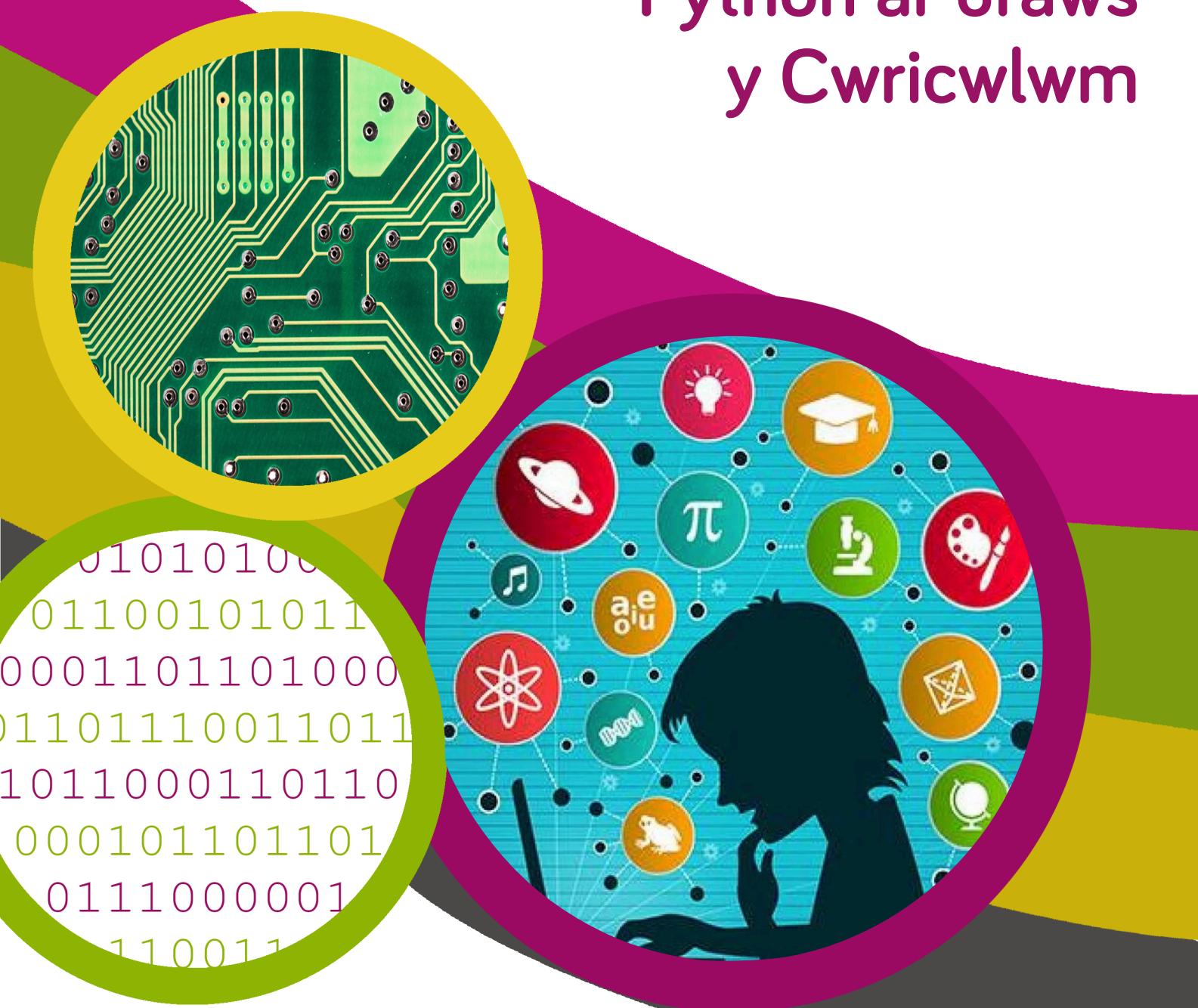


# technocamps

## Python ar draws y Cwricwlwm



Cronfa Gymdeithasol Ewrop  
European Social Fund



Prifysgol  
Abertawe  
Swansea  
University



Cardiff  
Metropolitan  
University

Prifysgol  
Metropolitan  
Caerdydd

i.twales

PRIFYSGOL  
ABERYSTWYTH  
UNIVERSITY

PRIFYSGOL  
glyndŵr  
Wrecsam | Wrexham  
glyndŵr  
UNIVERSITY

University of  
South Wales  
Prifysgol  
De Cymru

# Cyflwyniad

## Trosolwg

Gellir gweithredu codio ar draws yr holl Feysydd Dysgu a Phrofiad, gan atgyfnerthu dysgu yn yr ystafell ddosbarth a gwella llythrennedd digidol yn y broses.

Yn y byd heddiw, mae llythrennedd digidol yn sgil hanfodol i ddysgwyr ei ddatblygu. Mae'r gofynion technolegol ar gyfer swyddi yn cynyddu'n barhaus, a bydd dechreuad cryf mewn sgiliau digidol yn paratoi dysgwyr ac yn rhoi mantais iddynt.

Adnoddau Digidol:

<https://tc1.me/educonf22resources>

Tiwtorialau YouTube:

<https://tc1.me/progacrosscurriculum>

## Adnoddau Ar-Lein

## Syniadau i Raglennu



### Iechyd a Lles

- Pyramid Bwyd
- Pong



### Leithoedd, Llythrennedd a Chyfathrebu

- Cwis Cyfeithu
- Cwis Rhagenwau



### Mathemateg a Rhifedd

- Llunio Siapau
- Amcangyfrif Pi



### Celfyddydau Mynegiannol

- Celf Algorithm
- Cydweddu Steiliau



### Gwyddoniaeth a Thechnoleg

- Cyflyrau Mater
- Cylched Dŵr



### Dyniaethau

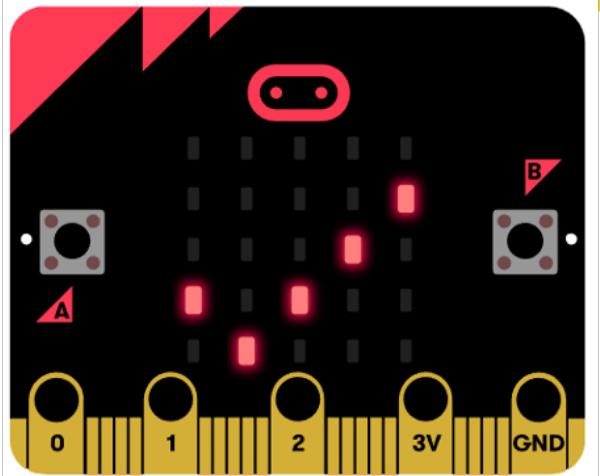
- Llinell Amser Rhwgweithiol
- Efelychiad Ymfudiad

# Pedomedr Micro:bit

## Pedomedr Micro:bit

Dyma god enghreifftiol ar sut y gallwch ddefnyddio BBC Micro:bit i wneud pedomedr.

Bydd y tudalennau canlynol yn rhoi'r cod sydd ei angen arnoch i gael BBC Micro:bit i gyfrif y camau a gymerwch.



## Mewnforio Llyfrgelloedd

Yn gyntaf ewch i'r wefan ganlynol:

<https://python.microbit.org/v/3>

Mae hyn yn eich galluogi i ryngweithio â BBC Micro:bit efelychiedig. Dileu'r holl god sydd yna'n barod.

Er mwyn defnyddio'r BBC Micro:bit i gyfrif ein camau i ni, bydd angen i ni fewnforio un llyfrgell. Byddwn yn defnyddio amrywiaeth o ffwythiannau o'r llyfrgell, felly gallwn fynd ymlaen i fewnforio'r llyfrgell gyfan.

```
from microbit import *
```

Bydd hyn yn mewnforio'r holl lyfrgelloedd sydd eu hangen er mwyn i'n cod Python ryngweithio â'r BBC Micro:bit.

# Pedomedr Micro:bit

## Creu'r Pedomedr

Yn gyntaf rydym yn mewnforio ein llyfrgell Micro:bit, ac yn mewnforio pob dull ohoni (fel y dangosir gan y \*).

Mae'r cod hwn wedyn yn datgan a gosod newidyn o'r enw camau i 0. Bydd hyn yn cadw golwg ar faint o gamau rydym yn eu cymryd.

Rydyn ni nawr yn dechrau dolen, gydag amod a fydd bob amser yn wir. Gwneir hyn i sicrhau bod ein cod yn parhau i redeg am byth.

O fewn y ddolen hon rydym yn ychwanegu datganiad os (if statement). Mae hwn yn amodolyn, pan yn wir, gweithredir y cod a osodir ar ôl y datganiad. Yn yr achos hwn, rydym yn gwirio a yw'r cyflymromedr wedi canfod bod y ddyfais wedi'i ysgwyd.

Os yw'r ddyfais wedi'i ysgwyd, rydym ni'n ychwanegu un at ein newidyn camau, ac yna rydym ni'n arddangos y nifer bresennol o gamau.

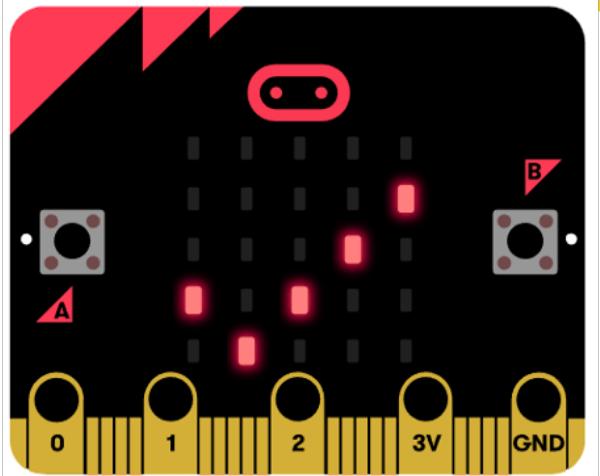
```
from microbit import *
camau = 0
while True:
    if accelerometer.was_gesture('shake'):
        camau += 1
        display.show(camau)
```

# Cerddoriaeth Micro:bit

## Cerddoriaeth Micro:bit

Dyma god enghreifftiol ar sut y gallwch ddefnyddio BBC Micro:bit i gyfansoddi a chreu cerddoriaeth.

Bydd y tudalennau canlynol yn rhoi'r cod sydd ei angen i gael BBC Micro:bit i chwarae'r gân Aderyn Melyn.



## Mewnforio Llyfrgelloedd

Yn gyntaf ewch i'r wefan ganlynol:

<https://python.microbit.org/v/3>

Mae hyn yn eich galluogi i ryngweithio â BBC Micro:bit efelychiedig. Dileu'r holl god sydd yna'n barod.

Er mwyn defnyddio'r BBC Micro:bit i greu cerddoriaeth i ni, bydd angen i ni fewnforio un llyfrgell a ffwythiant. Byddwn yn defnyddio amrywiaeth o ffwythiannau o'r llyfrgell, felly gallwn fynd ymlaen a mewnforio'r llyfrgell gyfan.

```
from microbit import *
import music
```

Yn gyntaf, rydym yn mewnforio'r llyfrgell microbit. Mae hyn yn ein galluogi i ddefnyddio Python i reoli a rhngweithio â'n BBC Micro:bit.

Yn ail, rydym yn mewnforio'r llyfrgell music. Gallwn ddefnyddio'r ffwythiant "play" a rhoi traw a hyd pob nodyn iddo.

# Cerddoriaeth Micro:bit

## Creu Nodiadau

Rydym yn creu nodiadau gan ddefnyddio Python gan ddefnyddio'r llyfrgell music a fewnforiwyd gennym yn gynharach. Mae'r ffwythiant play yn gofyn am uchafswm o ddau ddarn o wybodaeth: y nodyn, a'r hyd. Nid yw'n hanfodol rhoi'r hyd, ond rhaid inni ddarparu nodyn bob amser.

Mae'r darn isod o god yn chwarae nodyn C4 (C Canol), am 4 bar (hyd safonol nodyn gyda'r ffwythiant play).

```
music.play(['C4:4'])
```

Nid ydym yn gyfyngedig i un nodyn fesul llinell yn unig, gallwn linio nodau lluosog at ei gilydd, i'w chwarae un ar ôl y llall.

```
music.play(['C4:4', 'B4', 'D3:3'])
```

Gallwn hefyd ychwanegu seibiannau yn ein cod, a fydd yn rhoi gorffwys i ni rhwng nodiadau. Sylwch ar y gweddill rhwng nodiadau B4 a D3.

```
music.play(['C4:4', 'B4', 'R', 'D3:3'])
```

Yn olaf, gallwn gyflwyno dolenni, i ailadrodd ein nodiadau sawl gwaith. Mae'r cod canlynol yn ailadrodd pob un o'r 3 nodyn yn eu trefn 2 waith, gan arwain at gyfanswm o 6 nodyn: C4, B4, D3, C4, B4, D3.

```
for i in range(2):
```

```
    music.play(['C4', 'B4', 'D3'])
```

# Cerddoriaeth Micro:bit

## Aderyn Melyn

Dyma raglen enghreifftiol ar gyfer chwarae'r gân "Yellow Bird, Up High in Banana Tree" neu "Aderyn Melyn".

```
from microbit import *
import music

for i in range(2):
    music.play(['C4:4', 'C5:12', 'B4:4', 'C5:2',
'C5:2', 'R', 'A4:4', 'A#4:2', 'A#4', 'A#4', 'A#4',
'A#4', 'A#4', 'C5', 'A4:4', 'A4', 'R:2'])

for i in range(2):
    music.play(['D4:4', 'D4:2', 'G4:3', 'A#4:2',
'D5', 'R', 'C4:4', 'C4:2', 'E4:3', 'F4:2', 'G4',
'R'])

music.play(['C4', 'C4:3', 'A#4:4', 'A4:2',
'G4:3', 'R:1', 'F4:5'])
```

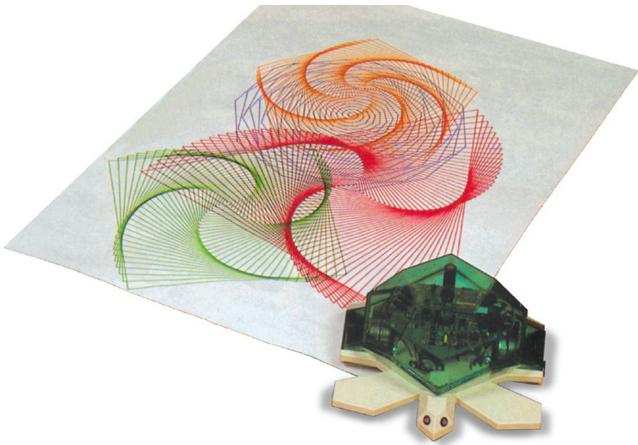
**Sylwer:** Mae rhai o'r llinellau cod yn rhy hir i ffitio ar un llinell yn y llyfr gwaith hwn, ni ddylech roi llinellau newydd cyn gorffen y ffwythiant play. Mae hyn yn wir am y cod sy'n dilyn hefyd.

# Celf Turtle

## Celf Turtle

Dyma god enghreifftiol ar gyfer creu siapiau a chelf trwy Python gan ddefnyddio'r Llyfrgell **turtle**.

Bydd y tudalennau canlynol yn dangos dwy raglen: sut i lunio sgwâr, a sut i wneud sbirograff.



## Mewnforio Llyfrgelloedd a Mynediad i'r Crwban

I ddechrau gwneud rhaglen crwban, ewch i'r wefan hon (mae'n bosib gwneud hwn yn IDLE os ydych wedi gosod Python ar eich dyfais):

<https://www.pythonsandbox.com/turtle>

Unwaith y byddwch ar y wefan, diléwch unrhyw god yn y derfynell ar ochr chwith y sgrin, byddwn yn gwneud un ein hunain o'r dechrau!

Er mwyn defnyddio'r Crwban i wneud siapiau a chelf i ni, yn gyntaf mae angen i ni fewnforio'r llyfrgell **turtle**

```
import turtle
```

## Llunio Sbirograff

Yn gyntaf rydym yn mewnforio ein llyfrgell a creu newidynnau. Mae mewnforio'r llyfrgell yn ein galluogi i gael mynediad i'r ffwythiannau crwban, tra bod y newidynnau yn caniatáu inni olygu'r math o grwban y byddwn yn ei ddefnyddio. Byddwn yn galw ein crwban yn 'pin' gan ein bod yn tynnu llun, ac yn ei wneud yn siâp crwban, ac yn ei wneud yn wyrdd.

```
import turtle  
pin = turtle.Turtle()  
pin.shape("turtle")  
pin.color("green")
```

Nawr gallwn roi gorchmynion penodol i'n crwban. Mae gorchmynion fel **forward** yn gwneud i'r crwban symud mewn llinell syth, i nifer penodol o 'unedau'. Gallwn hefyd wneud i'n crwban gylchdroi naill ai i'r **left** neu'r **right**, gan ddefnyddio graddau.

```
pin.forward(100)  
pin.right(90)  
pin.forward(100)  
pin.right(90)  
pin.forward(100)  
pin.right(90)  
pin.forward(100)  
pin.right(90)
```

Y canlyniad fydd crwban a fydd yn symud mewn sgwâr i ni.

Awgrym: Meddyliwch am y crwban fel un sydd â 'symudiad tanc', dim ond ymlaen neu yn ôl y gall fynd, a rhaid iddo cylchdroi ar y fan i newid cyfeiriad.

# Celf Turtle

## Llunio Sbirograff

Yn gyntaf rydym yn mewnforio ein llyfrgell a newidynnau, fel y gwnaethom o'r blaen.

```
import turtle
pin = turtle.Turtle()
pin.shape("turtle")
pin.color("green")
```

Nawr yn wahanol i'r tro diwethaf, byddwn yn gwneud dolen i ailadrodd ein cod sawl gwaith. Rydym wedi nodi ein bod am i'n cod ailadrodd 350 o weithiau. Bydd y newidyn **i** yn cael ei ddefnyddio i gyfrif hyd at 350, gan ddechrau o 1, a chynyddu gan 1 bob iteriad (iteration).

Mae hyn yn golygu i ddechrau mai dim ond 1 uned y bydd ein crwban yn symud, ond bydd ei symudiadau yn cynyddu wrth i'r cod barhau.

```
for i in range(350):
    pin.forward(i)
    pin.right(98)
```

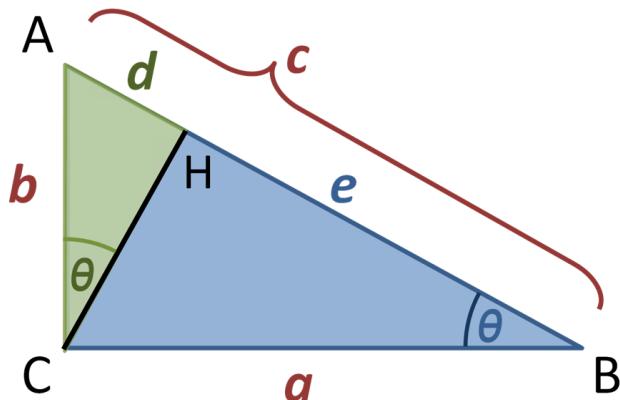
Y canlyniad fydd crwban a fydd yn gwneud sbirograff i ni.

Beth am droi chwarae gyda rhai o'r rhifau? Beth sy'n digwydd os byddwch chi'n newid yr ongl?

## Pythagoras

Dyma god enghreifftiol ar gyfer cyfrifiannell theorem Pythagoras yn Python.

Bydd y tudalennau isod yn rhoi'r cam nesaf yn y rhaglen i chi.



## Mewnforio Llyfrgelloedd

I ddefnyddio'r theorem Pythagoras i gyfrifo hyd ac onglau triongl sgwâr bydd angen i ni fewnforio rhai ffwythiannau o'r llyfrgell **math** python.

```
from math import sqrt, asin, acos, atan

print('Cyfrifiannell theorem Pythagoras! Cyfrifwch
ochrau ac onglau triongl')
print('Tybiwch mai'r ochrau yw a, o ac h, lle mai
h yw'r hypotenws')

hafaliad = input('Pa ochr (a, o, h) ydych chi am
ei chyfrifo?')
```

Mae'r rhaglen yn dechrau trwy argraffu esboniad o'i swyddogaeth, a gofyn i'r defnyddiwr fewnbynnu pa ochr y dylai ei chyfrifo.

## Cyfrifo Ochr A

I gyfrifo hyd 'a' (yr ochr gyferbyn â'r ongl  $\theta$ ) mae datganiad os (if statement) yn cael ei ddefnyddio i wirio mewnbwn y defnyddiwr a ddewiswyd.

Yna mae'r rhaglen yn gofyn i'r defnyddiwr am hyd ochrau **o** ac **h**.

Yna cyfrifir yr hyd gan ddefnyddio'r theorem Pythagoras wedi'i aildrefnu.

Gan ddefnyddio **sin<sup>-1</sup>** mae'r ongl theta yn cael ei gyfrifo.

Mae'r ddau werth yn cael eu hargraffu i'r sgrin ynghyd â llinyn yn disgrifio'r gwerth (Nodyn: mae **\n** yn cael ei ddefnyddio i osod llinell newydd yn Python).

```
if hafaliad == 'a':
    ochrO = int(input('Mewnbwnn hyd ochr o: '))
    ochrH = int(input('Mewnbwnn hyd ochr h: '))

    ochrA = sqrt((ochrH * ochrH) - (ochrO * ochrO))

    theta = asin(ochrO / ochrH)

    print('\nHyd ochr a yw : ' + str(ochrA))
    print('\nYr ongl a roddir mewn radianau yw: ' +
str(theta))
```

## Cyfrifo Ochr B

I gyfrifo hyd 'o' (yr ochr gyferbyn â'r ongl  $\theta$ ) defnyddir gosodiad **elif** (else if) i ychwanegu amod at y datganiad os. Yna mae'r rhaglen yn gofyn i'r defnyddiwr am hyd ochrau **a** ac **h**.

Yna cyfrifir yr hyd gan ddefnyddio'r theorem Pythagoras wedi'i aildrefnu.

Gan ddefnyddio **cos<sup>-1</sup>** mae'r ongl theta yn cael ei gyfrifo.

Mae'r ddau werth yn cael eu hargraffu i'r sgrin ynghyd â llinyn yn disgrifio'r gwerth.

```
elif hafaliad == 'o':  
    ochrA = int(input('Mewnbynnu hyd ochr a: '))  
    ochrH = int(input('Mewnbynnu hyd ochr h: '))  
  
    ochrO = sqrt((ochrH * ochrH) - (ochrA * ochrA))  
  
    theta = acos(ochrA / ochrH)  
  
    print('\nHyd ochr a yw: ' + str(ochrO))  
    print('\nYr ongl mewn radianau yw: ' + str(theta))
```

## Cyfrifo Ochr C

I gyfrifo hyd 'h' (yr hypotenws) defnyddir gosodiad **elif** i ychwanegu amod arall at y datganiad os Yna mae'r rhaglen yn gofyn i'r defnyddiwr am hyd ochrau **a** ac **o**.

Yna cyfrifir yr hyd gan ddefnyddio'r theorem Pythagoras. Gan ddefnyddio **tan<sup>-1</sup>** mae'r ongl theta yn cael ei gyfrifo. Mae'r ddau werth yn cael eu hargraffu i'r sgrin ynghyd â llinyn yn disgrifio'r gwerth.

Defnyddir datganiad **else** i sicrhau bod mewnbwn dilys wedi'i roi.

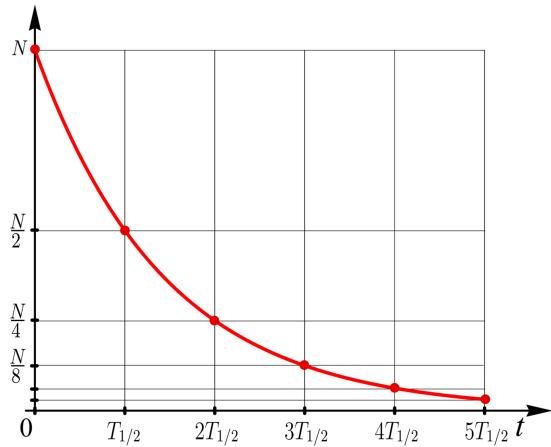
```
elif hafaliad == 'h':  
    ochrA = int(input('Mewnbynnu hyd ochr a: '))  
    ochrO = int(input('Mewnbynnu hyd ochr o: '))  
  
    ochrH = sqrt((ochrA * ochrA) + (ochrO * ochrO))  
  
    theta = atan(ochrO / ochrA)  
  
    print('\nHyd ochr a yw : ' + str(ochrH))  
    print('\nYr ongl mewn radianau yw: ' + str(theta))  
  
else:  
    print('Dewiswch ochr a, o neu h')
```

## Dadfeiliad Niwclear Ar Hap

Dyma god enghreifftiol ar gyfer Dadfeiliad Niwclear yn Python.

Bydd y rhaglen yn rhedeg efelychiad ar hap ac yn allbwn y canlyniadau, yna gellir plotio'r rhain yn Excel.

Bydd y tudalennau isod yn rhoi'r cam nesaf yn y rhaglen i chi.



## Mewnforio Llyfrgelloedd

Er mwyn efelychu dadfeiliad ar hap niwclysau ymbelydrol bydd angen i ni fewnforio rhai ffwythiannau o'r llyfrgelloedd mathemateg a gwyddoniaeth Python.

```
from random import randint
```

Gellir naill ai mewnforio llyfrgelloedd yn llawn lle bydd angen mewnforio'r llyfrgell ar gyfer pob ffwythiant (gan ddefnyddio `import _`), neu gellir mewnforio ffwythiannau penodol o lyfrgell, y gellir eu galw'n uniongyrchol wedyn (gan ddefnyddio `from _ import _`).

## Cychwyn

I gychwyn y rhaglen gofynnir i'r defnyddiwr fewnbynnu nifer y niwclysau ar ddechrau'r efelychiad a'r tebygolrwydd y bydd niwclews yn dadfeilio bob eiliad.

Mae'r gwerthoedd hyn yn cael eu cadw fel cyfanrif (integer) a rhif pwynt arnawf (float) yn y drefn honno, fel bod rhaid mewnbynnu nifer y niwclysau fel rhif llawn lle gall y tebygolrwydd gymryd unrhyw werth.

Mae rhai gwerthoedd cychwynnol yn cael eu diffinio fel bod yr efelychiad yn dechrau ar 0 eiliad gyda'r hanner oes yn anhysbys (bydd y rheswm bod angen hyn yn dod yn glir).

Nodyn: Mae'r **llinellau coch** yn sylwadau o fewn y cod. Gellir ychwanegu sylwadau yn Python gyda **#**.

```
# Nifer cychwynnol o niwclysau
niwclysauCychwynnol = int(input('Beth yw nifer
cychwynnol y niwclysau?'))
niferNiwclysau = niwclysauCychwynnol

# Y tebygolrwydd y bydd niwclews penodol yn dadfeilio
tebygolrwydd = float(input('\nRhowch werth canrannol
ar gyfer y siawns y bydd niwclews yn dadfeilio bob
eiliad h.y. 50:'))

# Gosod gwerthoedd cychwynnol ar gyfer newidynnau
eiliadau = 0
hannerOes = 0
hannerOesWediBod = False
```

## Prif Rhaglen

Mae prif gorff y rhaglen yn cynnwys dolen **while** a fydd yn rhedeg nes bod yr holl niwclysau wedi dadfeilio.

Bob tro mae'r ddolen yn rhedeg mae nifer y niwclysau sy'n weddill yn cael ei argraffu ac mae nifer yr eiliadau sy'n cael eu pasio yn cynyddu.

Mae'r ddolen **for** yn rhedeg trwy bob niwcleysws sy'n weddill ac yn cynhyrchu cyfanrif ar hap rhwng 0 a 100.

Os yw'r haprif yn llai na'r tebygolrwydd o dadfeilio, yna mae'r niwcleysws yn dadfeilio ac mae nifer y niwclysau'n gostwng gan 1.

```
while niferNiwclsau > 0:
```

```
# Argraffu nifer y niwclysau sy'n weddill a chynyddu'r amser gan 1
```

```
    print(niferNiwclsau)
    eiliadau += 1
```

```
# Ystyried pob niwcleys yn ei dro a phenderfynu a yw'n dadfeilio ai peidio
```

```
    for niwcleys in range(niferNiwclsau):
        rand = randint(0,100)
        if rand < tebygolrwydd:
            niferNiwclsau -= 1
```

## Amcangyfrif Hanner Oes

Fel estyniad i'r brif raglen gellir amcangyfrif yr hanner oes. Gwneir hyn o fewn y ddolen **for**.

Os yw nifer y niwclysau sy'n weddill yn llai na hanner y rhif cychwynnol yna mae amcangyfrif o werth yr hanner oes yn cael ei gofnodi. Felly dim ond un gwerth sy'n cael ei gymryd, mae gweithrediad **and** wedi'i gynnwys i wirio gwerth Boole sydd wedyn yn cael ei newid.

Mae'r amcangyfrif ar gyfer hanner oes yn cael ei argraffu ynghyd â'r eiliad o'r blaen i roi amrediad ar gyfer pryd y digwyddodd yr hanner oes. Mae hwn wedi'i ysgrifennu y tu allan i'r ddolen **for**.

# Storio gwerth yr hanner oes pan fydd mwy na hanner y niwclysau wedi dadfeilio

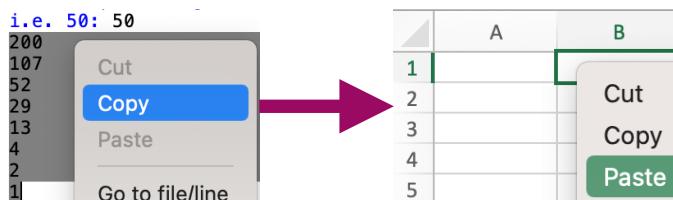
```
if niferNiwclysau < round(niwclysauCychwynnol/2) and
hannerOesWediBod == False:
    hannerOes = eiliadau
    hannerOesWediBod = True

print('Hanner oes y sampl hwn yw rhwng %d a %d eiliad.' %
(hannerOes - 1, hannerOes))
```

# Dadfeiliad Niwclear

## Plotio Data

Gellir copio'r gwerthoedd sy'n cael eu hallbynnu yn y Python Shell i Excel mewn un trwy dewis pob un ohonynt, eu copio a'u gludo yn Excel.



Gludwch y gwerthoedd yng ngholofn B, fel ei fod yn plotio fel yr echelin y. Gellir ychwanegu canlyniadau pellach yn y colofnau canlynol, i blotio efelychiadau lluosog.

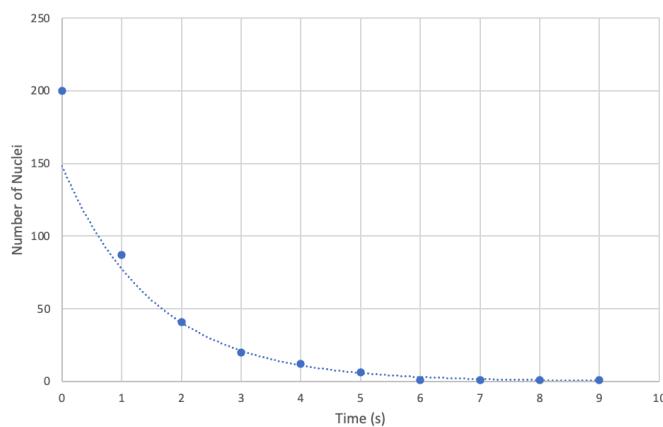
Yng ngholofn A ychwanegwch yr eiliadau a basiwyd o 0. Gellir gwneud hyn trwy nodi 0 ac 1, dewis y dduau, yna llusgo'r gornel isaf.

The diagram shows two screenshots of an Excel spreadsheet. The first screenshot on the left shows column A with values 0, 1, 2, 3, 4, 5, 6, 7, 8. The second screenshot on the right shows column A with values 1, 2, 3, 4, 5, 6, 7, 8, followed by column B with values 200, 107, 52, 29, 13, 4, 2, 1. A large purple arrow points from the first screenshot to the second, indicating the addition of time values to column A.

| A | B   |
|---|-----|
| 1 | 200 |
| 2 | 107 |
| 3 | 52  |
| 4 | 29  |
| 5 | 13  |
| 6 | 4   |
| 7 | 2   |
| 8 | 1   |

| A | B   |
|---|-----|
| 1 | 200 |
| 2 | 107 |
| 3 | 52  |
| 4 | 29  |
| 5 | 13  |
| 6 | 4   |
| 7 | 2   |
| 8 | 1   |

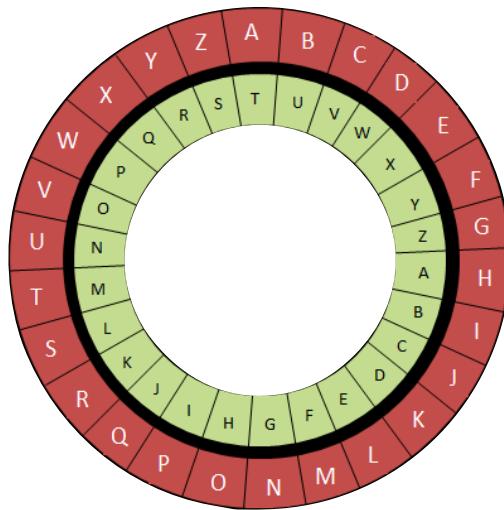
Dewiswch yr holl ddata a phlotiwch trwy Insert > Charts > X Y (Scatter). Ychwanegu Tueddiad Esbonyddol o Add Chart Element i ffitio'r data. Gellid gosod llinell duedd i ganlyniad cyfartalog llawer o efelychiadau.



## Seiffr Caesar

Dyma god enghreifftiol ar gyfer rhaglen Cesar Cipher yn Python.

Bydd pob tudalen yn rhoi ffwythiant bosibl i chi ei gweithredu, gan gynyddu cymhlethdod y rhaglen.



## Diffinio Newidynnau

Er mwyn adeiladu Seiffr Cesar gweithredol bydd angen i ni ddiffinio'r set o symbolau (nodau) y byddwn yn eu defnyddio ac yn symud i amgryptio ein testun.

```
global alpha, bet
alpha = 'ABCDEFGHIJKLMNPQRSTUVWXYZ'
beta = 'abcdefghijklmnopqrstuvwxyz'
```

Yn yr enghraift hon crëwyd dau newidyn ar gyfer llythrennau mawr a bach. Gallem gynnwys un arall ar gyfer rhifau, un arall ar gyfer atalnodau, neu fel arall eu cynnwys mewn un newidyn seiffr mawr!

Mae'r newidynnau wedi'u gwneud yn rhai eang (global) fel y gellir eu defnyddio yn unrhyw un o'r ffwythiannau canlynol, gan osgoi gorfod eu trosglwyddo i bob un.

## Ffwythiant Amgryptio

Bydd y ffwythiant hon yn amgryptio unrhyw destun plaen sy'n cael ei rhoi iddo gan ddefnyddio'r allwedd a nodir a'r newidynnau eang.

```
def amgryptio(testun, allwedd):
    canlyniad = ''

    for llythyren in testun:

        if llythyren.isupper():
            rhif = alpha.find(llythyren)
            rhif += allwedd

            if rhif > 25:
                rhif -= len(alpha)
            canlyniad += alpha[rhif]

        elif llythyren.islower():
            rhif = bet.find(llythyren)
            rhif += allwedd

            if rhif > 25:
                rhif -= len(bet)
            canlyniad += bet[rhif]

    else:
        canlyniad += llythyren

    return canlyniad
```

**Nodyn:** Nid oes galwad na gorchymyn print yn y ffwythiant hon - i alw ac argraffu'r ffwythiant gallem ddefnyddio rhywbeth fel `print(amgryptio('Fy Nhestun Plaen', 5))`

## Ffwythiant Dadgryptio

Bydd y ffwythiant hon yn dadgryptio unrhyw destun wedi'i amgryptio sy'n cael ei rhoi iddo gan ddefnyddio'r allwedd a nodir a'r newidynnau eang.

```
def dadgryptio(testun, allwedd):
    canlyniad = ''

    for llythyren in testun:

        if llythyren.isupper():
            rhif = alpha.find(llythyren)
            rhif -= allwedd

            if rhif < 0:
                rhif += len(alpha)
            canlyniad += alpha[rhif]

        elif llythyren.islower():
            rhif = bet.find(llythyren)
            rhif -= allwedd

            if rhif < 0:
                rhif += len(bet)
            canlyniad += bet[rhif]

    else:
        canlyniad += llythyren

    return canlyniad
```

**Nodyn:** Nid oes galwad na gorchymyn argraffu yn y ffwythiant hon - i alw ac argraffu gallem ddefnyddio rhywbeth fel `print(dadgryptio('khor', 3))`

## Ffwythiant Hacio

Bydd y ffwythiant hon yn hacio unrhyw destun wedi'i amgryptio sy'n cael ei rhoi iddo gan ddefnyddio dull "Brute-Force" i redeg trwy bob symbol yn y newidynnau eang.

Mae'r ffwythiant hon yn cynnwys un ddolen ychwanegol i roi cynnig ar bob allwedd bosibl ac yna argraffu'r canlyniadau.

```
def hacio(testun):
    for allwedd in range(len(alpha)):
        canlyniad = ''

        for llythyren in testun:
            if llythyren in alpha:
                rhif = alpha.find(llythyren)
                rhif -= allwedd
                if rhif < 0:
                    rhif += len(alpha)
                canlyniad += alpha[rhif]

            elif llythyren in bet:
                rhif = bet.find(llythyren)
                rhif -= allwedd
                if rhif < 0:
                    rhif += len(bet)
                canlyniad += bet[rhif]

        else:
            canlyniad += llythyren

    print('Allwedd Hacio #{}: {}'.format(allwedd,
canlyniad))
```

## Prif Ffwythiant

Tasg estynedig yw'r ffwythiant hon i greu dewislen unedig ar gyfer y rhaglen gyda'r tair ffwythiant yn hygrych.

Mae'r ffwythiant yn gofyn yn gyntaf am fewnbwn (1, 2 neu 3) i ddewis ffwythiant, yna mae'n gofyn am y wybodaeth sydd ei hangen i redeg pob ffwythiant.

**Nodyn:** Mae'r ffwythiannau amgryptio a dadgryptio yn cael eu rhedeg o'r tu mewn i'r gorchymyn print.

```
def main():
    print('Croeso i'r Seiffr Caesar!\n')
    rhaglen = input('Beth hoffech chi ei wneud? \n1.
Amgryptio \n2. Dadgryptio \n3. Hacio \n')

    if rhaglen == '1':
        testun = input('Rhowch y testun i amgryptio: ')
        allwedd = int(input('Rhowch yr allwedd: '))
        print('\nTestun Plaen: ' + testun + 'Allwedd
Amgryptio: ' + str(allwedd) + 'Testun amgryptiedig: ' +
amgryptio(testun,allwedd))

    elif rhaglen == '2':
        testun = input('Rhowch y testun wedi'i amgryptio
i ddadgryptio: ')
        allwedd = int(input('Rhowch yr allwedd: '))
        print('\nTestun amgryptiedig: ' + testun +
'Allwedd Amgryptio: ' + str(allwedd) + 'Testun Plaen: ' +
dadgryptio(testun, allwedd))
```

## Prif Ffwythiant

Dyma weddill ffwythiant y brif ddewislen.

Gan fod y ffwythiant hacio eisoes yn argraffu pob iteriad wrth iddo redeg, nid oes angen ei gynnwys o fewn gorchymyn argraffu.  
Cynhwysir gosodiad arall i sicrhau bod mewnbwn dilys yn cael ei roi, yna gelwir prif() fel bod y brif ddewislen yn dolennu am gyfnod amhenodol!

**Nodyn:** gelwir main() unwaith eto ar y diwedd, tu fas y ffwythiant, i gychwyn y rhaglen ac agor y ddewislen!

```
elif rhaglen == '3':  
    testun = input('Rhowch y testun wedi'i amgryptio i  
gracio: ')  
    print('\n')  
    hacio(testun)  
  
else:  
    print('RHAID I ROI OPSIWN DILYS RHWNG 1 A 3')  
  
main()  
  
main()
```

# technocamps



@Technocamps



Find us on  
Facebook