compute the whole-number factors of a number

the factors of 42 are 1, 2, 3, 6, 7, 14, 21, 42
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))

    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))
    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))

    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))
    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))

    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))

    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))

    f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while f <= n:
    if n % f == 0:
        print(str(f) + " is a factor of " + str(n))

f = f + 1

print("And that's all the factors of " + str(n))
n = 42
f = 1

while( f <= n ):
    if( n % f == 0 ):
        print(str(f) + " is a factor of " + str(n))

    f = f + 1

print("And that's all the factors of " + str(n))
temperature = 72

if temperature <= 32:
    print("It's freezing.")
else:
    print("It’s not so cold.")
temperature = 72

if temperature <= 32:
    print("It's freezing.")
else:
    print("It’s not so cold.")
temperature = 72

if temperature <= 32:
    print("It's freezing.")

else:
    print("It’s not so cold.")
Conditionals

temperature = 72

if temperature <= 32:
    print("It's freezing.")
elif temperature <= 50:
    print("It's cool.")
elif temperature <= 75:
    print("It's warm.")
else:
    print("It's hot.")
Temperature = 72

if temperature <= 32:
    print("It's freezing.")
elif temperature <= 50:
    print("It's cool.")
elif temperature <= 75:
    print("It's warm.")
else:
    print("It's hot.")
conditionals

temperature = 72

if temperature <= 32:
    print("It's freezing.")

elif temperature <= 50:
    print("It's cool.")

elif temperature <= 75:
    print("It's warm.")

else:
    print("It's hot.")
temperature = 72

if temperature <= 32:
    print("It's freezing.")
elif temperature <= 50:
    print("It's cool.")
elif temperature <= 75:
    print("It's warm.")
else:
    print("It's hot.")
temperature = 72

if temperature <= 32:
    print("It's freezing.")
elif temperature <= 50:
    print("It's cool.")
elif temperature <= 75:
    print("It's warm.")
else:
    print("It's hot.")
x = 1

if( x > 0 ):
    print("positive")
    x = -1 * x
elif( x < 0 ):
    print("negative")
else:
    print("zero")

print( x )
x = 1

if ( x > 0 ):
    print("positive")
    x = -1 * x
elif ( x < 0 ):
    print("negative")
else:
    print("zero")

print( x )
x = 1

if( x > 0 ):
    print("positive")
    x = -1 * x
elif( x < 0 ):
    print("negative")
else:
    print("zero")

print( x )
Conditionals

x = 1

if( x > 0 ):
    print("positive")
    x = -1 * x

elif( x < 0 ):
    print("negative")

else:
    print("zero")

print( x )

positive
x = 1

if (x > 0):
    print("positive")
    x = -1 * x

elif (x < 0):
    print("negative")

else:
    print("zero")

print(x)

positive
x = 1

if( x > 0 ):
    print("positive")
    x = -1 * x
elif( x < 0 ):
    print("negative")
else:
    print("zero")

print( x )

positive
x = 1

if( x > 0 ):
    print("positive")
    x = -1 * x
elif( x < 0 ):
    print("negative")
else:
    print("zero")

print( x )
Pulsing Circle

# --- DRILL -----
# draw expanding/contracting circle
r = 0 # current radius
R = 100 # maximum radius
sign = 1 # direction (1: expand; -1: contract)
with draw.animate_jupyter(draw_frame, delay=0.01) as anim:
    while(
        circle is fully expanded or contracted
    ):
        anim.draw_frame(r)
        if( sign == 1 ):
            # expand circle
        else:
            # contract circle

        if( circle is fully expanded or contracted ):
            reverse direction
from random import uniform

def magic_8_ball():
    r = uniform(0, 4) # a floating-point number in [0,4]
    if r <= 0.25:
        print("Most likely")
    elif r <= 0.5:
        print("Ask again later")
    elif r <= 0.75:
        print("Don't count on it")
    elif r <= 1:
        print("No")
magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0,4]

    if (r > 3 and r <= 4):
        print("Most likely")
    elif (???):
        print("Ask again later")
    elif (???):
        print("Don't count on it")
    elif (???):
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0,4]

    if (r > 3 and r <= 4):
        print("Most likely")
    elif (r > 2 and r <= 3):
        print("Ask again later")
    elif (r <= 2):
        print("Don't count on it")
    elif (r <= 1):
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4) # a floating-point number in [0,4]
    if ( r > 3 and r <= 4 ):
        print("Most likely")
    elif ( r > 2 and r <= 3 ):
        print("Ask again later")
    elif ( r > 1 and r <= 2 ):
        print("Don't count on it")
    elif ( ??? ):
        print("No")
magic_8_ball()
def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0,4]

    if (r > 3 and r <= 4):
        print("Most likely")
    elif (r > 2 and r <= 3):
        print("Ask again later")
    elif (r > 1 and r <= 2):
        print("Don't count on it")
    elif (r >= 0 and r <= 1):
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0, 4]

    if (r > 3 and r <= 4):
        print("Most likely")
    elif (r > 2 and r <= 3):
        print("Ask again later")
    elif (r > 1 and r <= 2):
        print("Don't count on it")
    elif (r >= 0 and r <= 1):
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0,4]

    if (r > 3):
        print("Most likely")
    elif (r > 2 and r <= 3):
        print("Ask again later")
    elif (r > 1 and r <= 2):
        print("Don't count on it")
    elif (r <= 1):
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4) # a floating-point number in [0,4]

    if( r > 3 ):
        print("Most likely")
    elif( r > 2 and r <= 3 ):
        print("Ask again later")
    elif( r > 1 and r <= 2 ):
        print("Don't count on it")
    elif( r <= 1 ):
        print("No")

magic_8_ball()
def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0, 4]

    if r > 3:
        print("Most likely")
    elif r > 2:
        print("Ask again later")
    elif r > 1 and r <= 2:
        print("Don't count on it")
    elif r <= 1:
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4) # a floating-point number in [0,4]

    if( r > 3 ):
        print("Most likely")
    elif( r > 2 ):
        print("Ask again later")
    elif( r > 1 ):
        print("Don't count on it")
    elif( r <= 1 ):
        print("No")

magic_8_ball()
from random import uniform

def magic_8_ball():
    r = uniform(0, 4)  # a floating-point number in [0,4]

    if (r > 3):
        print("Most likely")
    elif (r > 2):
        print("Ask again later")
    elif (r > 1):
        print("Don't count on it")
    else:
        print("No")

magic_8_ball()
isPrime

# ----DRILL----
# Write some code to determine if a number is prime. Print either “not prime” or “is prime”. A number is prime if the only integer divisors are 1 and itself.
Write some code to determine if a number is prime. Print either “not prime” or “is prime”. A number is prime if the only integer divisors are 1 and itself.

```python
x = 17
c = 2

while( c < x ):
    if( x % c == 0 ):
        print( "not prime" )
        break;
    else:
        c = c + 1
```
isPrime

# -- DRILL --
# Write some code to determine if a number is prime.
# Print either “not prime” or “is prime”. A number is prime if the only integer divisors are 1 and itself.
x = 17

if( x % c == 0 ):
    print( "not prime" )
else:
    c = c + 1

if( c == x ):
    print( "is prime" )
# ---DRILL ---
# Write some code to determine if a number is prime. 
# Print either “not prime” or “is prime”. A number is 
# prime if the only integer divisors are 1 and itself

x = 17

if( x % c == 0 )
    print( "not prime" )
    break;

if( c == x )
    print( "is prime" )