Code ↔ Nature

Processing @ UdK Raum 115

Part 5

April - June, 2016

Arrays

primitive data types (int, float, String) can hold just one value:

```
int x = 100;
float a = 3.3333;
```

But very often we need to work with collections of values. For that purpose we use the **composite** data types (Array, ArrayList, etc).

```
int[] x = new int[50]; // declare and initialize an array with 50 integers
x[0] = 66;             // assign a value to the first item
x[1] = 47;             // assign a value to the second item
println(x[0], x[1]);   // print the first two items of the array
```

`int[] x` means: x is an array of integers
`new int[50]` means: reserve space for 50 integers

Since x contains 50 integers, when assigning a value to x or reading a value from x we need to specify (inside square brackets) the index of the element we want (a number between 0 and 49).

Arrays: usage

computer memory

declare array "x" of type int

```
int[] x;
```

x = ? (uninitialized array)

initialize array x (set length)

```
x = new int[6];
```

x =

0	1	2	3	4	5
0	0	0	0	0	0

set third element

```
x[2] = 33;
```

x =

0	1	2	3	4	5
0	0	33	0	0	0

access third element

```
println(x[2]);
```

output to the console: 33

Arrays: falling circles example

```
// without arrays

float x0, y0, dy0;
float x1, y1, dy1;

void setup() {
  size(100, 400);
  x0 = random(width);
  y0 = 0;
  dy0 = random(10);
  x1 = random(width);
  y1 = 0;
  dy1 = random(10);
}
void draw() {
  background(255);
  ellipse(x0, y0, 20, 20);
  ellipse(x1, y1, 20, 20);
  y0 = y0 + dy0;
  y1 = y1 + dy1;
}
```

```
// with arrays

float[] x, y, dy;

void setup() {
  size(100, 400);
  x = new float[10];
  y = new float[10];
  dy = new float[10];
  for(int i=0; i<10; i++) {
    x[i] = random(width);
    y[i] = 0;
    dy[i] = random(10);
  }
}
void draw() {
  background(255);
  for(int i=0; i<10; i++) {
    ellipse(x[i], y[i], 20, 20);
    y[i] = y[i] + dy[i];
  }
}
```

Arrays: pulsating circles example

```
float[] x, y, speed, sz; // four arrays
int AMOUNT = 30;

void setup() {
  size(600, 600);
  x = new float[AMOUNT];
  y = new float[AMOUNT];
  speed = new float[AMOUNT];
  sz = new float[AMOUNT];

  for(int i=0; i<AMOUNT; i++) { // initialize all array items
    x[i] = random(width);
    y[i] = random(height);
    speed[i] = random(0.1);
    sz[i] = random(5, 30);
  }
}
void draw() {
  background(255);
  for(int i=0; i<AMOUNT; i++) {
    float thisSize = sz[i] * sin(frameCount * speed[i]);
    ellipse(x[i], y[i], thisSize, thisSize);
  }
}
```

References

Processing tutorial about Arrays:

https://processing.org/tutorials/arrays/

Fun Programming videos about Arrays:

54, 55, (56, 57, 58, 59), **78**

http://funprogramming.org/

Code ↔ Nature

April - June 2016 // UdK - Berlin

Abe Pazos
@fun_pro | fun@funprogramming.org
http://hamoid.com | http://funprogramming.org