

technoteach[^] technocamps



UNDEB EWROPEAIDD
EUROPEAN UNION



Llywodraeth Cymru
Welsh Government

Cronfa Gymdeithasol Ewrop
European Social Fund



Prifysgol
Abertawe
Swansea
University



PRIFYSGOL
BANGOR
UNIVERSITY



Cardiff
Metropolitan
University

Prifysgol
Metropolitan
Caerdydd

it.wales



PRIFYSGOL
ABERYSTWYTH
UNIVERSITY

PRIFYSGOL
Glyndŵr
Wrecsam

Wrexham
glyndŵr
UNIVERSITY

University of
South Wales
Prifysgol
De Cymru

micro:bits

Python





Python

Yr Iaith Python

Yn y 1990au gynnar, dyluniodd Guido van Rossum yr Iaith Rhaglennu Python

Roedd ieithoedd arall wedi'i optimeiddio i ysgrifennu rhaglennu mawr gymhleth – ond roedd van Rossum eisiau iaith:

- Gyflym, i ysgrifennu rhaglennu bach
- Hawdd i newid rhaglennu presennol
- Syml, gyda chystawennau digymhleth
- Rhwydd i ddysgu



micro:bit Python

[Get started](#)[Projects](#)[Teach](#)[Let's code](#)

Let's code

[Share](#)

Quick links

New to coding or new to micro:bit

[{ } MakeCode editor](#)

Text-based programming, widely used in education

[Python editor](#)

>> NEW! <<

Manage whole class micro:bit coding sess

[micro:bit classroom](#)

micro:bit Python

The image shows the micro:bit Python IDE interface. On the left is a sidebar with a search bar and a list of categories: Variables, Display, Buttons, Loops, and Logic. The central area is the code editor, titled 'Untitled project', containing Python code for a micro:bit. The right side features a simulator of the micro:bit hardware with a red arrow pointing to the top of the board. Below the simulator is a 'Show serial' panel with a dropdown menu set to 'shake' and several sliders.

micro:bit

Variables
Keep track of data that...

Display
The micro:bit's LED...

Buttons
Use button inputs in you...

Loops
Count and repeat sets of...

Logic
Making decisions in code

Untitled project

```

1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     display.show(Image.HEART)
8     sleep(1000)
9     display.scroll('Hello')
10

```

Simulator

shake

Dewislen Cod

Ffenestr Cod

Cystrawennau Python

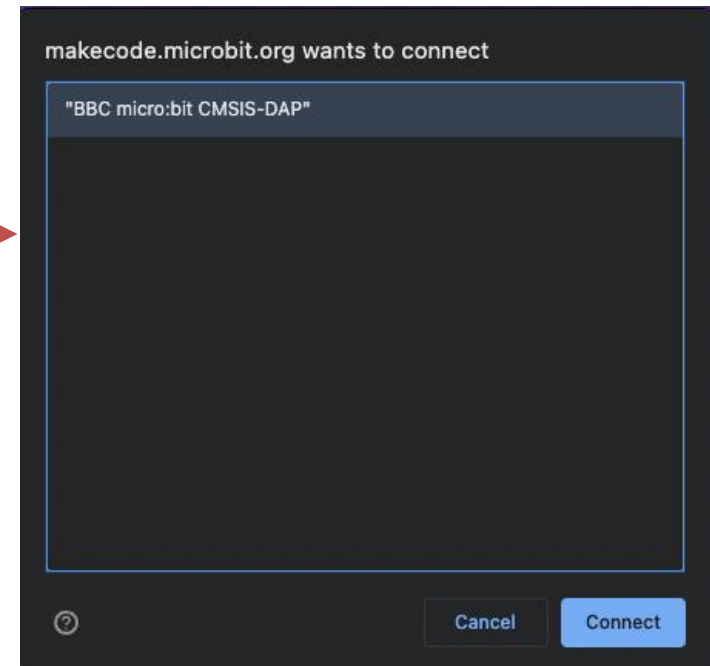
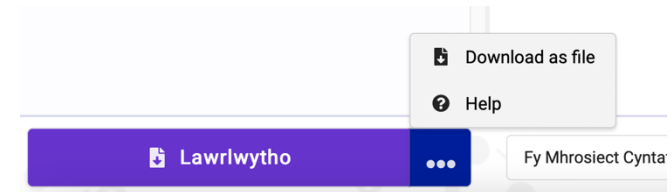
Beth sy'n digwydd os fydddech yn...

- Camsillafu gair: `primp("Hello World!")`
- Defnyddio'r prif llythrennau: `Print("Hello World!")`
- Anghofio dyfyniadau: `print(Hello World!)`
- Cam-gymharu dyfyniadau: `print("Hello World!")`
- Cam-gymharu cromfachau: `print('Hello')`

Ceisiwch weld pa wall-negesau sy'n cael eu taflu!

Cysylltu'r micro:bit

1. Plygiwch y micro:bit i'ch cyfrifiadur
2. Ar waelod chwith eich sgrin, cliciwch ar y 3 dot wrth ymyl 'Lawrlwytho', yna cliciwch ar 'Connect Device'
3. Dilynwch y cyfarwyddiadau ar y sgrin nes i chi weld y ffenestr naid hon
4. Cliciwch enw eich dyfais (dylai fod yr unig opsiwn)
5. Cliciwch **Connect**





Tasg:
Gwenu!

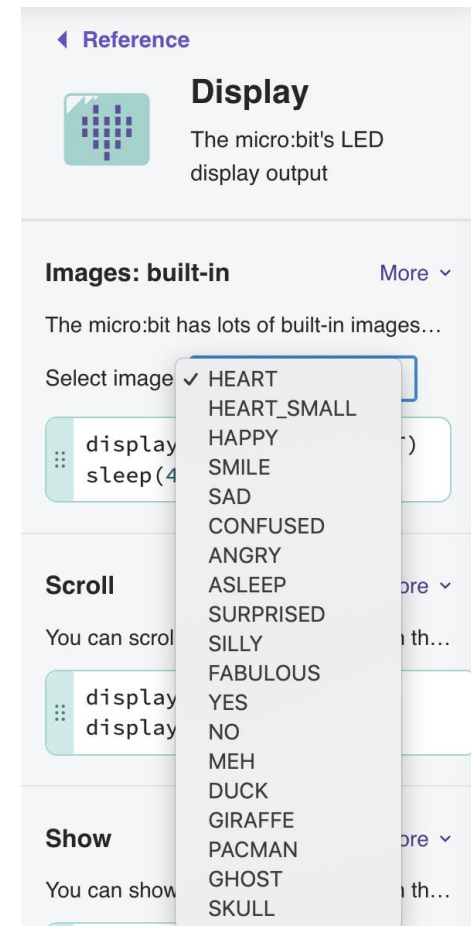
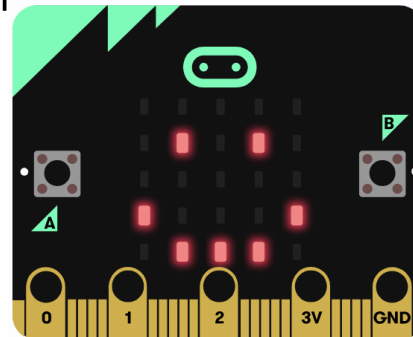
Gwenu!

Gadewch i ni ddechrau gan ddangos wyneb:

1. Dechrau trwy ddileu'r cod rhagosodedig, ond adael y llinell **"from micro:bit import *"**
2. Cliciwch ar yr adran **Display**
3. Dewiswch *Smile* o'r gwymplen **Select image**
4. Llusgwch y cod i'r ffenestr codio
5. Cliciwch ar "send to micro:bit"

Beth sy'n digwydd i'r micro:bit?

Ceisiwch newid y llun!



Creu Eicon

- Mae'n bosib i ni reoli yn gwmws pa LEDs i oleuo.
- Rydym ni gallu defnyddio hyn i greu eiconau ein hun
- Llusgwch y gorchymyn **Images: make your own** mewn i'ch cod
- Newidiwch y rhifau i greu eiconau eich hun. Mae'r rhif yn dynodi pa mor llachar mae'r golau am y LED yn y lleoliad yna
- Ceisiwch yr esiamplau isod a gweld pa batrymau arall gallwch chi greu

```
display.show(Image('99099:'  
                  '90999:'  
                  '09990:'  
                  '99909:'  
                  '99099'))
```

```
display.show(Image('90909:'  
                  '09990:'  
                  '99099:'  
                  '09990:'  
                  '90909'))
```

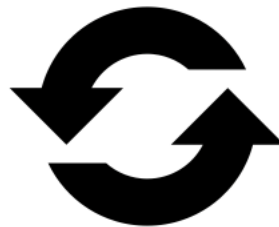
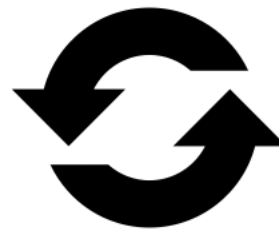
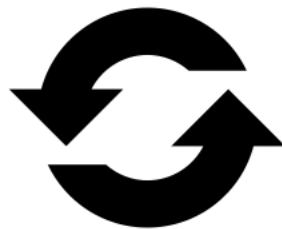
```
display.show(Image('90009:'  
                  '09090:'  
                  '90909:'  
                  '09090:'  
                  '00900'))
```



Tasg: Newid Wynebau

Dolennau

- Mae dolennau yn ein galluogi i ailadrodd gorchmynion.
- Gellir eu hailadrodd am byth, am nifer penodol o weithiau, neu am gyflwr penodol. Gelwir y broses hon yn Iteriad (Iteration).
- Mae'r gorchmynion hyn i'w cael yn yr adran **Dolennau**



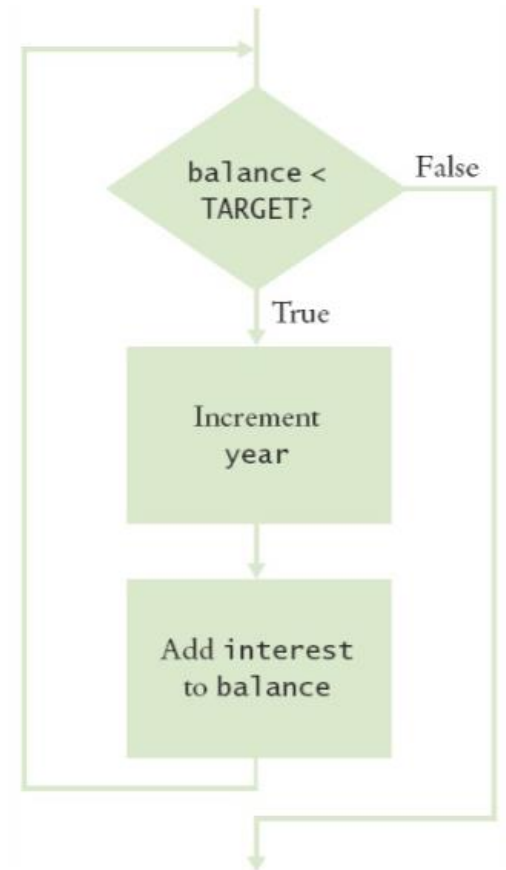
Y Dolen Ragosodedig

- Mae gennym orchymyn dolen ragosodedig pan fyddwn yn cychwyn y prosiect micro:bit - y ddolen '**while true**'
- Mae hyn yn gweithio fel dolen "am byth" ac yn rhedeg set o orchmynion nes bod y micro:bit wedi'i ddad-blygio neu ei ailosod.

```
while True:
    display.show(Image.HEART)
    sleep(1000)
    display.scroll('Hello')
```

Y Dolen While

- Mae dolen **while** yn ailadrodd cyfarwyddiadau tra bod yr amodau yn wir
- ```
while x < y:
 x = x + 1
```
- Mae hyn yn gwneud dolenni **while** yn ddefnyddiol iawn ar gyfer dilysu mewnbwn



# Newid Wynebau

- Rydym yn barod wedi creu rhaglen syml i ddangos eicon ar y micro:bit.
- Nawr gallwn ni ymestyn ein rhaglen i ddangos animeiddiad.
- Ychwanegu ail eicon ar ôl yr eicon gyntaf
- Beth sy'n digwydd ar ôl lawrlwytho'r cod i'ch micro:bit?

```
while True:
 display.show(Image.SMILE)
 sleep(1000)
 display.show(Image.SAD)
 sleep(1000)
```



# Creu Animeiddiad

- Hyd at hyn, rydym ni wedi defnyddio'r eiconau rhagosodedig i greu ein hanimeiddiadau.
- Ceisiwch ddefnyddio'r gorchymyn creu eicon o gynharach i greu animeiddiad eich hun

```
while True:
 display.show(Image('00300:'
 '03630:'
 '36963:'
 '03630:'
 '00300'))
 sleep(1000)
 display.show(Image('00000:'
 '00300:'
 '03630:'
 '00300:'
 '00000'))
 sleep(1000)
```



Tasg:  
Positifrwydd!

# Generadur Positifrwydd

- Dyfais sy'n dangos neges positif ar hap yw generadur positifrwydd.
- Mae'n gweithio yn debyg i bêl wyth hud – sy'n dangos neges ar hap wrth ysgwyd.
- I wneud hyn bydd rhaid defnyddio newidynnau i ddiffinio'r dyfyniadau.



# Newidynnau

- Mae newidynnau yn eitemau y gellir eu cofio a'u newid gan y micro:bit
- Mae newidyn medru cymryd ffurf wahanol, megis rhif neu destun
- Mae'n bosib newid werth y newidyn mewn sawl ffordd o fewn ein cod, ond i ddechrau mae rhaid i ni ddiffinio'r newidyn

# Creu Newidynnau

- I greu newidyn, dewis enw a pharu gyda gwerth gan ddefnyddio'r symbol '='
- Gadewch i ni ddechrau trwy greu newidyn am y dyfyniad. Byddwn yn osod y gwerth i fod yn 0.

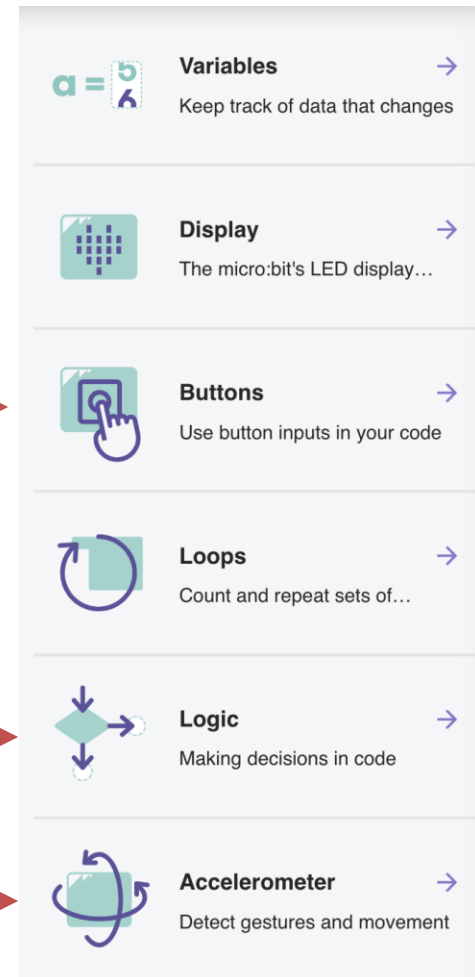
```
quote = 0
```

# Amodau

- Nodwedd o raglennu sy'n helpu ni dewis gweithredai gwahanol o dan amodau gwahanol.
- Mae rhaid i'r amodau yma rhoi'r canlyniad Gwir/Anwir.
- Esiampl:
  - Os ydych eich oedran > 17 yna:**  
**Gallwch chi yrru (ar ôl pasio eich prawf).**
  - Fel arall:**  
**Rhaid aros nes bod chi'n 17 i allu gyrru.**
- Mae'r amod uwch naill ai'n wir neu'n anwir, naill ai chi'n henach nag 17 neu ifancach nag 17.
- Beth sy'n digwydd os rydych chi'n 17 yn gwmws?

# Rhesymeg

- Yn y micro:bit Python Editor, mae rhan fwyaf o'r gorchmynion amodol wedi lleoli yn yr adran **Logic**
- Mae yna orchmynion amodol penodol yn yr adrannau **Buttons** ac **Accelerometer**



# Generadur Positifrwydd

Gadewch i ni greu generadur positifrwydd!

1. I ddechrau, llusgwch y gorchymyn **shake gesture** o'r adran **accelerometer** i'r ffenestr codio

Dileu'r gorchymyn **display.show()**

```
quote = 0
while True:
 if accelerometer.was_gesture('shake'):
```



# Generadur Positifrwydd

Rydym ni eisiau'r newidyn '**quote**' i gael ei osod i rif ar hap rhwng 1 a 5. Bydd hyn yn galluogi'r micro:bit dewis rhwng 5 dyfyniad.

1. Mewnbynnu'r modiwl **random**
2. Tu fewn i'r **if-statement** *shake*, defnyddiwch y gorchymyn **random.randint(1, 5)** i osod y newidyn i rif ar hap

```
import random
quote = 0
while True:
 if accelerometer.was_gesture('shake'):
 quote = random.randint(1, 5)
```

Beth ydych chi'n meddwl bydd y cam nesaf?

# Generadur Positifrwydd

Wrth ysgwyd y ddyfais bydd y newidyn **'quote'** yn cael ei osod i rif ar hap rhwng 1 a 5. Ond sut fydd y micro:bit yn gwybod beth i wneud â'r rhif yma?

4. Ychwanegwch **if-statement** i gyfarwyddo'r micro:bit i ddangos dyfyniad os mae'r newidyn **quote** yn hafal i 1

```
import random
quote = 0
while True:
 if accelerometer.was_gesture('shake'):
 quote = random.randint(1, 5)
 if quote == 1:
 display.show("You rock!")
```

5. Ailadrodd hyn am y gwerthoedd 2, 3, 4, a 5

# Generadur Positifrwydd

Dylech god edrych fel hyn:

```
import random
quote = 0
while True:
 if accelerometer.was_gesture('shake'):
 quote = random.randint(1, 5)
 if quote == 1:
 display.show("You rock!")
 elif quote == 2:
 display.show("You've got this!")
 elif quote == 3:
 display.show("Keep going!")
 elif quote == 4:
 display.show("Believe in yourself!")
 elif quote == 5:
 display.show("You are enough!")
```

Ceisiwch redeg eich cod! Ydy'r Generadur Positifrwydd yn gweithio? Allwch chi weld y broblem?

# Generadur Positifrwydd

Pan ddefnyddiwn ein generadur positifrwydd, mae'n arddangos yr un neges yn ailadrodd. Bydd yn fwy defnyddiol pe bai'n dangos y neges unwaith, ac yna gallem ei defnyddio eto i gynhyrchu neges wahanol

Sut ydych chi'n meddwl y gallwn ni wneud hyn? Edrychwch drwy'r adran **Display** am unrhyw orchmynion y gallem eu defnyddio

# Generadur Positifrwydd

Pan ddefnyddiwn ein generadur positifrwydd, mae'n arddangos yr un neges yn ailadrodd. Bydd yn fwy defnyddiol pe bai'n dangos y neges unwaith, ac yna gallem ei defnyddio eto i gynhyrchu neges wahanol

Sut ydych chi'n meddwl y gallwn ni wneud hyn? Edrychwch drwy'r adran **Display** am unrhyw orchmynion y gallem eu defnyddio

Bydd y gorchymyn **display.clear()** yn clirio sgrin y micro:bit unwaith y bydd y neges wedi'i harddangos. Ychwanegwch hwn i mewn i'ch **if-statements**

# Generadur Positifrwydd

Dylech god edrych  
fel hyn:

```
import random
quote = 0
while True:
 if accelerometer.was_gesture('shake'):
 quote = random.randint(1, 5)
 if quote == 1:
 display.show("You rock!")
 display.clear()
 elif quote == 2:
 display.show("You've got this!")
 display.clear()
 elif quote == 3:
 display.show("Keep going!")
 display.clear()
 elif quote == 4:
 display.show("Believe in yourself!")
 display.clear()
 elif quote == 5:
 display.show("You are enough!")
 display.clear()
```

# Tasg: Cod Morse



# Cyfathrebiad Radio

- Mae radio yn ffordd o gyfathrebu trwy donnau radio.
- Mae hyn yn golygu y gallwn anfon gwybodaeth o un lle i'r llall dros bellteroedd hir iawn.
- Allwch chi feddwl am rywbeth a allai ddefnyddio cyfathrebiad radio?



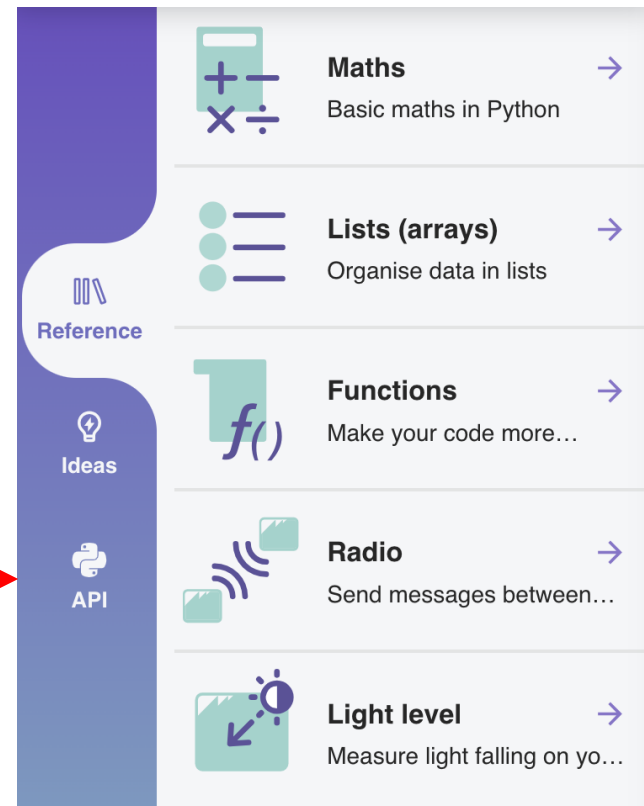


# Modiwlau (Llyfrgelloedd Python)

- Mae gan Python **llyfrgell safonol** o **fodiwlau** sy'n angenrheidiol i gynnwys ar unrhyw system.
- Mae'n bosib mewnbynnu'r **modiwlau** yma neu **ffwythiannau** penodol wrthynt.
- Mae hyn yn galluogi Python i wneud llawer fwy heb gynnwys llwyth o ffwythiannau diangen.
- Mae llyfrgell o fodiwlau micro:bit gan y Micro:Bit Python Editor – yn cynnwys y modiwl **Radio**

# Cyfathrebiad Radio

- Mae gan micro:bits nodwedd Bluetooth sy'n caniatáu iddynt gyfathrebu â'i gilydd.
- Gallwn ddefnyddio hwn i anfon negeseuon rhwng micro:bits.
- Mae'r blociau sydd eu hangen arnom ar gyfer hyn i'w gweld yn yr adran **Radio**.



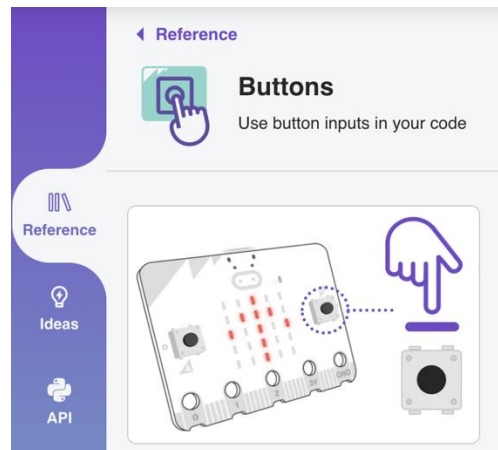
# Grwpiau Radio

- Cyn y gallwn anfon negeseuon at ein gilydd mae'n rhaid i ni osod y grŵp radio.
- Mae grwpiau radio fel sianeli - bydd unrhyw micro:bit sy'n defnyddio'r un grŵp yn derbyn y neges.
- Ceisiwch osod y ddau micro:bit i'r un grŵp radio - gall hyn fod yn unrhyw rif (hyd at 255), gwnewch yn siŵr eu bod yr un peth!
- Bydd rhaid i ni droi'r radio ymlaen i allu defnyddio. Mae'r gorchmynion yma i gael yn yr adran **Radio**.

```
import radio
radio.config(group=23)
radio.on()
```

# Digwyddiadau

- Mae gan bob micro:bit ddau fotwm, A a B.
- Mae'r botymau hyn yn ein helpu i ddewis pa gamau i'w cymryd heb ail-raglennu'r micro:bit bob tro.
- E.e. gallwn ddangos wyneb hapus pan fyddwn yn pwysu botwm A a wyneb trist pan fyddwn yn pwysu botwm B.
- Mae'r gorchmynion sydd eu hangen arnom i'w cael yn yr adran **Buttons**.



# Anfon...

- Gyda'r micro:bits yn yr un grŵp radio, gallwn ddechrau anfon negeseuon rhyngddynt.
- Byddwn yn defnyddio'r gorchymyn **radio.send()** yn yr adran **Radio** i wneud hyn
- Ceisiwch anfon y llinyn "dot" pan fydd botwm A wedi'i wasgu

```
while True:
 if button_a.was_pressed():
 radio.send('dot')
```

## ... a Derbyn

- Nawr bod un micro:bit wedi anfon neges, mae angen i ni ddweud wrth y micro:bit arall beth i'w wneud â'r neges a dderbyniodd.
- Llusgwch y cod **Receive a message** o'r ardal **Radio** mewn i'r ffenestr codio
- Newidiwch y gorchymyn **display.show()** i ddangos "dot ar y sgrin"

```
while True:
 message = radio.receive
 if message:
 display.show(Image('00000:'
 '00000:'
 '00900:'
 '00000:'
 '00000'))
```

- Rhwch gynnig ar y cod; ydy'r anfonydd yn anfon neges i'r derbynnnydd?

# Anfon Negeseuon Wahanol

Gallwn ddefnyddio'r botymau ar y micro:bit i ddewis gwahanol negeseuon i'w hanfon.

Ychwanegwch y cod am ail fotwm i anfon y llinyn "llinell".

Ceisiwch redeg y cod i weld beth sy'n digwydd.

Allwch chi weld y broblem?

Sut ydych chi'n meddwl y gallwn ni ddatrys hyn?

```
while True:
 if button_a.was_pressed():
 radio.send('dot')
 if button_b.was_pressed():
 radio.send('dash')
```

# Derbyn Negeseuon Wahanol

- Codio'r micro:bit i ddangos eiconau gwahanol wrth dderbyn negeseuon gwahanol.
- Ychwanegwch **if-statement** arall i'r cod
- Newid yr amodau i wirio os yw'r neges a dderbynnir yn "dot" neu "llinell"
- Dangos eicon "dot" neu "llinell"
- Ceisiwch redeg y cod i weld beth sy'n digwydd!

```
while True:
 message = radio.receive()
 if message == 'dot':
 display.show(Image('00000:'
 '00000:'
 '00900:'
 '00000:'
 '00000'))
 sleep(400)
 display.clear()
 if message == 'dash':
 display.show(Image('00000:'
 '00000:'
 '09990:'
 '00000:'
 '00000'))
 sleep(400)
 display.clear()
```



# Cod Morse micro:bit

Dylech cod terfynol edrych fel hyn:

Anfonydd:

```
from microbit import *
import radio
radio.config(group=23)
radio.on()
while True:
 if button_a.was_pressed():
 radio.send('dot')
 if button_b.was_pressed():
 radio.send('dash')
```

Derbynnydd:

```
from microbit import *
import radio
radio.on()
radio.config(group=23)
while True:
 message = radio.receive()
 if message == 'dot':
 display.show(Image('00000:'
 '00000:'
 '00900:'
 '00000:'
 '00000'))
 sleep(400)
 display.clear()
 if message == 'dash':
 display.show(Image('00000:'
 '00000:'
 '09990:'
 '00000:'
 '00000'))
 sleep(400)
 display.clear()
```

# Dotiau a Llinellau

- Ceisiwch anfon a chyfiethu negeseuon mewn Cod Morse!

|           |           |           |
|-----------|-----------|-----------|
| A ● -     | J ● - - - | S ● ● ●   |
| B - ● ● ● | K - ● -   | T -       |
| C - ● - ● | L ● - ● ● | U ● ● -   |
| D - ● ●   | M - -     | V ● ● ● - |
| E ●       | N - ●     | W ● - -   |
| F ● ● - ● | O - - -   | X - ● ● - |
| G - - ●   | P ● - - ● | Y - ● - - |
| H ● ● ● ● | Q - - ● - | Z - - ● ● |
| I ● ●     | R ● - ●   |           |



# Tasg: Melodi

# Nodyn Sengl

Gadewch i ni ddechrau trwy raglenni micro:bit i chwarae nodyn sengl!

1. Dileu'r cod rhagosodedig.
2. Ychwanegwch y gorchymyn **import music** i gael mynediad i'r modiwl cerddoriaeth.
3. Ychwanegwch y gorchymyn **music.play([])**.
4. Tu fewn y cromfachau, ychwanegu nodyn mewn dyfynodau.

Beth sy'n digwydd i'r micro:bit?  
Ceisiwch newid y nodyn.

```
import music

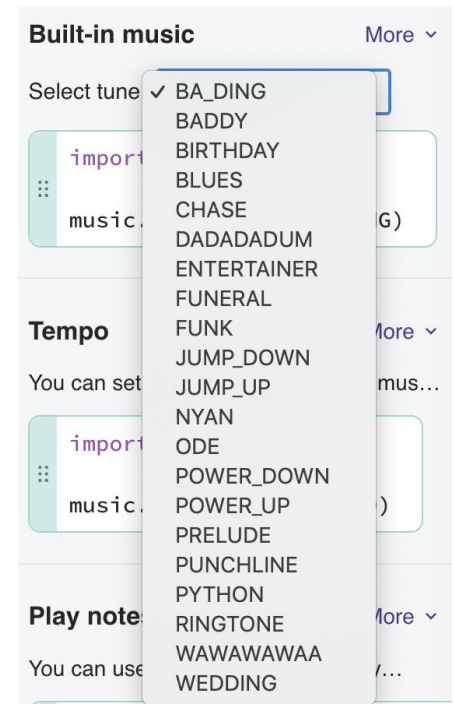
music.play(['c'])
```

# Melodi

- Mae gan MakeCode melodïau i ddefnyddio.
- Ceisiwch chwarae melodi trwy ychwanegu'r cod 'Built-in music' i'r ffenestr.
- Ceisiwch chwarae melodïau gwahanol!

```
import music

music.play(music.BA_DING)
```

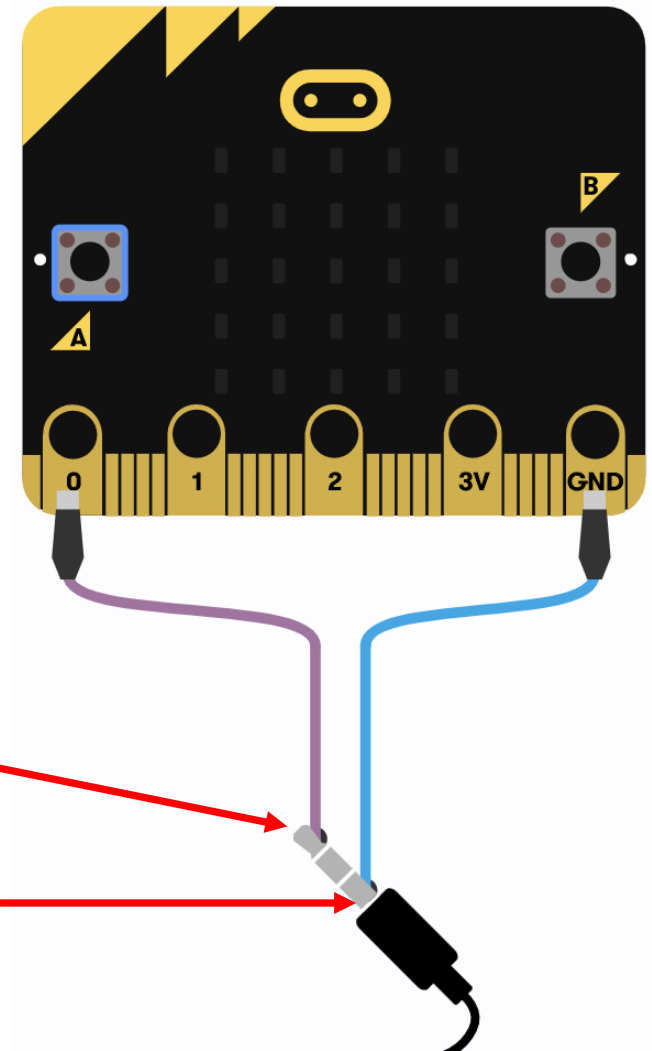


# Tasg: Clustffonau



# Cysylltu Clustffonau

- Gyda chymaint o micro:bits yn chwarae cerddoriaeth ar yr un pryd, mae'n galed i glywed unrhyw beth.
- Gallwn ni gysylltu clustffonau i'r pinnau ar waelod y micro:bits gyda chlipiau crocodeil.
- Cysylltwch bin 0 i flaen y plwg clustffon
- Cysylltwch y pin GND i ddiwedd y plwg clustffon



# Creu Cerddoriaeth

- Dysgom ni sut i greu nodyn a melodi gyda'r micro:bit
- Nawr byddwn yn ceisio creu cerddoriaeth ein hun
- Ychwanegu sawl nodyn, un ar ôl y llall, yn y gorchymyn **music.play([])**
- Bod yn greadigol a gweld pa felodïau medrwch chi greu!

```
import music

music.play(['c', 'd', 'e', 'c'])
```





# Tasg: Ailadrodd

# Melodi Hir

- Beth am chwarae melodi hir?
- Ceisiwch adeiladu a rhedeg y cod yma.

```
music.play(['c', 'd', 'e', 'c', 'c', 'd', 'e', 'c', 'e', 'f', 'g', 'e', 'f', 'g'])
```

# Melodi Hir

- Beth am chwarae melodi hir?
- Ceisiwch adeiladu a rhedeg y cod yma.

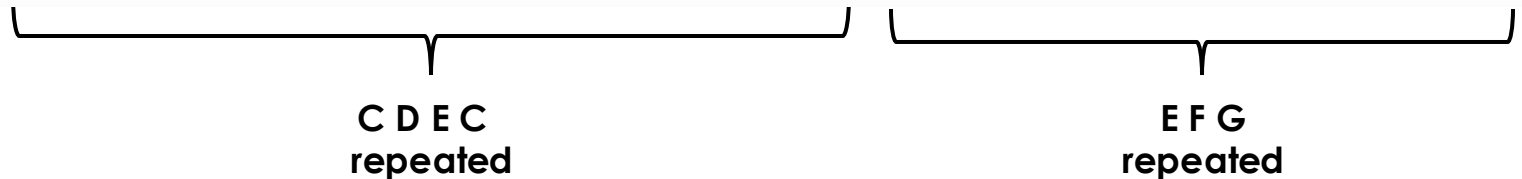
```
music.play(['c', 'd', 'e', 'c', 'c', 'd', 'e', 'c', 'e', 'f', 'g', 'e', 'f', 'g'])
```

- Nawr dychmygu sawl bloc bydd angen i chwarae can dwywaith, pum waith neu can gwaith yr hyd!
- Sut gallwn ni gwneud hyn heb ychwanegu bloc pob tro?

# Melodi Hir

- Mae'r melodi yma gyda nodau wedi'i ailadrodd
- Yn lle ychwanegu blociau newydd pob tro, mae'n bosib defnyddio dolennau i ailadrodd y rhan yma.

```
music.play(['c', 'd', 'e', 'c', 'c', 'd', 'e', 'c', 'e', 'f', 'g', 'e', 'f', 'g'])
```



# Y Dolen Rhagosodedig

- Mae gennym orchymyn dolen rhagosodedig pan fyddwn yn cychwyn y prosiect micro:bit - y ddolen **while true**
- Mae'r ddolen **while true** yn rhedeg set o orchmynion nes bod y micro:bit wedi'i ddatgysylltu neu ei ailosod.

```
while True:
```

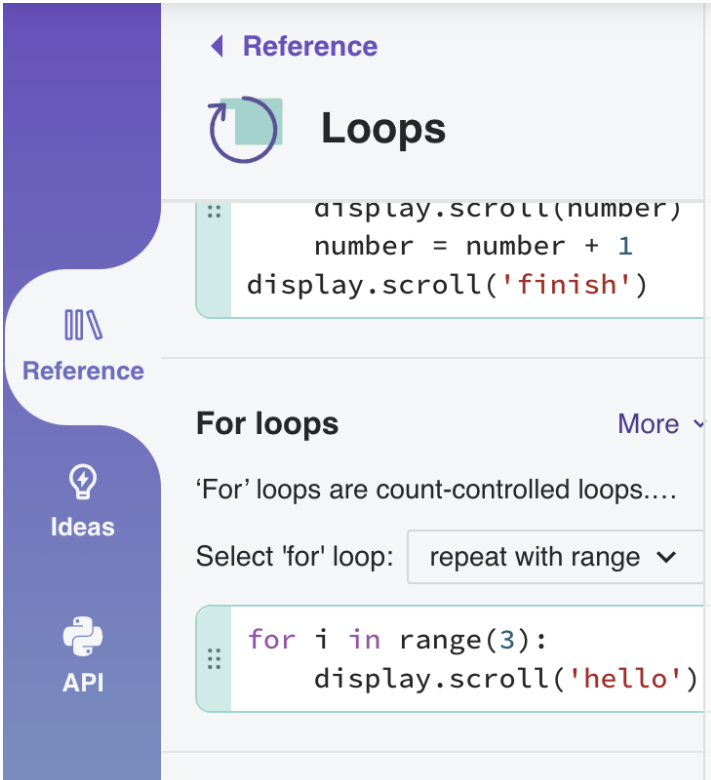
```
 display.show(Image.HEART)
```

```
 sleep(1000)
```

```
 display.scroll('Hello')
```

# Dolennau Penodol

- Mae yna ddolennau penodol sydd ddim yn ailadrodd am byth.
- Er enghraifft, gallwn ni defnyddio **for loop** i wneud darn o god ryw nifer o weithiau.
- Mae hyn yn ddefnyddiol i ailadrodd rhan o'r melodi.
- Defnyddiwch y ddolen i greu can!



The screenshot shows a web interface with a purple sidebar on the left containing icons for 'Reference', 'Ideas', and 'API'. The main content area has a 'Reference' header with a circular arrow icon and the title 'Loops'. Below this, there are two code snippets. The first snippet shows a loop that scrolls a display and increments a number until it reaches 'finish'. The second snippet shows a 'for' loop that scrolls a display with the word 'hello' three times. A 'More' link is visible next to the 'For loops' section header.

```
display.scroll(number)
number = number + 1
display.scroll('finish')
```

### For loops [More](#)

'For' loops are count-controlled loops....

Select 'for' loop:

```
for i in range(3):
 display.scroll('hello')
```

# Dolen Melodi

- Dechrau trwy lusgo'r **for loop** mewn i'ch cod a newid y **range** i 2.
- Dileu'r gorchymyn **display.scroll()** ac ychwanegu gorchymyn **music.play()**.
- Ychwanegu'r 4 nodyn gyntaf o'r melodi (C, D, E, C).

```
for i in range(2):
 music.play(['c', 'd', 'e', 'c'])
```

# Dolen Melodi

- Ychwanegu **for loop** arall gyda gweddill y nodau.
- Ceisiwch redeg y cod!



# Dolen Melodi

- Ychwanegu **for loop** arall gyda gweddill y nodau.
- Ceisiwch redeg y cod!
- Nawr mae'r cod yn daclus ac effeithlon. I ailadrodd fwy neu lai medrwn ni newid y rhif yn y ddolen yn unig!

```
for i in range(2):
 music.play(['c', 'd', 'e', 'c'])
for i in range(2):
 music.play(['e', 'f', 'g'])
```