Processing To C

translator with library

Table of Contents

Structure

() (parentheses) – work like in C++
, (comma)– work like in C++ (probably)
. (dot) – in most cases translated into **->**
/* */ (multiline comment) – work like in C++
/** */ (doc comment)
// (comment) – work like in C++
; (semicolon) – work like in C++, but for the end of class declaration should be added manually. Such modification work both in Processing and in C++
= (assign) – work like in C++
[] (array access) – work like in C++, because of library object **array**, **sarray**, **matrix**, **smatrix (**and others)
{} (curly braces) – work like in C++
catch – same syntax in Processing/Java/C++ but different exception names!

class – syntax and semantics are different. Translator always try to make as many translation as possible, but often manual changes are needed.

draw() – function with special meaning translated into **processing_window::draw()**
exit() – function with special meaning translated into **processing_window::exit()**
extends – translated into "**: public**"
false – work like in C++
final - translated into "**const**"
implements - translated into "**: public**" (redundancy with extend are removed)
import – sometimes translated into #include
loop() - library **function**
new – work like similar C++ because of implementation of Processing::ptr<T>, and sarray etc...
noLoop() - library **function**
null - - translated into "**nullptr**"
pop() - NOT IMPLEMENTED
popStyle() - NOT IMPLEMENTED

private public - syntax and meaning is different. Translator always try to make as many translation as possible, but often manual changes are needed (see → "class")

push() - NOT IMPLEMENTED
pushStyle() - NOT IMPLEMENTED
redraw() - library function
return – work like in C++
setup() – function with special meaning translated into **processing_window::setup()**
static – remain in code, bat may not work properly!
super – must be manually translated!
this – All ***this.*** are replaced by **this->**
thread() - there is different philosophy in C++, so NOT IMPLEMENTED!
true – work like in C++
try - same syntax in Processing/Java/C++ but different exception names!
void – work like in C++

Control

Relational Operators

!= (inequality) – work like in C++

< (less than) – work like in C++

<= (less than or equal to) – work like in C++

== (equality) – work like in C++

> (greater than) – work like in C++

>= (greater than or equal to) – work like in C++

Iteration

for while – work like in C++

Conditionals

?: (conditional) – work like in C++

break – work like in C++

case – work like in C++

continue – work like in C++

default – work like in C++

else – work like in C++

if – work like in C++

switch – work like in C++

Logical Operators

! (logical NOT) – work like in C++
&& (logical AND) – work like in C++
|| (logical OR) – work like in C++

Environment

cursor() - library **function. Only makes a cursor visible if already hidden**
delay() - library **function (imported from symshell)**
displayDensity() - always return 1
focused - NOT IMPLEMENTED! Confirms if a Processing program is "focused," meaning that it is active and will accept mouse or keyboard input. This variable is "true" if it is focused and "false" if not.
frameCount - library variable with **const**
frameRate() - translated into **setFrameRate()** library function
frameRate - library variable with **const**
fullScreen() - library **function**
height - library variable with **const**
noCursor() - library **function**
noSmooth() - library **function**
pixelDensity() - IGNORED
pixelHeight – same as height
pixelWidth -same as width
settings() - The **settings()** function is new with Processing 3.0. It's not needed in most sketches.
size() - library **function**
smooth() - library **function**
width - library variable with **const**

Event handling

Mouse

MouseButton

mouseClicked() - empty library **function for reiplementation by user**

mouseDragged() - NOT IMPLEMENTED

mouseMoved() - NOT IMPLEMENTED

mousePressed()

mousePressed

mouseReleased() - NOT IMPLEMENTED

mouseWheel() - NOT IMPLEMENTED

mouseX

mouseY

pmouseX

pmouseY

Keyboard

key

keyCode

keyPressed()

keyPressed

keyReleased() - NOT IMPLEMENTED

keyTyped()

Data

Primitive

boolean - translated into **bool**

byte - NOT IMPLEMENTED YET

char – work like in C++

color - NOT IMPLEMENTED YET

double – work like in C++

float – work like in C++

int – work like in C++

long – work like in C++

String - translated into **String, _param_string** library classes derived from **std::string**

Object – In Processing, like in JAVA, objects are instances of classes accessed by some kind of reference with counting (“.” operator, but managed heap with garbage collection is used). Similar meaning in C++ have **std::shared_ptr**s, but they have different interface. So we translate such references into **Processing::ptr<T>** templates opaquing **shared_ptr**s. It saves compatibility, but it is not very efficient. In many cases, especially as function parameters, such **ptr**s could be replaced with **Processing::ptr<T>&** or even **T&**. But this should be done manually and very carefully.

Composite

Array - translated into **array, matrix** library classes

ArrayList

DoubleDict

DoubleList

FloatDict

FloatList

HashMap

IntDict

IntList

JSONArray

JSONObject

LongDict

LongList

StringDict

StringList

Table

TableRow

XML

Conversion

binary()

boolean()

byte()

char()

float()

hex()

int()

str()

unbinary()

unhex()

String Functions

join()

match()

matchAll()

nf() nfc() nfp() nfs() - library **functions**

split()

splitTokens()

trim()

Array Functions

append()
arrayCopy()
concat()
expand()
reverse()
shorten()
sort()
splice()
subset()

Math

PVector

Operators

% (modulo) – work like in C++

* (multiply) – work like in C++

*= (multiply assign) – work like in C++

+ (addition) – work like in C++

++ (increment) – work like in C++

+= (add assign) – work like in C++

- (minus) – work like in C++

-- (decrement) – work like in C++

-= (subtract assign) – work like in C++

/ (divide) – work like in C++

/= (divide assign) – work like in C++

Bitwise Operators

& (bitwise AND) – work like in C++

<< (left shift) – work like in C++

>> (right shift) – work like in C++

| (bitwise OR) – work like in C++

Calculation

abs()

ceil() – work like in C++

constrain()

dist()

exp()

floor() – work like in C++

lerp()

log() – work like in C++

mag()

map()

max() – work like in C++

min() – work like in C++

norm()

pow() – work like in C++

round()

sq()

sqrt() – work like in C++

Trigonometry

acos() – work like in C++

asin() – work like in C++

atan() – work like in C++

atan2() – work like in C++

cos() – work like in C++

degrees()

radians()

sin() – work like in C++

tan() – work like in C++

Random

noise()
noiseDetail()
noiseSeed()
random()
randomGaussian()
randomSeed()

Constants

HALF_PI
PI
QUARTER_PI
TAU
TWO_PI

Color

Setting

background() - NOT IMPLEMENTED YET

clear() - NOT IMPLEMENTED YET

colorMode() - NOT IMPLEMENTED

fill() - library **function**

noFill() - library **function**

noStroke() - library **function**

stroke() - library **function**

Creating & Reading

alpha()
blue()
brightness()
color()
green()
hue()
lerpColor()
red()
saturation()

Shape

createShape()
loadShape()

PShape

Attributes

ellipseMode() - library **function**

rectMode() - library **function**

strokeCap() - library **function**

strokeJoin() - library **function**

strokeWeight() - library **function**

2D Primitives

arc() - library **function**

circle() - NOT IMPLEMENTED

ellipse() - library **function**

line() - library **function**

point() - library **function**

quad() - NOT IMPLEMENTED

rect() - library **function**

square() - NOT IMPLEMENTED

triangle() - NOT IMPLEMENTED

Curves

bezier()

bezierDetail()

bezierPoint()

bezierTangent()

curve()

curveDetail()

curvePoint()

curveTangent()

curveTightness()

3D Primitives

box() - 3D GRAPHIX NOT IN MY PLAN

sphere() - 3D GRAPHIX NOT IN MY PLAN

sphereDetail() - 3D GRAPHIX NOT IN MY PLAN

Vertex

beginContour()

beginShape()

bezierVertex()

curveVertex()

endContour()

endShape()

quadraticVertex()

vertex()

Loading & Displaying

shape()
shapeMode()

Input

Files

BufferedReader - NOT IMPLEMENTED

createInput() - NOT IMPLEMENTED

createReader() - NOT IMPLEMENTED

launch()

loadBytes()

loadJSONArray()

loadJSONObject()

loadStrings()

loadTable()

loadXML()

parseJSONArray()

parseJSONObject()

parseXML()

selectFolder()

selectInput()

Time & Date

day()
hour()
millis()
minute()
month()
second()
year()

Output

Text Area

print() - library **functions**

printArray()

println() - library **function**

Image

save()

saveFrame()

Files

beginRaw()
beginRecord()
createOutput()
createWriter()
endRaw()
endRecord()
PrintWriter
saveBytes()

saveJSONArray()
saveJSONObject()
saveStream()
saveStrings()
saveTable()
saveXML()
selectOutput()

Transform

applyMatrix() - 3D GRAPHIX NOT IN MY PLAN
popMatrix() - 3D GRAPHIX NOT IN MY PLAN
printMatrix() - 3D GRAPHIX NOT IN MY PLAN
pushMatrix() - 3D GRAPHIX NOT IN MY PLAN
resetMatrix() - 3D GRAPHIX NOT IN MY PLAN
rotate() - 3D GRAPHIX NOT IN MY PLAN
rotateX() - 3D GRAPHIX NOT IN MY PLAN
rotateY() - 3D GRAPHIX NOT IN MY PLAN
rotateZ() - 3D GRAPHIX NOT IN MY PLAN
scale() - 3D GRAPHIX NOT IN MY PLAN
shearX() - 3D GRAPHIX NOT IN MY PLAN
shearY() - 3D GRAPHIX NOT IN MY PLAN
translate() - 3D GRAPHIX NOT IN MY PLAN

Lights, Camera

Lights

ambientLight() - 3D GRAPHIX NOT IN MY PLAN

directionalLight() - 3D GRAPHIX NOT IN MY PLAN

lightFalloff() - 3D GRAPHIX NOT IN MY PLAN

lights() - 3D GRAPHIX NOT IN MY PLAN

lightSpecular() - 3D GRAPHIX NOT IN MY PLAN

noLights() - 3D GRAPHIX NOT IN MY PLAN

normal() - 3D GRAPHIX NOT IN MY PLAN

pointLight() s- 3D GRAPHIX NOT IN MY PLAN

potLight() - 3D GRAPHIX NOT IN MY PLAN

Camera

beginCamera() - 3D GRAPHIX NOT IN MY PLAN

camera() - 3D GRAPHIX NOT IN MY PLAN

endCamera() - 3D GRAPHIX NOT IN MY PLAN

frustum() - 3D GRAPHIX NOT IN MY PLAN

ortho() - 3D GRAPHIX NOT IN MY PLAN

perspective() - 3D GRAPHIX NOT IN MY PLAN

printCamera() - 3D GRAPHIX NOT IN MY PLAN

printProjection() - 3D GRAPHIX NOT IN MY PLAN

Coordinates

modelX()

modelY()

modelZ() - 3D GRAPHIX NOT IN MY PLAN

screenX()

screenY()

screenZ() - 3D GRAPHIX NOT IN MY PLAN

Material Properties

ambient() - 3D GRAPHIX NOT IN MY PLAN
emissive() - 3D GRAPHIX NOT IN MY PLAN
shininess() - 3D GRAPHIX NOT IN MY PLAN
specular() - 3D GRAPHIX NOT IN MY PLAN

Image

createImage()

PImage

Loading & Displaying

image()

imageMode()

loadImage()

noTint()

requestImage()

tint()

Textures

texture() - 3D GRAPHIX NOT IN MY PLAN

textureMode() - 3D GRAPHIX NOT IN MY PLAN

textureWrap() - 3D GRAPHIX NOT IN MY PLAN

Pixels

Blend() -
copy() -
filter() -
get()
loadPixels()
pixels[]
set()
updatePixels()

Rendering

blendMode() - 3D GRAPHIX NOT IN MY PLAN

clip() - 3D GRAPHIX NOT IN MY PLAN

createGraphics() - 3D GRAPHIX NOT IN MY PLAN

noClip() - 3D GRAPHIX NOT IN MY PLAN

PGraphics - 3D GRAPHIX NOT IN MY PLAN

Shaders

loadShader() - 3D GRAPHIX NOT IN MY PLAN
PShader - 3D GRAPHIX NOT IN MY PLAN
resetShader() - 3D GRAPHIX NOT IN MY PLAN
shader() - 3D GRAPHIX NOT IN MY PLAN

Typography

PFont

Loading & Displaying

createFont()

loadFont()

text()

textFont()

Attributes

textAlign()

textLeading()

textMode()

textSize()

textWidth()

Metrics

textAscent()
textDescent()