Variables and Expressions

Variables

Variables Giving a name to data values

Exercise 1
Which of the following are valid variable names? If they are invalid names, why are they invalid?

(a) `jeff`

Answer: Valid

(b) `def`

Answer: Invalid (can't use Python keywords)

(c) `hunter_name`

Answer: Valid

(d) `_is_ready`

Answer: Valid

(e) `Area9`

Answer: Valid

(f) `4_hunters_face_off_against_big_wyvern`

Answer: Invalid (starts with digit)

(g) `angry wyverns`

Answer: Invalid (contains a space)

(h) `uSeRnAmE`

Answer: Valid. Mixed case is allowed.

(i) `wyvern+hunter_fodder`

Answer: Invalid (contains special character `"+"`)

Exercise 2
Write a Processing program with the following behaviour:

When the user clicks the mouse: A **white circle** appears at the mouse's location. If the user clicks somewhere else, a white circle appears at that location and the old circle disappears.

When the user presses a key: The circle **turns black** but does not move.

Solution

```
# CMPT 140 - Data, Expressions, Variables
# Topic(s): Variable Assignment

# initialize circle coordinates so as to be off the screen
x = -100
y = -100


def setup():
    size(300, 300)
    fill(255)


def draw():
    global x
    global y
    background(210)
    ellipse(x, y, 30, 30)

def mouseClicked():
    global x
    global y
    x = mouseX
    y = mouseY
    fill(255)

def keyPressed():
    fill(0)
```

This code can be found in `cmpt140-ch07-py/cmpt140_ch07_var_circle/cmpt140_ch07_var_circle.pyde`

Expressions

Exercise 3
What are the values of these Python expressions?

(a) `3 + 2`

Answer: 5

(b) `3.0 + 2`

Answer: 5.0

(c) `7 / 2`

Answer: 3

(d) `7.0 / 2.0`

Answer: 3.5

(e) `7 / 2.0`

Answer: 3.5

(f) `11 % 5`

Answer: 1

(g) `12 % 3`

Answer: 0

(h) `3.0 + 3 / 2`

Answer: 4.0

(i) `(3.0 + 4) / 2`

Answer: 3.5

(j) `"c" + "o" * 6 + "kie"`

Answer: 'coooooookie'

Exercise 4
Write Python expressions for the following mathematical terms:

(a) $20 \bmod 3$

(b) $\frac{1}{2}(55.0)$

(c) $-(-3^3)$

(d) $(3 + \frac{35}{5})$

(e) $\frac{3.5+1.5}{11-6} - \frac{7}{(5-3)^8}$

Solution

```
# CMPT 140 - Data, Expressions, Variables
# Topic(s): Expressions


# Part (a)
20 % 3

# Part (b)
0.5 * 55.0

# Part (c)
-(-3 ** 3)

# Part (d)
(3 + 35/5)

# Part (e)
(3.5+1.5) / (11-6) - 7.0 / (5-3) ** 8
```

This code can be found in `cmpt140-ch07-py/cmpt140_ch07_arith_expressions/cmpt140_ch07_arith_expressions.pyde`

Exercise 5
Pretend that the Processing canvas represents a map of a city and that each pixel is a plot of land. When the user clicks the mouse, display the **price** and **property tax** of the plot of land that was clicked.

The **price** of a plot is `mouseX + mouseY`

The **tax** is 14% of the price

Use `text()` to display the price and tax on the canvas

Solution

```
# CMPT 140 - Data, Expressions, Variables
# Topic(s): Variable Assignment

def setup():
    background(200)
    size(300, 300)

def draw():
    return

def mouseClicked():
    """ Calculates land price based on mouse location.
    """
    background(200)
    price = mouseX + mouseY
    tax = price*0.14
    text("Land price: ", 230, 10)
    text(price, 230, 20)
    text("Property tax: ", 230, 30)
    text(tax, 230, 40)
```

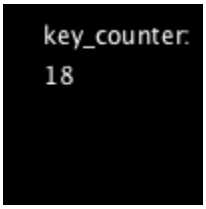
This code can be found in `cmpt140-ch07-py/cmpt140_ch07_var_assignment_1/cmpt140_ch07_var_assignment_1.pyde`

Variables in Processing

Exercise 6
Write an interactive Processing program which keeps track of the number of keys pressed using a variable. Display the **name of that variable** and its **value** on the canvas.

You'll need to use the keyword `global`

Use `text()` to display the information on the canvas



Solution

```
# CMPT 140 - Data, Expressions, Variables
# Topic(s): Variables in Processing


key_counter = 0  # tracks number of keys pressed since program start

def setup():
    """ draw black background """
    background(0,0,0)

def draw():
    """ draw current key press count against black background """
    background(0,0,0)

    global key_counter
    text("key_counter: ",20,20)
    text(key_counter,20,40)

def keyPressed():
    """ update key press counter """
    global key_counter
    key_counter = key_counter + 1
```

This code can be found in `cmpt140-ch07-py/cmpt140_ch07_vars_in_processing/cmpt140_ch07_vars_in_processing.pyde`