Repetition

While-Loops

Exercise 1
What does this do?

```
i = 0
if i < 10:
    print(i)
    i = i + 1
```

Answer: The Processing program prints out the integers 0 through 9 on the console.

Exercise 2
What does this do?

```
screen_size = 220
min_square_size = 5

def setup():
    global screen_size
    size(screen_size,screen_size)

def draw():
    global screen_size
    global min_square_size
    square_size = 200
    while (square_size >= min_square_size):
        rect( 0,0,square_size,square_size )
        square_size = square_size/2
```

Answer: The Processing program draws a series of progressively smaller squares overlapping each other in the top-left corner of the canvas.

Exercise 3

Write the following **interactive** program.

First, define a variable `n_circles` with an initial value of 3.

Then, draw `n_circles` white circles (diameter 20) in a diagonal line.

if the user hits '+', increase the number of circles drawn.

if the user hits '-', decrease the number of circles drawn.

Hint: The formula for determining the ith circle's centre is:

$$(i * 20 + 10, i * 20 + 10)$$

Solution

```
# CMPT 140 - Repetition
# Topic(s): Counting While-Loops


n_circles = 3  # number of circles to draw

def setup():
    # black background to see changes more easily
    background(0,0,0)

def draw():
    global n_circles
    background(0,0,0)
    circle = 0
    # draw circles as a diagonal on screen
    while circle < n_circles:
        ellipse(circle*20+10,circle*20+10,20,20)
        circle = circle + 1

def keyPressed():
    global n_circles
    # draw one more circle
    if key == "+":
        n_circles = n_circles + 1
    # draw one less circle
    elif key == "-":
        n_circles = n_circles - 1
```

This code can be found in `cmpt140-ch12-py/cmpt140_ch12_while_loops/cmpt140_ch12_while_loops.pyde`

For-Loops

Exercise 4
Given the following pre-defined variables:

```
screen_size = 200
total_rows = 10
row_height = screen_size/total_rows
```

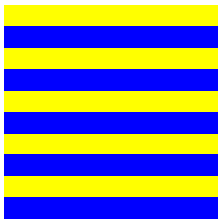
Write a program that displays alternating yellow and blue rows.

the number of rows is equal to `total_rows`

the height of each row is given by `row_height`

the first row is yellow

Note: Your program must work for **any value** of `total_rows`!



Solution

```
# CMPT 140 - Repetition
# Topic(s): For-Loops


screen_size = 200  # width, height of screen in px
total_rows = 10     # number of coloured rows
row_height = screen_size/total_rows  # height of a row in px

def setup():
    global screen_size
    # set screen size accordingly & yellow bg
    size(screen_size,screen_size)
    background(255,255,0)
    # don't draw shape borders for prettiness
    noStroke()

def draw():
    global screen_size
    global total_rows
    global row_height
    # alternating rows are blue (start with second row)
    fill(0,0,255)
    for row in range(row_height,screen_size,row_height*2):
        rect(0,row,screen_size,row_height)

    # alternate draw() definition
    # this solution loops over rows instead of screen pixels
    #fill(0,0,255)
    #for row in range(1,total_rows,2):
    #    rect(0,row*row_height,screen_size,row_height)
```

This code can be found in `cmpt140-ch12-py/cmpt140_ch12_for_loops/cmpt140_ch12_for_loops.pyde`

The alternate definition of `draw()` (as seen in the block of comments at the end of the function) reveals that it is acceptable to loop over the number of rows instead of loop over the blocks of pixels.

Infinite Loops

Demo 1

Beware infinite loops!

Solution

```
# CMPT 140 - Repetition
# Topic(s): Infinite Loops


screen_size = 200  # canvas width, height
max_squares = 6    # max number of squares to draw
square_size = 25   # square side in pixels

def setup():
    global screen_size
    size(screen_size,screen_size)
    noStroke()
    background(0,0,0)

def draw():
    global max_squares
    global square_size
    square = 0
    # draw diagonal line of white squares until we reach max number to draw
    while square < max_squares:
        rect(square*square_size,square*square_size, square_size,square_size)
        # uh oh! forgot to increment square, so 0 is always < max_squares!
        # uncomment the following line to increment our loop counter
        # square = square + 1
```

This code can be found in `cmpt140-ch12-py/cmpt140_ch12_infinite_loops/cmpt140_ch12_infinite_loops.pyde`