Data Types, Variables, & Structures

Data Types

Processing can store and modify many different kinds of data, including numbers, letters, words, colors, image, fonts, and boolean values.

Primitive Data Types

Name	Size	Value Range
boolean	1 bit	true or false
byte	8 bits	0 - 255 numbers
char	16 bits	keyboard characters
int	32 bits	Integer numbers
float	32 bits	Real fractional numbers
color	32 bits	Red, Green, Blue, and Alpha
string	64 bits	Set of characters that form a word

Variables

A variable is a container for storing data. Variables allow a data element to be reused many times within a program.

// **Comments** are ignored by the computer but are important for people. They allow you to tell the story of the program and write notes for others.

```
int x;  // Declare the variable x of type int
float y; //Declare the variable y of type float
boolean b; //Declare the variable b of type boolean
x = 100;  //Assign the value 50 to x
y = 12.6  //Assign the value 5.4 to y
b = true;  //Assign the value true to b
```

A variable is first declared and then a value is assigned to it. When a variable is declared the computer allocates a portion memory to store the variable according to its size.

A variable can be declared and assigned separately or at the same time.

```
int x;
x = 100;
```

or

```
int x = 100;
```

Syntax

Variables cannot begin with numbers. By convention a variable name always begins with a lowercase letter.

Two widely used formats are Camel Case and Snake Case.

In Camel case the letter after a break is capitalized.

```
mouseY
camelCaseVariable
```

In Snake Case breaks are seperated with the `_` symbol.

```
snake_case_variable
computational_media
```

Processing Variables

```
width  //stores the width of the display window
height  //stores the height of the display window
mouseX //stores the mouse position on the x axis
mouseY //stores the mouse position on the y axis
```

Data Structures

A Data Structure is a collection of variables. Grouping variables together allows for easy access to related data.

Arrays

An array is an ordered set of data. It appears as a variable and can hold multiple values of the same data type at the same time. We can have arrays of booleans, integers, strings, etc. Array's are defined with the `[]` symbol.

```
String [] names = {"Michael", "Curry", "John", "Doe"};
float [] temperature = [88.9, 89.1, 89.0, 93.4, 95.2, 101.2];
int [] transactions = new int[50];
boolean [] isOff;
```

To access the elements of an array we use integers between the `[]` . Arrays start at 0, so it access the 1st element you use 0. The second element is at position 1.

```
names[0]
temperature[1]
```

Array Methods

- array.length: returns the number of elements in the array
- sort(array): sorts the elements in alphabetic order
- append(array,data): expands an array by one element and adds data value
- subset(array, offset, length): creates a subset of an array at offset for specified length
- expand(array, size): expands array by the size and retains the existing elements.

Functions and Animation Loop

A **function** is a set of code within a program that performs a specific task. They are also reusable and can be used over and over again. Functions are a powerful programming tool that make programs easier to read and change.

Processing has two main functions that form the backbone of the language: ***setup()*** *and* ***draw()***.

The code inside setup() runs once when the program first starts. The size of the display window is set along with intitalizing global variables used in the program.

The code inside draw() runs continuously in a loop and draws one image to the display window at the end of each loop.