언어 종류

* C, C++, Java

Processing

* 비 전산 전공자들
* JAVA의 방언
* 자바와 동일하지만 사용자의 기능 추가

대화법

* Compile
  + 소스를 기계어로 변환
  + 전체를 번역하므로 번역 속도는 느리지만 실행 속도는 빠름
* Interpret
  + 소스코드를 기계어로 변환
  + 프로그램 한 줄씩 번역하고 실행, 변역 속도는 빠르지만 실행 속도 느림

거인들 (초기 컴퓨터 프로그램)

* Sketchpad
* Diana Book
* Harmony books

프로그램 실행 방식

* PDE(Processing Development Environment)
  + Run 버튼으로 실행

프로그래밍 방식

* 정적 static 모드
  + 간단하고 소규모 프로그램 작업 시 유용
  + 정해진 기능만 수행
* 액티브 Active 모드
  + 사용자와 interaction
  + 종료하기 전까지 사용자 입력을 받고 출력을 생성
  + setup(), draw() 함수를 필수적으로 사용

기본 도형 그리기

|  |  |  |
| --- | --- | --- |
| 점 | point(x, y); |  |
| 선 | line(x1, y1, x2, y2); |  |
| 사각형 | rect(x, y, w, h, tl, tr, br, bl); | 각 사이드의 둥글기 |
| 굵기 | strokeWeight(); | 선의 굵기 |
| 원 | ellipse(x, y, w, h); |  |
| 삼각형 | triangle(x1, y1, x2, y2, x3, y3); |  |
| 사변형 | quad(x1, y1, x2, y2, x3, y3, x4, y4); |  |
| 호 | arc(x, y, width, height, start, stop, mode); | HALF\_PI  PI  QUARTER\_PI  TAU = TWO\_PI의 다른 이름  Mode = OPEN,CHORD,PIE |
| 도 | radians(각도) |  |
| 부드러운 선 | smooth(); |  |
| 선 속성(도형) | strokeJoin( ); | 도형의 모서리(ROUND,BEVEL) |
| 선 속성(선) | strokeCap( ); | 선의 양끝(SQUARE,ROUND) |
| 배경화면 | background(); | rgb, 투명도  흑백, 투명도 |
| 도형의 색 | fill();  noFill(); | rgb, 투명도  흑백, 투명도 |
| 선의 색 | stroke();  noStroke(); | rgb, 투명도  흑백, 투명도 |
| 주석 | // 코드 | 메모로 사용  코드 무시 |
| 사각형 모드 | rectMode(); | CORNER=기본모드  CENTER=가운데 기준  CORNERS=왼쪽 위 오른쪽 아래  RADIUS= 가운데 점 반지름 |
| 원 모드 | ellipseMode(); | CORNER=기본모드  CENTER=가운데 기준  CORNERS=왼쪽 위 오른쪽 아래  RADIUS= 가운데 점 반지름 |
| 정수 | int | 비교적 작은 수 |
| 실수 | float | 비교적 큰 수 |
| True 혹은 False | boolean | 참 거짓 |
| 문자 열 | String | abc(묶음) |
| 문자 하나 | char | a,b,c |
| modulo | % | 5 % 4 = 1  30.0 % 33.0 = 30.0 |
| Increment | a++ | 1씩 증가 |
| Decrement | a-- | 1씩 감소 |
| 산술 연산자 | +=, -=, \*=, /= | int a= 10  int b =20  a += b  a += 20  a = 30 |
| 관계 연산자 | !=, <, >, <=, >= |  |
| 논리 연산자 | !, &&, || |  |
| 연산자 우선 순위 | 산술 > 관계 > 논리 |  |
| for 반복문 | for (처음값; 실행 범위; 변경 정도) {  statement;  } | (int x = a; x<=300; x+=30) |
| while 반복문 | 처음값;  while(실행 범위); {  statement; 변경 정도} | int x= a;  (x<=300)  x+=30; |
| if 조건문 | if (조건문) { statement;  } else if (조건문) {  statement;  } else {  statement;  } |  |
| switch 문 | char a = ‘n’;  switch(조건문) {  case 무엇:  statement;  default:  statement;  } | 조건문 = a, b, c  무엇 = ‘a’, ‘b’, ‘c’ |
| ? = conditional 문 | (test) ? 맞을 경우 할 일 : 틀릴 경우 할 일; | (x < 50) ? 0 : 255; |
| continue 문 | for (int x =0; x<300; x+=5) {  if (x=20) {  continue;  }  statement;  } | for문이나 while문 내부에서 아래 명령을 무시하고 다시 실행한다.  옆의 경우 x=20일때 statement를 실행 하지 않는다. |
| break 문 | int x =30;  while(x) {  case 30;  statement;  break;  case 20;  statement;  break; | switch, for, while 내부에서 반복문을 종료함 |
| translate 문 | translate(x, y); | 원점 좌표를 이동한다 |
| rotate 문 | rotate(각); | 좌표계를 회전 |
| scale 문 | scale(s);  scale(x,y); | 도형의 사이즈를 늘리거나 줄이거나 한다 |
| 변형문 저장 | statement //기본 좌표계  pushMatrix();  translate(x, y);  statement; //이동 후 좌표계  popMatrix()  statement; //기본 좌표계 | pushMatrix 전의 좌표계를 저장 후 popMatrix 사용 시 다시 가져온다. |
| 마우스 관련 변수 | mouseButton  mouseClicked()  mouseDragged()  mouseMoved()  mousePressed()  mouseWheel()  mouseX  mouseY  pmouseX  pmouseY | LEFT,RIGHT,CENTER  마우스가 클릭할 때  마우스를 drag할 때  마우스를 움직일 때  True or False  마우스를 돌렸을때+,-  현재의 x 좌표  현재의 y 좌표  이전 프레임의 x좌표  이전 프레임의 y좌표 |
| 거리 차이 사용법 | dist(x1, y1, x2, y2); |  |
| 점진적 변화 | float easing = 0.05  x += (mouseX – x) \* easing;  ellipse(x , 40, 12, 12); | 0 < easing < 1 |
| 마우스 거리 비율 | map(value, start1, stop1, start2, stop2); | 숫자를 특정한 범위로 변환하여 값을 바꾼다1 |
| 키보드 관련 변수 | key  keyCode  keyPressed()  keyPressed  keyReleased()  keyTyped() | 키보드 키 저장 CODE  UP,LEFT,ALT,SHIFT  입력된 키 값 저장  True or False  키 땠을때 누른 값  Ctrl, shift, alt 무시 |
| 프레임 속도 | frameRate(fps); | 초당 프레임 장수 조절. 기본 60 |