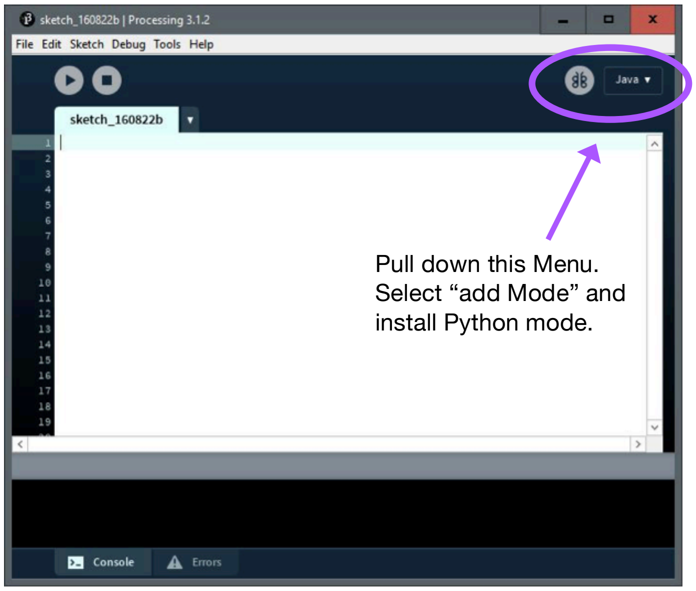
Creative Computing Cheat Sheet

1. Download Processing (no need to donate) for your platform from https://processing.org/download/
2. Unzip into a suitable location (Remember where it was saved!)
3. Locate the executable and click on it to run.
4. Install Python Mode:



sketch_160822b | Processing 3.1.2
File Edit Sketch Debug Tools Help
Java ▾
sketch_160822b
Console
Errors

Pull down this Menu. Select "add Mode" and install Python mode.

COMMONLY USED FUNCTIONS

`size(width, height)`

`size(500, 400)` creates a window 500 pixels wide by 400 pixels high.

`ellipse(x, y, width, height)`

`ellipse(100, 200, 60, 80)` draws an ellipse centered at position `(100, 200)` with a height of 60 pixels and a width of 80 pixels. To draw a circle, set width and height to the same value.

`rect(x, y, width, height)`

`rect(300, 100, 60, 90)` draws a rectangle with the upper left vertex at (300, 100) with a width of 60 pixels and a height of 90 pixels.

`line(x1, y1, x2, y2)`

`line(100, 200, 300, 400)` draws a line from point (100, 200) to (300, 400).

`background(r, g, b)`

`background(255, 0, 0)` sets the background of the window to red. r, g, b can be any number from 0-255. To find a specific colour code, check out Tools->Color Selector. Click on any colour that you like and note the r (red), g (green), b (blue) values.

`fill(r, g, b)`

`fill(50, 100, 50)` sets the colour for a shape to nice shade of green.

`stroke(r, g, b)`

Changes the outline of the shape to a different colour.

`strokeWeight(n)`

`n` is a number – changes the thickness of the outline of the shape.

COMMON FORMAT FOR RESPONSIVE SKETCHES

```
def setup():
    size(500, 500)

def draw():
    background(255, 255, 255)
    if mousePressed:
       #do something
```

PROCESSING GLOBAL VARIABLES

mousePressed	set to `True` if the user presses a mouse button
keyPressed	set to `True` if the user presses a key
width	the width of the canvas
height	the height of the canvas
mouseX, mouseY	the current position of the mouse

PYTHON SYNTAX

Arithmetic Operators

`+`	Add	`-`	Subtract	`*`	Multiply
`/`	Divide	`//`	Integer division (drops decimal)		
`**`	Exponent				

Comparison Operators

`==`	Equal to	`!=`	Not equal to
`>`	Greater than	`<`	Less than
`>=`	Greater than or equal to		
`<=`	Less than or equal to		

Boolean Operators - evaluate to True or False

`and` Example evaluating to True:

```
(1 > 0) and (4 > 0)
```

`or` Example evaluating to True:

```
(1 > 3) or (4 > 3)
```

`not` Example evaluating to True:

```
not (1 == 2)
```

Variable Assignment

```
x =
```

Conditionals

```
if x > 6 and y < 5 or p == 7:
  #do something
else:
  #do something else
```

Counted Loops

```
for i in range (1, 10):
      print (i)
```

This prints the values 1-9.

Conditional Loops

```
i = 1
while (i <= 10):
      print (i)
      i = i + 1
```

This also prints the values 1-9.