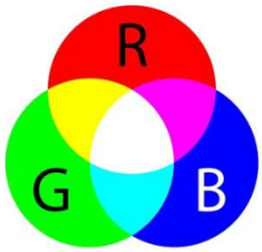
Splash 2023 - C15729: Creating Generative Art with Code
Reference Sheet

Detailed documentation for all functions can be found at https://processing.org/reference/

Section 1

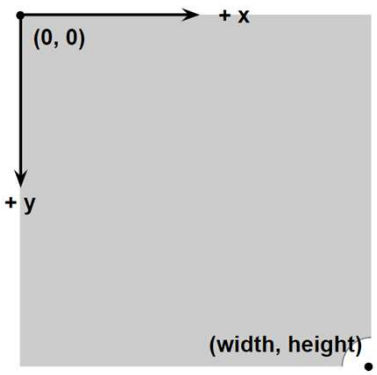
General
- `size(width, height)` – sets the drawing window **width** and **height**, run within `setup(){ }`

Drawing Tools
- `Background(r, g, b)` – sets RGB background color
- `stroke(r, g, b, alpha)` – sets RGB border color of subsequent shapes
  - **alpha** = 0 is clear, = 1 is opaque/solid
- `noStroke()` – no border for subsequent shapes
- `strokeWeight(px)` – border/line width in pixels (**px**)
- `fill(r, g, b, alpha)` – sets RGB fill color of subsequent shapes
- `noFill()` – no fill for subsequent shapes



Shape Primitives
- `circle(x, y, diameter)` – circle centered at (**x**, **y**) with set **diameter**
- `rect(x, y, width, height)` – rectangle with top left corner at (**x**, **y**) and set **width** and **height**
- `triangle(x1, y1, x2, y2, x3, y3)` – triangle with three vertices at (**x1**, **y1**), (**x2**, **y2**), (**x3**, **y3**)

Extras
- `line(x1, y1, x2, y2)` – draws a line between (**x1**, **y1**) and (**x2**, **y2**)
- `rotate(rad)` – rotates subsequent shapes by **rad** radians



Section 2

Randomness
- `random(high)` – randomly generates a float between 0 and high
- `random(low, high)` – randomly generates a float between low and high

Variables
- `float var = value` – stores a decimal (floating point) number
- `int var = value` – stores an integer

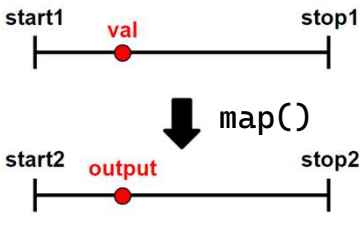
For Loops

General Expression

```
for (initialize; condition; update) {
  // repeated code
}
```

Example

```
for (int i; i < 10; i++) {
  // code repeated 10 times
}
```



Utilities
- `map(val, start1, stop1, start2, stop2)` – remap **val** from between **start1** and **stop1** to **start2** and **stop2**
- `println(variables)` – prints value of variables to console

Section 3

Perlin Noise
- `noise(x)` – generates 1D noise value at **x** between 0 and 1
- `noise(x, y)` – generates 2D noise value at (**x**, **y**) between 0 and 1

General If/Else Expression

```
if (condition1) {
  // runs in cond1 true
} else if (condition2) {
  // runs if cond2 true bit not cond1
} else {
  // runs if no conditions are met
}
```