythiant! Hefyd, er mwyn cadw at gonfensiynau Python, byddwn yn dechrau'r rhaglen trwy alw swyddogaeth o'r enw `main()`.

Nodyn: byddem hefyd yn ddelfrydol yn gwahanu ein ffwythiannau ar draws sawl modiwl hunangynhwysol, er mwyn osgoi cael un sgript Python fawr iawn!

Nodiadau

Tasg 3 - Ffwythiannau (Dewislen)

Ar gyfer y ffwythiant `dewislen()`, byddwn yn cymryd cod y ddewislen o fewn `mewngofnodiad()`, ac yn ei osod o fewn diffiniad ei hun.

```
def dewislen():  
  
    dewis = input("""  
    Dewis opsiwn:  
    1. Gweld  
    2. Ychwanegu  
    3. Dileu  
    """)  
  
    try:  
        dewisiad = int(dewis)  
        if dewisiad == 1:  
            print("Dewislen -> Gweld")  
        elif dewisiad == 2:  
            print("Dewislen -> Ychwanegu")  
        elif dewisiad == 3:  
            print("Dewislen -> Dileu")  
        else:  
            print("Dewisiad annilys!")  
  
    except ValueError:  
        print("Rhaid derbyn cyfanrif dilys!")
```

Tasg 3 - Ffwythiannau (Mewngofnodi)

Ar gyfer y ffwythiant mewngofnodi, byddwn yn gosod y ffwythiant o fewn diffiniad, gan gael gwared ar y ddewislen a ychwanegwyd yn flaenorol. Ni fyddwn yn galw'r ffwythiant dewislen eto, ond yn ychwanegu dau ddatganiad i ddychwelyd, y byddwn yn gweld eu pwrpas pan fyddwn yn gweithredu'r ffwythiant `main()`.

```
def mewngofnodi():  
  
    enw_stor = "gweinyddwr"  
    cyfrinair_stor = "12345"  
  
    enw_mewn = input("Rhowch enw defnyddiwr: ")  
    cyfrinair_mewn = input("Rhowch cyfrinair: ")  
  
    if (enw_stor == enw_mewn) and  
        (cyfrinair_stor == cyfrinair_mewn):  
        print("Croeso!")  
        return True  
  
    else:  
        print("Mewngofnodiad wedi Methu!")  
        return False
```

Tasg 3 - Ffwythiannau (Main)

Byddwn hefyd yn ychwanegu ffwythiant main ar frig ein sgript (mae hyn yn gonfensiwn), ac yna rhaid i ni ei galw o waelod ein sgript (h.y. mae'r sgript wedi'i gwasgu rhyngddynt). Nawr pan fydd Python yn cyrraedd y gwaelod mae'n galw'r ffwythiant `main()` sydd yn ei thro yn galw `mewngofnodi()` a `dewislenn()`.

```
def main():  
    mewngofnodi()  
    dewislenn()
```

```
~ ~ ~
```

```
main()
```

Nodiadau

Tasg 4 - Dolenni

Disgrifiad Y Dasg

Nawr mae'r cod yn edrych yn llawer taclusach, a bydd yn llawer haws ei addasu ac ychwanegu ato. Fodd bynnag, cyn ychwanegu nodweddion mae problem newydd - does dim ots os mae'r defnyddiwr yn nodi'r enw defnyddiwr neu'r cyfrinair yn gywir, maent yn symud ymlaen i'r ddewislen beth bynnag. Ac ar ôl dewis eitem yn y ddewislen, mae'r rhaglen yn dod i ben. Gellir datrys y ddwy broblem hyn gyda dolenni!

Tasg 4 - Dolennau (Main)

Bydd y ddolen `while` gyntaf yn cadw'r defnyddiwr yn sownd wrth fewngofnodi nes bod mewngofnodiad llwyddiannus, tra bod yr ail ddolen `while` yn cadw'r defnyddiwr yn sownd yn y ddewislen.

```
def main():  
  
    wedi_mewngofnodi = False  
  
    while not wedi_mewngofnodi:  
  
        wedi_mewngofnodi = mewngofnodi()  
  
    mewn_dewislen = True  
  
    while mewn_dewislen:  
        dewislen()
```

Tasg 5 - Eitemau

I weithredu nodweddion yn y ddewislen hon, rhaid adeiladu strwythur data i ddal eitemau i'w gwerthu. Mae sawl ffordd o wneud hyn, ond mae'r enghraifft hon yn defnyddio geiriadur i storio manylion pob eitem, gyda phob eitem wedi'i nythu o fewn rhestr. Datganir hyn yn **eang** i allu ei gyrchu o unrhyw le.

Rhaid llwytho'r data hwn i mewn o **ffeil allanol** (a'i ddiweddarau pryd bynnag y gwneir unrhyw newidiadau). Ni ellir storio yn y cod gan fydd hyn yn gwneud unrhyw ddiweddariadau'n dros dro (h.y. ar ôl cau'r rhaglen, bydd ail-lansio yn llwytho'r hen restr sydd wedi'i godio).

Byddai'r strwythur yn edrych fel hyn (pe bai wedi'i godio):

```
eitemau = [ {
                "Enw": "Coca-Cola",
                "Pris": 1.50,
                "NiferMewnStoc": 50
            }, {
                "Enw": "Irn-Bru",
                "Pris": 1.20,
                "NiferMewnStoc": 30
            },
            ...
        ]
```

Tasg 5 - Eitemau (Llwytho)

```
def llwytho_eitemau():  
  
    global eitemau  
  
    ffeil_stoc = open('stock.csv', 'r')  
  
    eitemau = []  
  
    for llinell in ffeil_stoc:  
        eitem = []  
  
        for colofn in llinell.strip().split(','):   
            eitem.append(colofn)  
  
            eitemau.append( {  
                "Enw":      eitem[0],  
                "Pris":     int(eitem[1]),  
                "MewnStoc": int(eitem[2])  
            })  
  
    ffeil_stoc.close()
```

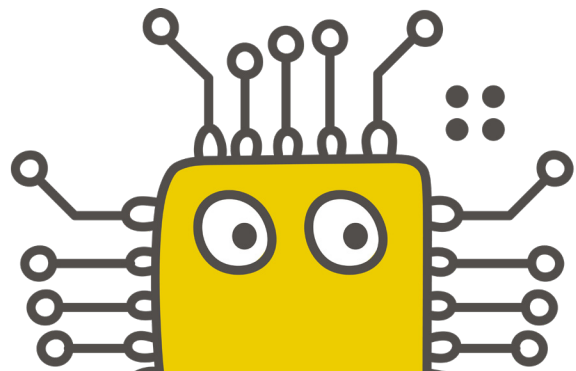
Tasg 5 - Eitemau (Cadw)

```
def cadw_eitemau():  
  
    global eitemau  
  
    ffeil_stock = open('stock.csv', 'w')  
  
    for eitem in eitemau:  
  
        llinell = eitem["Enw"] + "," +  
                  str(eitem["Pris"]) + "," +  
                  str(eitem["MewnStoc"]) + "\n"  
  
        ffeil_stock.write(llinell)  
  
    ffeil_stock.close()
```

Tasg 5 - Eitemau (csv)

Bydd yn rhaid i'r dysgwyr barhau gyda'r esiampl neu greu ffeil **csv** newydd, a'i llenwi ag eitemau yn y fformat canlynol:

```
Coca-Cola,1.5,50  
Irn-Bru,1.2,30  
...
```



Tasg 6 - Gweld

Disgrifiad Y Dasg

Nawr gallwn ddechrau gweithredu ffwythiannau yn y ddewislen. Byddwn yn dechrau gyda ffwythiant syml i weld yr holl eitemau.

Cyn y bydd hyn yn gweithio, bydd hefyd rhaid i ni alw'r ffwythiant `llwytho_eitemau()` yn `main()`.

Tasg 6 - Gweld

```
def gweld_eitemau():  
  
    global eitemau  
  
    for eitem in eitemau:  
  
        print("Enw: " + eitem["Enw"])  
        print("Pris: " + str(eitem["Pris"]))  
        print("Stoc: " + str(eitem["MewnStoc"]) + "\n")
```

Tasg 6 - Gweld (Dewislen)

```
def dewislen():  
  
    ~ ~ ~  
  
    try:  
        dewisiad = int(dewis)  
        if dewisiad == 1:  
  
            gweld_eitemau()  
  
    ~ ~ ~
```

Task 6 - Gweld (Main)

```
def main():  
  
    wedi_mewngofnodi = False  
    while not wedi_mewngofnodi:  
        wedi_mewngofnodi = mewngofnodi()  
  
    llwytho_eitemau()  
  
    mewn_dewislen = True  
    while mewn_dewislen:  
        dewislen()
```

Tasg 7 - Ychwanegu

Disgrifiad Y Dasg

Nawr byddwn yn gweithredu'r ffwythiant ychwanegu i ganiatáu i eitemau newydd gael eu hychwanegu i'r stoc. Bydd rhaid i ni ysgrifennu'r eitemau i ffeil ar ôl gwneud newidiadau. Mae hefyd yn hanfodol galw'r swyddogaeth hon o'r ddewislen!

Nodyn: dylai hyn hefyd gynnwys trin gwallau rhag ofn bod mewnbwn y defnyddiwr yn annilys, mae hyn wedi'i adael allan yn fwriadol.

Tasg 7 - Ychwanegu

```
def ychwanegu_eitem():  
  
    global eitemau  
  
    eitem_newydd = dict()  
  
    eitem_newydd["Enw"] = input("Enw'r eitem: ")  
    eitem_newydd["Pris"] =  
        float(input("Pris yr eitem: "))  
    eitem_newydd["MewnStoc"] =  
        int(input("Nifer mewn stoc: "))  
  
    eitemau.append(eitem_newydd)  
  
    cadw_eitemau()
```

Tasg 8 - Dileu

Disgrifiad Y Dasg

Nawr byddwn yn gweithredu'r ffwythiant dileu i ganiatáu i eitemau gael eu tynnu o stoc. Bydd rhaid i ni ailysgrifennu'r ffeil eitemau ar ôl gwneud newidiadau. Mae hefyd yn hanfodol galw'r ffwythiant hon o'r ddewislen!

Tasg 8 - Dileu

```
def dileu_eitem():  
  
    global eitemau  
    enw_eitem = input("Enw'r eitem i ddileu: ")  
  
    for eitem in eitemau:  
        if eitem["Enw"] == enw_eitem:  
  
            print("Enw: " + eitem["Enw"])  
            print("Pris: " + str(eitem["Pris"]))  
            print("Stoc: " + str(eitem["MewnStock"])) +  
                "\n")  
  
    cadarnhau = input("Dileu'r eitem? Y/N: ")  
    if cadarnhau.upper() == "Y":  
        eitemau.remove(eitem)  
        cadw_eitemau()
```


Tasg 9 - Estyniadau

Disgrifiad Y Dasg

Mae yna nifer o estyniadau y gellir eu gwneud i'r rhaglen hon, a byddem yn annog dysgwyr i roi cynnig ar estyniadau eu hunain i annog dysgu hunangyfarwyddyd. Ymarfer, treial a chamgymeriad yw'r ffordd orau o ddysgu iaith raglennu.

Mae estyniadau posibl yn cynnwys:

- Ffwythiant ddiweddarau i newid enw, pris neu faint eitem.
- Dull i brynu eitemau o'r siop - yn ddelfrydol dylai hyn leihau'r maint a chadw cyfanswm incwm rhedegog.
- Dulliau lluosog o chwilio am yr eitemau (i'w defnyddio wrth ddileu a diweddarau)

Mae estyniadau mwy cymhleth yn cynnwys:

- Argraffu/arbed log trafodion wrth allgofnodi, y mae'n rhaid iddo gadw golwg ar yr holl werthiannau a wneir yn ystod y sesiwn. Hyd yn oed yn well os gellir prynu sawl eitem wahanol fesul trafodiad!
- Tybio bod sawl eitem wahanol fesul trafodiad, argraffwch dderbynnwch cwsmer
- Gweithredu'r system gyfan gan ddefnyddio Tkinter.

Nodiadau

Cystrawen Python

Gweithred

Cod Python

Argraffu	<pre>print("Bydd hyn yn cael ei argraffu")</pre>
Gosod newidyn	<pre>fy_newyddin = 42 fy_newyddin_arall = "Helo"</pre>
os/arall	<pre>if oed > 17: print("Rydych chi'n oedolyn") else: print("Nid ydych chi'n oedolyn")</pre>
Storio mewnbwn	<pre>enw = input("Beth yw dy enw di?") oed = int(input("Beth yw dy oedran?"))</pre>
Taflu teip	<pre>mewnbwn = input("oedran:") oed = int(mewnbwn) print(oed)</pre>
Ymdrin â gwallau	<pre>mewnbwn = input("oedran:") try: oed = int(mewnbwn) print(oed) except ValueError: print("ni wnest ti roi rhif cyfan")</pre>
Ffwythiannau	<pre>def argraffu_helo(enw): print("Helo ", enw) argraffu_helo("Casey")</pre>

Cystrawen Python

Gweithred

Cod Python

Rhestrau

```
rhestr_wag = [] / rhestr_wag = list()
rhestr_testun = ["a", "b", "c"]
rhestr_rhifau = [1, 2, 3]
rhestr_gymysg = ["a", 1, "b", 2]

print(rhestr_testun[0])
print(len(rhestr_testun))
rhestr_testun.append("ch")
rhestr_testun[0] = "y"
rhestr_testun.remove("k")
```

Rhestrau 2D

```
rhestr_wag = [[], []]
fy_rhestr = [ [0,1,2,3,4] , [3,6,1] ]
print(fy_rhestr[0][1])
```

Geiriaduron

```
geir_wag = {} / geir_wag = dict()
ceir = {
    "Audi Q5" : 20000,
    "Volkswagen Polo" : 18000
}

print(list(ceir.keys()))
print(ceir)
print(ceir["Audi Q5"])
print(len(ceir))
ceir["Ford KA"] = 500
ceir["Audi Q5"] = 10000
ceir.pop("Audi Q5")
```

Cystrawen Python

Gweithred

Cod Python

Dolen "while"

```
cyfrinair = ""
while cyfrinair != "cyfrinach":
    print("Beth yw'r cyfrinair?")
    cyfrinair = input()
print("Da iawn!")
```

Dolen "for" (rhestr)

```
ffrwythau = ["banana", "afal"]
for ff in ffrwythau:
    print(ff)
```

Dolen "for" (cyfrif)

```
for i in range(10):
    print("i = ", i)
    print("i*i = ", i*i)
```

Darllen ffeil

```
ffeil = open("Prawf_1.txt", "r")
for llinell in ffeil:
    print(llinell)
ffeil.close()
```

Ysgrifennu i ffeil

```
ffeil = open("Prawf_2.txt", "w")
ffeil.write("Ysgrifennu.")
ffeil.close()
```

Atodi i ffeil

```
ffeil = open("Prawf_3.txt", "a")
ffeil.write("Atodi.")
ffeil.close()
```

Creu ffeil

```
ffeil = open("Prawf_4.txt", "x")
```

Trwsio Tkinter

technocamps

Cefndir

Bwriadwyd y dasg hon fel ymarfer ar gyfer cwricwlwm **TGAU Cyfrifiadureg**.

Mae hwn yn fersiwn o'r gweithgaredd PyShop a ysgrifennwyd gan ddefnyddio Tkinter. Nod y dasg yw deall y rhaglen a thrwsio'r holl wallau sy'n bresennol, wrth wella gallu'r dysgwyr i ddarparu sylwadau a dogfennaeth dda.

Yn y rhaglen hon rydym wedi ceisio cynnwys pob math o dechneg raglennu sydd wedi ymddangos yn y **Deunyddiau Enghreifftiol**. Er nad yw'r holl dechnegau hyn yn cynnwys gwallau i drwsio, y gobaith yw y bydd gorfod deall swyddogaeth arfaethedig y cod yn cynorthwyo dysgwyr i ddeall amrywiaeth yr arddulliau rhaglennu Python.

Deunyddiau

Mae'r deunyddiau ar gyfer y dasg hon - sy'n cynnwys y rhaglen wedi'i chwblhau i'r addysgwr gyfeirio ati a rhaglen sydd wedi torri i ddysgwyr ei thrwsio, yn ogystal â chyflwyniad i gynorthwyo gyda chyflwyno cynnwys - ar gael yn y ddolen ganlynol:
tc1.me/educonf2025



Tasg 1 - Trwsio Mewngofnodi

Disgrifiad Y Dasg

Yn y dasg gyntaf, rhaid i ddysgwyr edrych ar y ffeil **'gui_mewngofnodi.py'** a thrwsio'r **6 gwall** sydd ynddi. Bwriedir i hyn fod yn dasg gyflwyniadol, er mwyn osgoi gorlethu trwy roi 4 modiwl o raglen fawr iddynt.

Fodd bynnag, i redeg a dechrau profi'r ffeil hon, rhaid i ddysgwyr ysgrifennu **prif fodiwl** i'w galw (megis **'gui_main.py'**) Mae hon yn wybodaeth hanfodol am Python, gan fod y rhan fwyaf o raglenni'n cael eu **datgysylltu** mewn i fodiwlau; lle mae prif fodiwl yn galw'r modiwlau arall ac yn rhedeg y rhaglen!

Prif Fodiwl

Creu ffeil Python newydd o'r enw **'gui_shop_main.py'**, wedi'i chadw yn yr un cyfeiriadur â'r modiwlau eraill, a chynnwys y cod canlynol:

```
from gui_mewngofnodi import creu_mewngofnodi

creu_mewngofnodi ()
```

Gwall Mewngofnodi #1

Gadawyd y newidynnau enw defnyddiwr a chyfrinair a storiwyd yn wag, dylai'r ddau gynnwys gwerthoedd i sicrhau mai dim ond defnyddiwr dilys all gael mynediad at y rhaglen:

```
enw_stor = "Fy.Enw"
cyfrinair_stor = "cyfr1n41rDa"
```

Gwall Mewngofnodi #2

Sicrhaodd y gwiriad mewngofnodi fod **naill ai'r** enw defnyddiwr **NEU'R** cyfrinair yn cyfateb i'r newidynnau a storiwyd. Dylai system fewngofnodi sicrhau bod y **ddau** yn cyfateb:

```
if ((enw_mewn.get() == enw_stor) and  
(cyfrinair_mewn.get() == cyfrinair_stor)):
```

Gwall Mewngofnodi #3

Roedd teitl y ffenestr yn ddiystyr, dylai rhaglen sydd wedi'i chynllunio'n dda gael enwau perthnasol ar gyfer pob ffenestr:

```
root_mewngofnodi.title("Rhaglen Siop")
```

Gwall Mewngofnodi #4

Rhaid gosod unrhyw elfennau Tkinter o fewn Ffrâm gan ddefnyddio **rheolwr geometreg**, yn yr achos hwn, rydym yn defnyddio **pack**.

```
label_teitl = Label(ffram_mewngofnodi,  
                    text = "Croeso i'r Siop!")  
label_teitl.pack()
```

Gwall Mewngofnodi #5

Yn y rhyngwyneb defnyddiwr graffigol (GUI) roedd y maes cyfrinair yn ymddangos i'r chwith o'r label cyfrinair. Mae hyn yn ymddangos fel camgymeriad a dylid ei gywiro fel bod y label ar y chwith.

```
label_cyfrinair = Label(ffram_cyfrinair,
                        text = "cyfrinair")
label_cyfrinair.pack(side = LEFT)
maes_cyfrinair = Entry(ffram_cyfrinair)
maes_cyfrinair.pack(side = RIGHT)
```

Gwall Mewngofnodi #6

Pan gaiff y **Botwm Cyflwyno** ei wasgu, dylai redeg y ffwythiant `prosesu_mewngofnodi` (sy'n golygu y gellir dileu'r `ffwythiant_dibwrpas`).

```
botwm_cyflwyno = Button(ffram_mewngofnodi,
                        text = "cyflwyno",
                        bg = "White", fg = "Red",
                        command = prosesu_mewngofnodi)
```

Sylwch fod y ffwythiant hwn wedi'i darparu heb gromfachau (h.y. nid yw'n galw'r ffwythiant), sy'n golygu mai dim ond **cyfeiriad** at y ffwythiant ydyw (h.y. newidyn sy'n cynnwys y galwad i'r ffwythiant). Dyma pam mae `prosesu_mewngofnodi` wedi'i **nythu** o fewn `creu_mewngofnodi`, fel ei bod o fewn cwrmpas gwneud cyfeiriad.

Rhowch gynnig ar ddodi `command = print("Dim pwrpas.")` yn lle `command = ffwythiant_dibwrpas`, a gweld beth sy'n digwydd (caiff ei alw ar unwaith ar ôl rhedeg y rhaglen ac nid yw'r botwm yn gwneud dim mwyach)!

Tasg 2 - Ychwanegu Sylwadau

Disgrifiad Y Dasg

Yr ail dasg yw sicrhau bod y cod wedi'i sylwadu'n dda. Rhaid i'r dysgwyr ychwanegu sylwadau lle bo angen yn y modiwl **'gui_mewngofnodi'** - er y dylai'r dasg hon barhau drwy bob modiwl.

Mae ychwanegu sylwadau yn sgil rhaglennu hanfodol, ac mae wedi'i chynnwys fel cwestiynau enghreifftiol yn y cwricwlwm newydd.

Arfer gorau ar gyfer sylwadau:

- **Peidiwch â rhoi sylwadau ar bob llinell o god** - yn lle hynny, rhannwch adrannau o god a rhowch sylwadau ar y darnau sy'n gwneud rhywbeth penodol.
- **Ni ddylai sylwadau esbonio beth mae'r cod yn ei wneud** - dyna bwrpas y cod, dim ond esbonio'r pwrpas y dylent ei wneud. Gall hwn fod yr un anodd i ddeall, felly gallai'r enghraifft hon helpu:

► **Sylwadau Gwael**

```
# Argraffu 'Wedi Mewngofnodi!' wedyn ar y llinell nesaf
# 'Helo ' gydag enw'r defnyddwyr wedi atodi
print("Wedi Mewngofnodi!\nHelo ", enw)
```

► **Sylwadau Da**

```
# Neges croeso i'r defnyddwyr
print("Wedi Mewngofnodi!\nHelo ", enw)
```

- **Dylid rhoi sylwadau ar ffwythiannau gyda `"""dogfennau"""`** - dylai'r rhain esbonio pwrpas y ffwythiannau ac amlinellu unrhyw bamedrau (wedi'u hysgrifennu fel **Args:**) a/neu ddatganiadau **Return:** sydd yn bresennol yn y ffwythiant.

Tasg 3 - Atgyweirio'r Prif Fodiwl

Disgrifiad Y Dasg

Mae angen ffordd arnom i **newid** rhwng y ffenestr mewngofnodi a ffenestr y siop drwy'r botymau mewngofnodi/allgofnodi.

Mae hyn yn eithaf anodd i wneud pan rydyn ni'n defnyddio Python yn **drefniadol**, gan y gallwn ni ddod ar draws gwallau drwy newid ffenestri 'diddiwedd' (mae hwn yn **RecursionError**, gan mai dim ond cymaint o haenau o ddyfnder y caniateir i ffwythiannau dychweliadol fynd - **bwriedir** defnyddio Tkinter mewn strwythur Gwrthrych-Gyfeiriadol sy'n cael gwared ar lawer o'r problemau rydyn ni'n dod ar eu traws!).

Felly, bydd angen **datrysiad creadigol** arnom i osgoi'r gwall! Mae angen i ni **ddinistrio'r** ffenestr mewngofnodi **cyn newid** i'r siop (fel nad oes unrhyw ddychweliad yn digwydd) - ond os gwnawn hyn o fewn y ffenestr mewngofnodi, ni fydd y siop byth yn cael ei galw (gan fod y ffenestr wedi'i dinistrio cyn galw'r mewngofnodi)!

I osgoi hyn, bydd yn rhaid i ni weithredu popeth o fewn '**gui_main**', trwy ddiffinio ffwythiannau sy'n dinistrio'r ffenestr gyfredol ac yn galw'r ffenestr newydd. Bydd angen **pasio** cyfeiriadau at y ffwythiannau hyn i bob ffenestr, fel y gwelsom yn flaenorol yn Wall Mewngofnodi #6. Yn ogystal, bydd yn rhaid i ni addasu '**gui_mewngofnodi**' i dderbyn a defnyddio'r cyfeirnod hwn.

Nodiadau

Atgyweirio'r Prif Fodiwl

```
from gui_mewngofnodi import creu_mewngofnodi
from gui_siop import creu_siop

def newid_i_siop(root_mewngofnodi):
    root_mewngofnodi.destroy()
    creu_siop(newid_i_mewngofnodi)

def newid_i_mewngofnodi(root_siop):
    root_siop.destroy()
    creu_mewngofnodi(newid_i_siop)

creu_mewngofnodi(newid_i_siop)
```

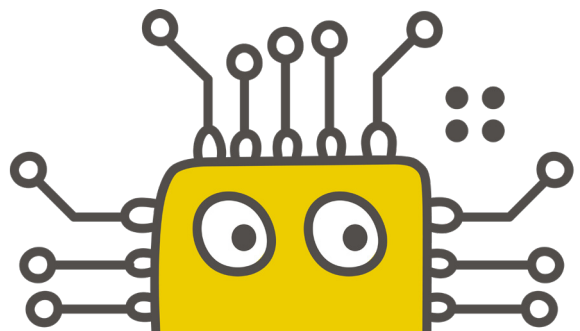
Diweddaru'r Modiwl Mewngofnodi

```
def creu_mewngofnodi(newid_i_siop):

    ~ ~ ~

if (enw_mewn.get() == enw_stor) and
(cyfrinair_mewn.get() == cyfrinair_stor):

    newid_i_siop(root_mewngofnodi)
```



Tasg 4 - Trwsio'r Siop

Disgrifiad Y Dasg

Nawr bod y ffenestr mewngofnodi yn gweithio fel y bwriadwyd, gall dysgwyr redeg y ffeil 'gui_main.py' i brofi'r rhaglen (rhaid iddynt gael yr holl ffeiliau, gan gynnwys 'gui_siop.py', 'gui_ffeiliau.py' ac 'eitemau.csv' yn yr un cyfeiriadur).

Dylai hyn nawr ganiatáu i'r defnyddiwr fewngofnodi a chael mynediad i'r siop, er y bydd y ffenestr hon yn ymddangos yn doredig iawn. Bydd yn rhaid i ni nawr drwsio'r problemau sy'n bresennol yn 'gui_siop.py' a 'gui_ffeiliau.py'.

Gellir darparu'r dasg hon fel un her fawr, ond efallai y bydd yn fuddiol i'r dysgwyr datrys y gwallau sy'n bresennol un ffwythiant ar y tro. Mae hwn yn fodiwl Python eithaf mawr, felly gall deimlo'n llethol heb arweiniad, gan fod cyfanswm o **13 o wallau** i'w canfod a'u trwsio ar draws 10 ffwythiant.

Nodiadau

creu_siop - Gwall Siop #1, 2

Mae dau wall yn bodoli o fewn y ffwythiant creu_siop ('gui_siop.py').

Mae'r cyntaf yn ei gwneud yn ofynnol i'r newidynnau stoc a gwerthiannau gael eu datganu'n **eang**. Mae'r newidynnau hyn yn cael eu harddangos o fewn y GUI, felly mae eu gwneud yn eang ac yn hygyrch trwy bob ffwythiant yn hanfodol er mwyn iddynt gael eu diweddarw o fewn y GUI.

Yr ail wall yw nad yw'r newidyn stoc byth yn cael ei osod. Rhaid gosod hyn trwy alw'r ffwythiant llwytho_stoc o 'gui_ffeiliau.py'.

```
def creu_siop(newid_i_mewngofnodi):

    # Datganu stoc a gwerthiannau'n eang...
    global stoc
    global gwerthiannau

    # Datganu ffram_siop yn eang...
    global ffram_siop

    # Ymgychwyn y geiriadur stoc gan lwytho...
    stoc = llwytho_stoc()
```

ol_ffeil_stoc - Gwall Ffeiliau #1

I lwytho'r stoc yn llwyddiannus i'r ffenestr siop, rhaid cywiro **2 wall** o fewn 'gui_ffeiliau.py'. Y cyntaf yw darparu'r enw ffeil cywir (mae'r llwybr yn cael ei gynhyrchu'n ddeinamig a dylai weithio ar bob dyfais).

```
def ol_ffeil_stoc(modd):

    cyfeiriadur = os.path.dirname(
        os.path.abspath(__file__))

    llwybr_stoc = os.path.join(cyfeiriadur, "eitemau.csv")
```

ol_ffeil_stoc - Gwall Ffeiliau #2

Mae'r ail yw llwytho'r pris o'r csv fel **rhif arnawf** - mae hyn oherwydd ei fod yn werth degol ac ni ellir ei deipio i gyfanrif.

```
def llwytho_stoc():  
  
    ~ ~ ~  
  
    pris = float(eitem[2].strip())
```

creu_bar_dewislen - ShopFace Error #3

Gellir dod o hyd i'r trydydd gwall o fewn y ffwythiant `creu_bar_dewislenn` ('**gui_siop.py**').

Fel y gwelir yn y mewngofnodi (yn benodol y botwm `cyflyno`), rhaid rhoi gorchymyn i'r botwm. Yn yr achos hwn, darperir y ffwythiant `newid_i_mewngofnodi` a basiwyd o **'gui_main.py'**, fel bod y ffenestr hon yn cael ei dinistrio a'r ffenestr mewngofnodi yn cael ei galw wrth allgofnodi.

Sylwch ar y defnydd o'r **ffwythiant lambda**, mae hyn yn caniatáu i ffwythiant gael ei basio i'r botwm heb gael ei alw ar unwaith - yn mewngofnodi fe wnaethon ni ddatrys hyn gyda chyfeiriad at ffwythiant nythol, ond yma mae angen pasio paramedr, felly ni fyddai cyfeirnod yn ddigonol. Mae hwn yn ateb gwell, gan fod ffwythiannau nythol yn erbyn confensiwn.

[illegible]

creu_tabl_stoc - Gwall Siop #4, 5

Mae **6 gwall** i'w cael o fewn y ffwythiant `creu_tabl_stoc` (`'gui_siop.py'`), mae'r rhain yn ymddangos yn olynol ac wedi'u grwpio i wallau #4 a 5. Mae'r holl wallau hyn yn gamgymeriadau syml yng ngwerth pa res i'w harddangos arni.

```
def creu_tabl_stoc():  
  
    ~ ~ ~  
  
    label_id = Label(ffram_tabl,  
                     text = "Rhif Eitem").grid(  
                     row = 0, column = 0)  
  
    label_enw = Label(ffram_tabl, text = "Name").grid(  
                     row = 0, column = 1)  
  
    ~ ~ ~  
  
    for i, eitem in enumerate(stoc):  
        label_id = Label(ffram_tabl, text = eitem).grid(  
                        row = i + 1, column = 0)  
        label_enw = Label(ffram_tabl,  
                           text = stock[item]["Name"]).grid(  
                           row = i + 1, column = 1)
```

creu_bar_prynu - Gwall Siop #6,7

Mae **2 wall** i'w cael o fewn y ffwythiant `creu_bar_prynu` (**'gui_siop.py'**). Y cyntaf o'r rhain yw bod nifer y lleoedd degol a ddangosir ar gyfer gwerthiannau yn y GUI yn anghywir. Diffinnir hyn gyda f-string, felly rhaid cywiro'r fformatio.

Mae'r ail wall eto yn y botwm. Mae'r botwm hwn yn fwy cymhleth na'r botwm diwethaf gan fod y ffwythiant `lambda` bellach yn cynnwys datganiad os - mae'n gwirio a yw'r ffwythiant `cynnig_pryniant` yn dychwelyd Gwir, ac os felly mae'n rhedeg y ffwythiant `prosesu_pryniant` (fel arall mae'n gwneud Dim). Mae hyn yn gwneud yn angenrheidiol fod y paramedr cywir yn cael ei basio i `cynnig_pryniant`.

```
def creu_bar_prynu():

    ~ ~ ~

    label_gwerth = Label(ffram_prynu,
        text = str(f"{gwerthiannau.get():.2f}"))
    label_gwerth.grid(row = 1, column = 2)

    # Creu botwm prynu i wneud pryniannau
    botwm_prynu = Button(ffram_prynu,
        text = "Prynu",
        bg = "White", fg = "Green",
        command = lambda: prosesu_pryniant(
            label_gwerth, prynu_mewn.get())
        if cynnig_pryniant(
            prynu_mewn.get()) else None)
```


diweddar_u_tabl_stoc - Gwall Siop #8

Dim ond **1 gwall** sydd yn bodoli o fewn `diweddar_u_tabl_stoc` ('`gui_siop.py`'). Ar ôl i'r tabl cyfredol gael ei ddinistrio, rhaid creu tabl arall yn ei le.

```
def diweddar_u_tabl_stoc():  
  
    global ffram_tabl  
    ffram_tabl.destroy()  
    creu_tabl_stoc()
```

cynnig_pryniant - Gwall Siop #9

Dim ond **1 gwall** sy'n bodoli o fewn y ffwythiant `cynnig_pryninat` ('`gui_siop.py`'). Os yw'r ymgais yn aflwyddiannus oherwydd mewnbwn annilys, yna dylid darparu gwall **defnyddiol** i'r defnyddiwr.

```
def cynnig_pryniant(mewnbwn):  
  
    ~ ~ ~  
    except (ValueError, KeyError):  
        messagebox.showwarning(  
            "Gwall Allwedd",  
            "Rhowch Allwedd Ddilys!")  
        return False
```

prosesu_pryniant - Gwall Siop #10

Dim ond **1 gwall** sy'n bodoli o fewn y ffwythiant `prosesu_pryniant` (`'gui_siop.py'`). Pryd bynnag y defnyddir newidynnau **eang** o fewn ffwythiant, rhaid **ail-ddatganu** o fewn y ffwythiant hwnnw (mae hyn i hysbysu Python bod y newidyn eang yn cael ei alw, nid newidyn lleol gyda'r un enw).

```
def prosesu_pryniant(label_gwerth, mewnbwn):  
  
    #...  
    global stoc  
    global gwerthiannau
```

ysgrifennu_stoc - Gwall Ffeiliau #3

Mae'r gwall olaf yn bodoli yn ôl yn `'gui_ffeiliau.py'` o fewn y ffwythiant `ysgrifennu_stoc`. Er mwyn ysgrifennu unrhyw newidiadau i ffeil, rhaid agor y ffeil yn y modd ysgrifennu, nid y modd darllen.

```
def ysgrifennu_stoc(stoc):  
  
    ffeil_stoc = ol_ffeil_stoc('w')
```

Nodiadau

Cystrawen Tkinter

Gweithred Cod Python

Mewnforio

```
import tkinter as tk      <- Gwell
OR
from tkinter import *     <- Syml
```

Er ei bod hi'n symlach mewnforio pob ffwythiant i'r cwmpas lleol (nid oes rhaid i chi roi 'tkinter' o'u blaenau) nid yw'n mewnforio is-fodiwlau (fel messagebox) a gellir trosysgrifennu ffwythiannau gyda'r un enw.

Bydd gweddill y twyll-len hon yn defnyddio'r mewnforio gwell.

```
root = tk.Tk()
```

Diffinio
Prif
Ffenestr

Rhaid creu'r brif ffenestr i ddal y GUI Tkinter, fel arfer rydym yn galw hwn yn root. Pe bai gennym sawl ffenestr, gallem ddefnyddio `tk.Toplevel()` i greu ffenestri newydd, ond dim ond un gwreiddyn sydd ei angen arnom ar gyfer y brif raglen.

Opsionau
Ffenestr

```
root.title("Bar Teitl - Top Ffenestr")
root.geometry("640x480")
```

Bydd y rhain yn gosod y teitl a'r dimensiynau ar gyfer y ffenestr gyfan. Mae llawer o opsiynau yn bodoli a gellir defnyddio llawer o'r opsiynau Ffrâm ar y Ffenestr hefyd.

```
fy_prif_ffram = tk.Frame(root)
```

Diffinio
Ffram

TheY Ffrâm yw'r bloc adeiladu sylfaenol ar gyfer cynnwys o fewn y Ffenestr. Rydym yn neilltuo rheolwr geometreg i'r Ffrâm i ddewis sut rydym yn gosod y widgets Tkinter ynddi. Gallwn hefyd osod Fframiau o fewn Fframiau!

Rheolwyr Geometreg

Noder bod Rheolwyr Geometreg yn amlbwrpas yn eu defnydd. Fe'u defnyddir i osod widgets a Fframiau y tu mewn i Fframiau neu Ffenestri eraill.

Nid oes angen glynu wrth un rheolwr geometreg, a gellir eu defnyddio i gyd o fewn rhaglen - fodd bynnag, rhaid i bob Ffrâm a Ffenestr ddefnyddio un rheolwr (ond gallai Ffrâm a osodir ynddo ddefnyddio un arall).

Mae gan bob rheolwr geometreg fwy o opsiynau a rheolaeth fanwl nag isod!

Cystrawen Tkinter

Gweithred Cod Python

Rheolwr
Geometreg -
Pack

```
fy_prif_ffram.pack()
```

```
fy_prif_ffram.pack(side = LEFT)
```

Pack yw'r rheolwyr geometreg symlaf a bydd yn pacio widgets mewn i'r gofod rhydd gyda'r ymdrech leiaf (a chydâ'r addasiad lleiaf).

Rheolwr
Geometreg -
Grid

```
fy_prif_ffram.grid(row = 0, column = 0)
```

Grid yw'r canolbwynt braf o blith y rheolwyr geometreg ac mae'n caniatáu ichi osod widgets fel pe baent ar rid (bydd hyn yn addasu i faint eich ffenestr, ond gallwch gael cymaint o resi/ colofnau ag y dymunwch).

Rheolwr
Geometreg -
Place

```
fy_prif_ffram.place(x = 0, y = 0,
                    width = 15,
                    height = 10)
```

Place yw'r rheolwr geometreg fwyaf cymhleth, ond mae hefyd yn caniatáu'r addasiadau mwyaf manwl gywir yn faint a lleoliad widgets.

Cystrawen Tkinter

Gweithred Cod Python

Dinistrio
Ffram

```
fy_ffram.destroy()
```

Mae hyn yn dinistrio'r Ffram neu'r Ffenestr gyfredol. Mae'n ddefnyddiol ar gyfer dewislenni opsiynau neu sgriniau mewngofnodi ac ati.

Prif Dolen

```
root.mainloop()
```

Mae hyn yn hanfodol gan ei fod yn diffinio dolen y rhaglen (h.y. dyma sy'n cadw'r rhyngwyneb defnyddiwr graffigol o fewn dolen fel y gellir mewnbwnnu testun, pwysu botymau, ac ati).

Yn gyffredinol, dim ond un brif ddolen sydd angen ei diffinio ar waelod y cod sy'n diffinio'r brif raglen/ffenestr.

Labeli

```
fy_label = Label(fy_ffram,  
                 text = "Helo i ti")  
fy_label.grid(row = 0, column = 0)
```

Mae label yn syml yn ddarn o destun sy'n ymddangos ar y GUI - yma bydd y label "Helo i ti" yn cael ei osod yng nghornel chwith uchaf fy_ffram.

Meysydd
Mewnbwn

```
fy_maes = Entry(fy_ffram)  
fy_maes.grid(row = 1, column = 1)
```

Mae maes mewnbwn yn caniatáu i'r defnyddiwr nodi testun. Ar ôl i'r testun gael ei nodi, gallem gael botwm sy'n adalw cynnwys fy_maes.

Sylwch fod y maes mewnbwn hwn ar hyn o bryd yn y gornel dde isaf - cyn belled nad ydym yn ychwanegu unrhyw widgets newydd i res neu golofn 2.

Cystrawen Tkinter

Gweithred Cod Python

Botwm
(syml)

```
def main():

    def shwmae_nythol():

        print("Mae hon yn ffwythiant
              nythol sy'n argraffu
              un peth - Shwmae!")

    botwm_shwmae = Button(
        fy_ffram, text = "Shwmae",
        bg = "Blue", fg = "Grey",
        command = shwmae_nythol
    )
```

Er mwyn i fotwm wneud unrhyw beth, rhaid rhoi gorchymyn iddo - fodd bynnag, byddai rhoi ffwythiant iddo yn achosi i'r ffwythiant hwnnw gael ei galw ar unwaith (yn hytrach na galw wrth glicio ar y botwm).

Felly yn lle hynny, gallwn basio **cyfeiriad** at **ffwythiant nythol** (newidyn yw **cyfeiriad** - nid ydym yn cynnwys y cromfachau - a rhaid nythu'r ffwythiant y tu mewn i'r **ffwythiant cyfredol** fel ei bod o fewn cwpas!).

Mae hyn yn gweithio'n gymharol dda, ond ni ellir defnyddio ffwythiant nythol mewn mannau eraill felly gallant arwain at god ailadroddus. Gallant hefyd niweidio darllenadwyedd os ydynt yn enwedig hir.

Cystrawen Tkinter

Gweithred Cod Python

Botwm
(safonol)

```
def argraffu(mewnbwn) :

    print("Dywedoch chi: " + mewnbwn)

def main() :

    fy_maes = Entry(fy_ffram)

    botwm_iawn = Button(
        fy_ffram, text = "Iawn",
        bg = "Blue", fg = "Grey",
        command = lambda:
            argraffu(fy_maes.get())
    )
```

Drwy ddefnyddio **ffwythiant lambda**, gallwn osgoi gorfod pasio **cyfeirnod** yn unig a gallwn basio'r ffwythiant cyfan yn lle hynny. Ni fydd y ffwythiant yn cael ei rhedeg ar unwaith pan gaiff ei ragflaenu gan y gorchymyn **lambda**.

Gall hyn fod yn ofyniad hanfodol ar gyfer unrhyw ffwythiannau sy'n cymryd paramedrau! Er bod yn bosibl creu datrysiadau clyfar i osgoi pasio paramedrau, mae'n llawer haws ysgrifennu 'lambda: '.

Cystrawen Tkinter

Gweithred Cod Python

Botwm
(lefel uwch)

```
def argraffu_na(mewnbwn):
    print("Rhif = " + mewnbwn)

def os_na(mewnbwn):
    if mewnbwn.isnumeric():
        return True
    else:
        return False

def main():
    fy_maes = Entry(fy_ffram)

    botwm_iawn = Button(
        fy_ffram, text = "Iawn",
        bg = "Blue", fg = "Grey",
        command = lambda:
            argraffu_na(fy_maes.get())
            if os_na(fy_maes.get())
            else print("Rhif yn unig")
    )
```

Gallwn wneud ein **ffwythiannau lambda** yn llawer mwy cymhleth drwy eu hatodi â datganiadau os! Yn ei hanfod, ffwythiant bach, tafladwy y gellir ei ysgrifennu ar un llinell yw ffwythiant lambda - felly gallant fod yn eithaf cymhleth!

Dim ond os yw cynnwys y blwch mynediad yn rhifiadol y bydd y ffwythiant hwn yn argraffu'r hyn y mae'r defnyddiwr yn ei roi yn y blwch mynediad, **fel arall** bydd yn nodi mai dim ond rhifau y mae'n eu derbyn!

Cystrawen Tkinter

Gweithred Cod Python

Blwch
Negeseuon

```
from tkinter import messagebox

messagebox.showinfo("Teitl", "Neges")
messagebox.showerror("Teitl", "Neges")
messagebox.showwarning("Teitl", "Neges")

messagebox.showerror("Gwall",
                     "Rhif Yn Unig!",
                     command = lambda:
                         clirio_maes())
```

Mae defnyddio'r ffwythiant messagebox yn Tkinter yn caniatáu i ni ddefnyddio'r naidlen rhagosodedig ein system (h.y. ar Windows bydd yn edrych fel ffenestri naidlen Windows...). Mae hon yn ffordd wych o ychwanegu proffesiynoldeb at ein prosiect!

Gall y blwch negeseuon gynnwys teitl syml ar y bar teitl a neges yn y blwch (mae'r gwahanol ffwythiannau `info/error/warning` ac ati yn newid yr eicon sy'n ymddangos!). Fodd bynnag, gallwn fynd yn fwy cymhleth trwy ychwanegu gorchmynion (trwy **ffwythiant lambda**) sy'n rhedeg ar ôl cau'r blwch.

Mae yna lwyth o amrywiadau i hyn, a llawer mwy o fathau o flychau (`.askquestion`, `.askokcancel`) sy'n caniatáu inni ddychwelyd a phrosesu mewnbwn defnyddwyr!

