v0.1

Processing.py / Python Mode Cheatsheet

Complete reference at
https://py.processing.org/reference

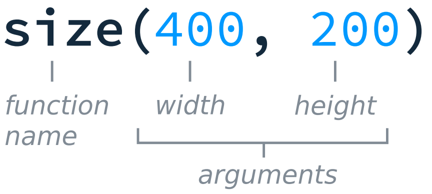
Sketch Structure

```
size(400, 200)
run this code once
```

Structuring an animated sketch:

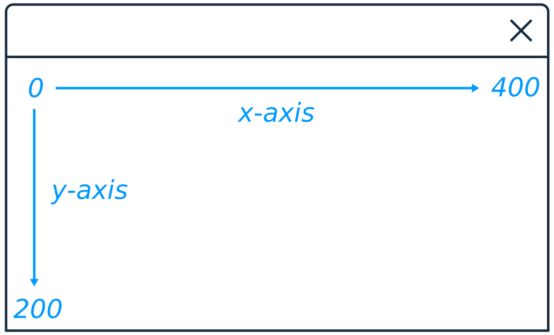
```
def setup():
    size(400, 200)
    run this code once at start
def draw():
    run this code every frame
```

Function anatomy:



```
size(400, 200)
```

Coordinate space:



Comments

```
# single line comment
'''
multiline comment
'''
```

Fills & Strokes

```
fill(color)
stroke(color)
background(color)
```

A red fill using three different color values:

```
fill('#FF0000')  # hexadecimal
fill(255, 0, 0)  # RGB
# HSB
colorMode(HSB, 360, 100, 100)
fill(0, 100, 100)
```

Arithmetic Operators

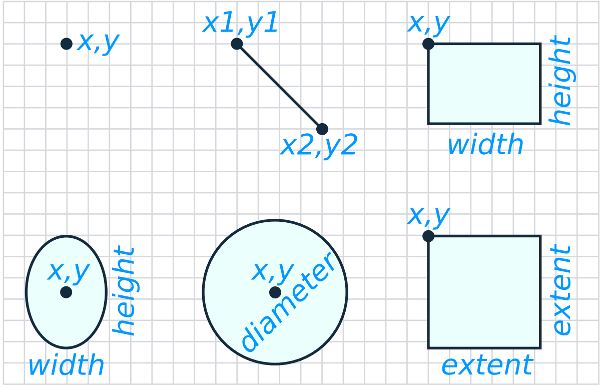
```
+ - * / %
```

Use brackets to override operator precedence:

```
(1 + 2) * 3  # equals 9, not 7
```

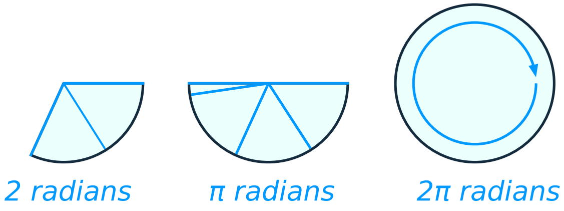
2D Primitives

```
point(x, y)
line(x1, y1, x2, y2)
rect(x, y, width, height)
ellipse(x, y, width, height)
circle(x, y, diameter)
square(x, y, extent)
```



```
arc(x, y, width, height, start, end)
```

For measuring arc angles, use radians:



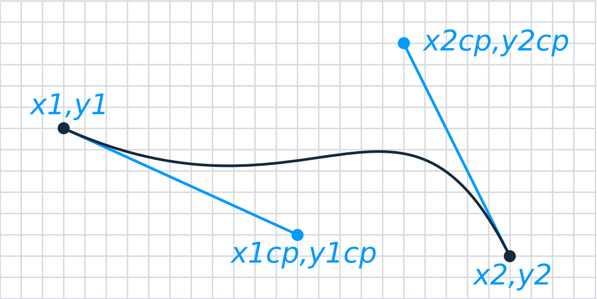
Shapes

Draw complex shapes using vertices nested within `beginShape()` and `endShape()` functions:

```
beginShape()
vertex(x1, y1)
vertex(x2, y2)
# add more vertices
endShape(CLOSE)
```

For curves, you use a Bézier vertex function:

```
beginShape()
vertex(x1, y1) # vertex 1
bezierVertex(
  x1cp, y1cp,  # control point 1
  x2cp, y2cp,  # control point 2
  x2, y2)      # vertex 2
endShape()
```



Typography

```
textFont(createFont(font, size))
textSize(render_size)
text(text, x, y)
```

Strings