Chapter 1

Introduction to Computers and Java

Side note: We will be using NetBeans as our IDE. Simply visit netbeans.org and download the latest version of the IDE.

Computer Basics

Computer systems consist of:

a) hardware – the tangible parts, the things you can touch, like the keyboard, monitor, mouse etc.

b) software – the intangible parts, the instructions that make computers do what they do so well. You can not touch the code itself, but you can definitely see its effects.

The CPU is the brain

Every computer has a CPU (Central Processing Unit or processor), that is the equivalent of the human brain. Well, in as much as being the computational component.

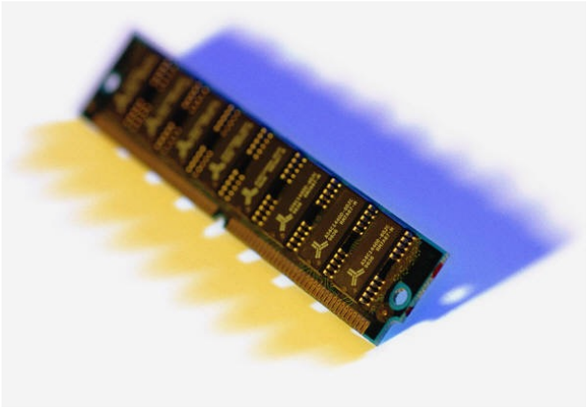
There are several families of cpu, usually manufactured by a single company. Known cpu families are the Intel, AMD, ARM and others.

Memory is the storage

Each computer uses memory to store both its instructions and data.

Primary memory, or RAM (Random Access Memory) is perhaps the most important component when it comes to the computer, since it directly communicates with the CPU, and acts as its storage facility.

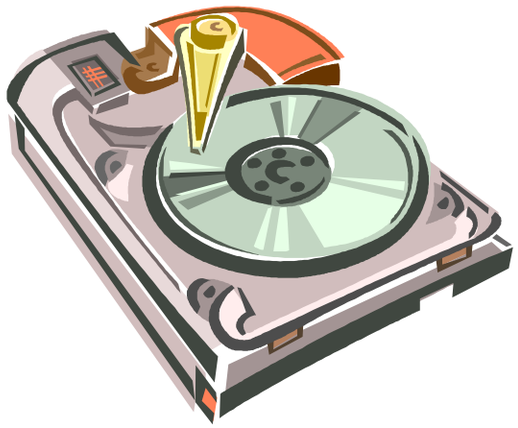
RAM is volatile



One characteristic of RAM is that it depends on the continued and uninterrupted supply of power, thus it is quite volatile. If the power was to be lost for whatever reason, everything in memory will be lost.

Although memory itself if volatile, realize that applications do make automatic backups of the documents they process, so it is not uncommon to recover a file after a power loss.

Secondary memory is permanent

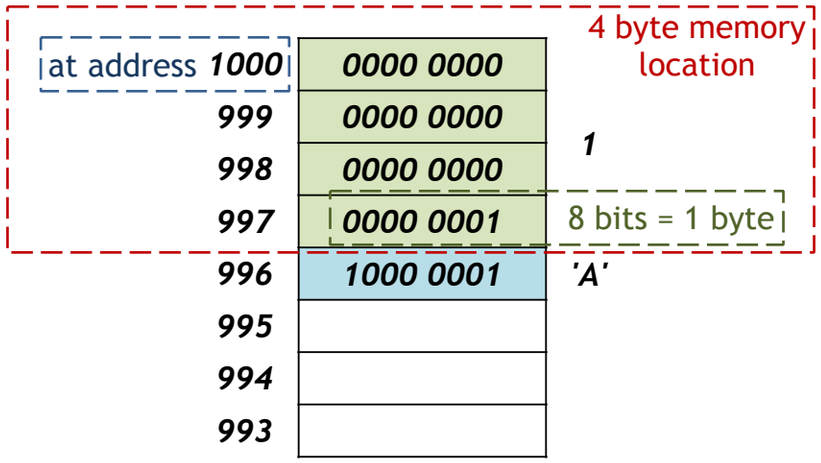


To provide for a more permanent storage facility, secondary memory is available in many forms and capacities.

The most common media is magnetic, such as hard disks, but flash memory is making some serious inroads and gaining popularity as an alternative.

The main point of secondary memory is that it retains its contents even after the power is turned off.

Memory is byte addressable

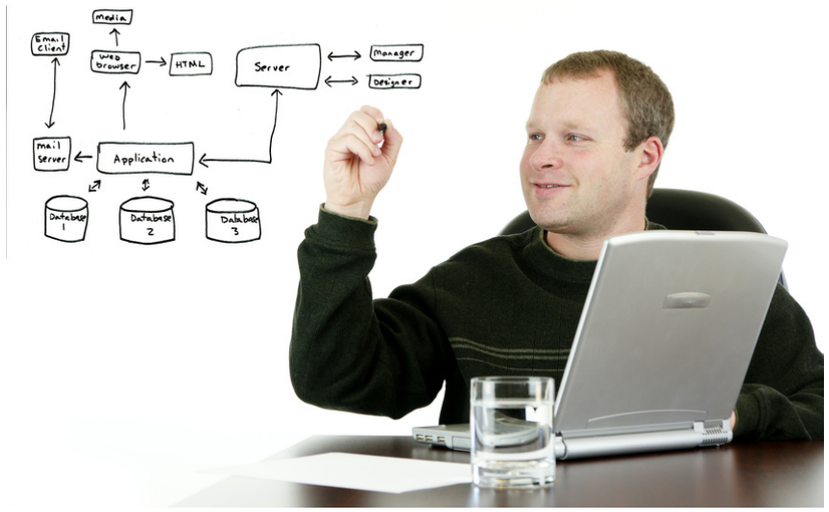


Memory is vast and used by many. Each memory location is 8 bits or 1 byte, and has a unique address, which enables the operating system to locate any piece of data and deliver it to the application requesting it.

All sorts of data can be stored, whether numeric, or text, and therefore variable amounts of bytes will be required to store such data. For instance, to store a single character, a 1 byte location would be needed, whereas to store an integer, 4 bytes would be needed.

The nice thing about this memory allocation, it is transparent to the programmer, and carried out by the operating system.

Programmers write programs



As a programmer you will be writing code, or programs. However, there is a significant amount of preparation that needs to be made before you start coding.

As you evolve as a programmer, you will learn the proper techniques and methodologies that will lead to properly designed programs.

Computers execute programs

Computers, the CPU to be exact, carries out the instructions that programmers write. This is the extent of the power of a computer, to execute exactly what it is told.

Therefore, the perceived power of computers is nothing other than the software written by programmers. The power lies with the human, not the machine. ☺

Programs must be translated

Java code

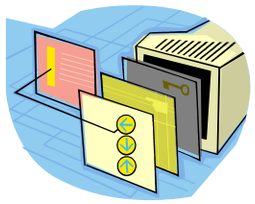
Machine code

Before a computer can execute your instructions, you must speak its language. As a programmer you will write in a high-level language, such as Java, but the computer (cpu) can not follow source code. The answer is to translate.

Translation basically converts source code to machine code, which the computer can then execute.

Using a Compiler
<Compile once – run many>

Source code
High level



Compiler

```
main:   subu    $sp, $sp, 32
        sw      $ra, 20($sp)
        sw      $fp, 16($sp)
        addiu   $fp, $sp, 28
        li      $v0, 4
        la      $a0, str
        syscall
        li      $a0, 10
        jal     fact
```

Assembly code
Low level

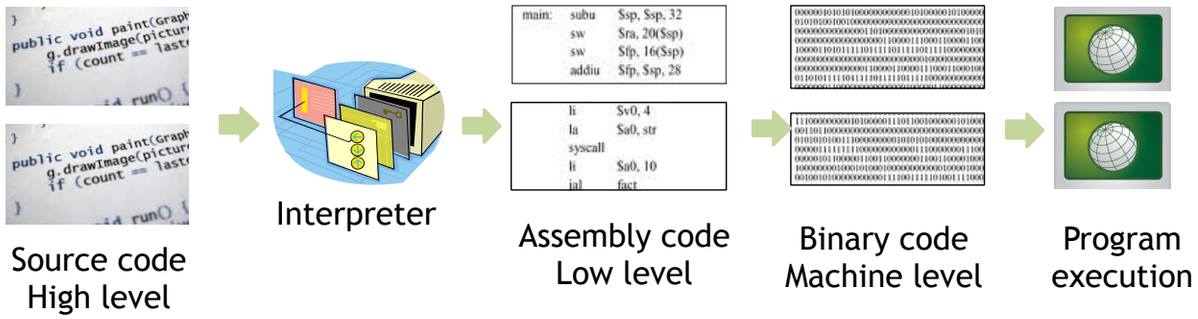
Binary code
Machine level

Translation can be accomplished using a compiler.

A compiler is a piece of software that converts the source code to the machine a family of cpu can understand. This makes this machine language very cpu dependent, albeit fast.

Compilation converts the source code to assembly, a low-level language that is more readable than machine level, and then to the actual machine code. Machine code is in binary code, consisting of 0s and 1s, called binary bits.

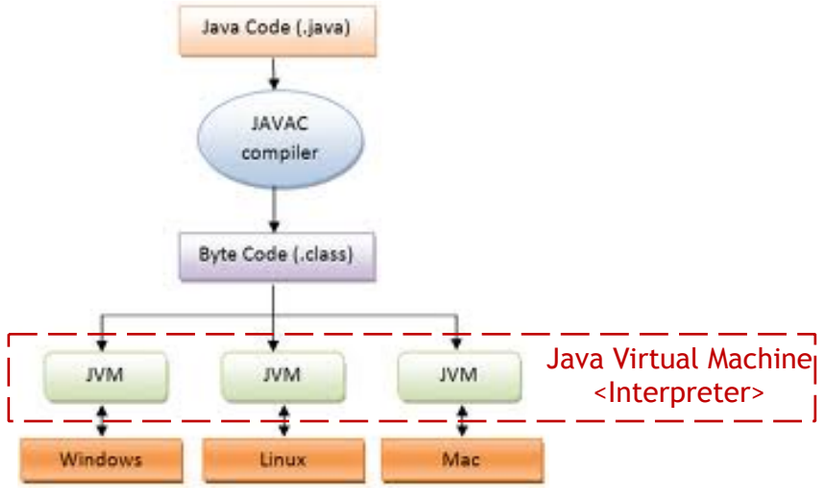
Using an Interpreter
<Compile-run cycle>



An interpreter is similar to a compiler in that it translates the source code to machine code, but it also executes the code as well.

This translation and execution is done in steps, each step translating and executing a portion of the program. This iterative process is a bit slower than compiled code however.

The Java way uses both techniques



The Java way is to combine the two techniques. A compiler first translates the source code to an intermediate language called bytecode.

The bytecode is a language that a virtual machine can understand and thus does not depend on any family of cpu. This provides the great benefit of portability of java code.

The bytecode is then interpreted by the JVM (Java Virtual Machine), code that resides on each machine, to the machine specific code. This interpretation is also a translation step, however it is a lot simpler than true compilation.

Thus, java offers great portability with just a small and most often unnoticed, penalty in execution time.

A class loader combines all the bytecode files

Since real application are made up of many components, called classes, a java program may consist of a number of bytecode files. All these files must be combined together into a single executable, and that is where the loader comes in.

A loader in java is analogous to a linker in other languages.

A Sip of Java

Java and Electronics?

1991 – Language for controlling electronic devices.

Project fizzled

Java and the Internet?

1994 – HotJava browser developed

1995 – Netscape adds Java support

Java is known as an Internet language, meaning a good choice for developing internet based applications.

Java is also a very powerful language for desktop development, and is used equally successfully in such environments.

Java application vs. Java Applet

Applications
run on PC

Applets
run in browser

A java applet is intended to be delivered to another computer for execution. This happens inside your java enabled browser.

A java application is your traditional application and runs under your operating system, whichever that is.

Application Deconstructed <FirstProgram.java>

```
import java.util.Scanner;
public class FirstProgram
{
   public static void main(String[] args)
   {
      System.out.println("Please enter two numbers and");
      System.out.println("I will compute their sum.");
      int n1;
      int n2;
      Scanner keyboard = new Scanner(System.in);
      n1 = keyboard.nextInt();
      n2 = keyboard.nextInt();
      System.out.println("Their sum is");
      System.out.println(n1 + n2);
   }// end main()
}// end FirstProgram
```

Like to use the scanner class

Inform the user

Declare variables

Let's read from the keyboard

Give user an answer

Application Deconstructed <output>

```
Please enter two numbers and
I will compute their sum.
10 30
Their sum is
40
```

Programming Basics

Java is OO

Object oriented programming is the buzz word these days, although it has existed for a long time. Although used at first for simulation purposes, it is the choice of programming current applications.

Objects can represent many things

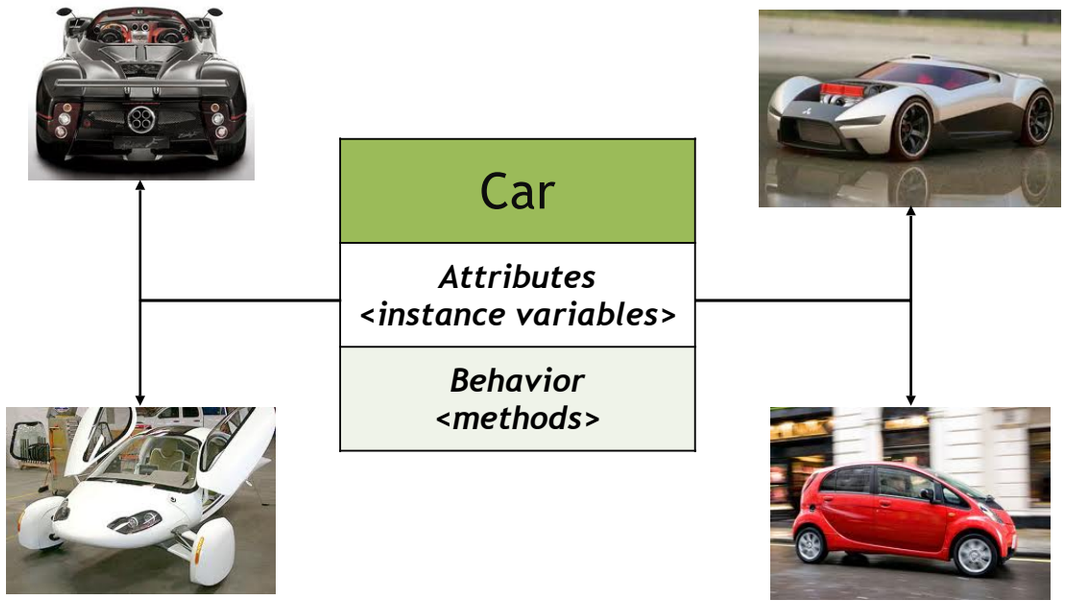
The basic idea of OOP revolves around the object. An object is a programmatic entity that can truly represent anything. Objects can model humans, aliens, inanimate objects, processes and whatever else you can think of.

Objects offer a means to store state, in terms of variables called instance variables, and behavior, through methods.

Objects interact with each other in many ways and communicate by calling each others' methods.

Each object represents an instance of a class, which is the blueprint of all objects of that class. Although you may create a number of objects from the same class, each object is different from all the others due to its state. A human object may have red hair, while another black. So even both are objects, they have different states.

A class provides the blueprint



In OOP you design your classes with objects in mind. A class recall is the blueprint of what an object will look like, but you cannot have an object lest you have a class first.

A class is instantiate as many times as needed to create all the objects that are used in an application. Instantiating means simply to create an instance of the class, i.e. an object.

OOP is based on three pillars

OOP languages must support three features if they are to be called object oriented. These are, encapsulation, inheritance and polymorphism.

Encapsulation = Information Hiding

You can drive a car, but you may not know
how an engine works.

Inheritance = Reuse / Extend

Objects can be reused

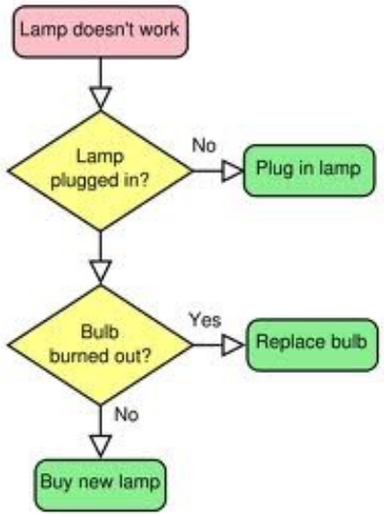
Objects can be extended

Polymorphism = React in own way

Each object behaves differently to the call to come back. Although all objects understand the command, they all react to it differently.

For instance, the rat may walk, the bird fly, the child run and the snake slither.

Before coding begins develop your algorithms



Algorithms come in many flavors

An algorithm is just the process of putting down your solution in a way that leads to a solution and anyone translating it will derive the same solution. In other words, the steps are clear and not ambiguous.

Once coding begins be aware of bugs

Syntax – Grammar not followed

Run-time – Compiler can't honor request

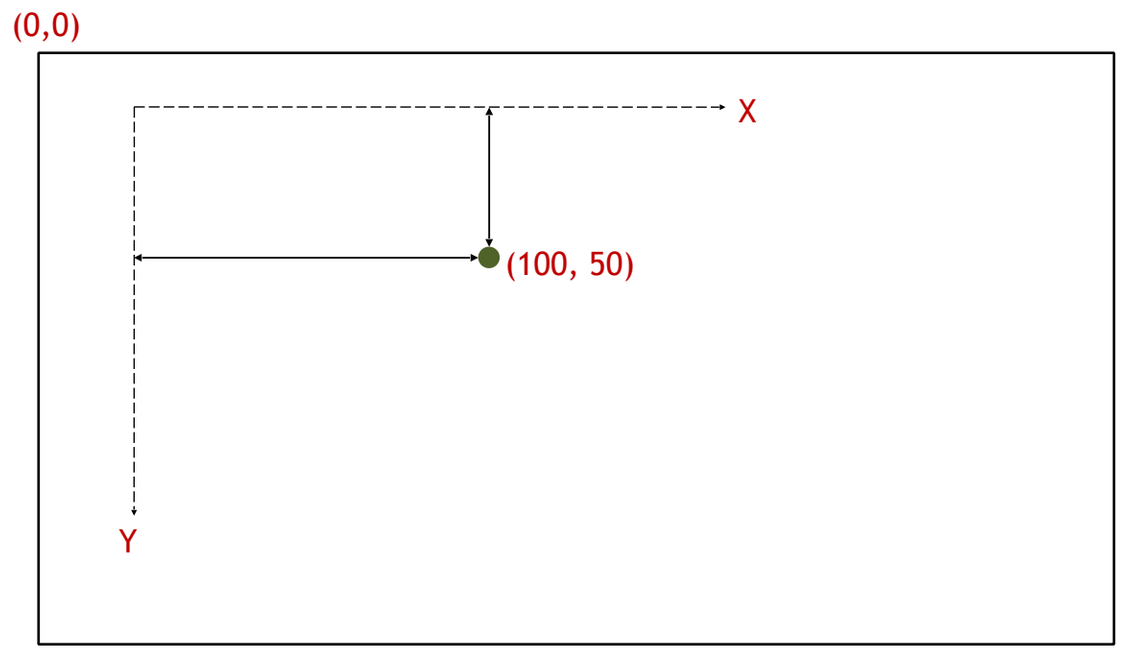
Logic – Programmer slip

Bugs are a way of life for programmers. You learn how to deal with them as you become more experienced in coding, but they could be the source of hours of frustration.

The errors above are listed from the easy to fix to hard to fix. We all make mistakes, but the true sign of a good programmer is how easy they can detect and fix them.

Graphics Supplement

The Coordinate System

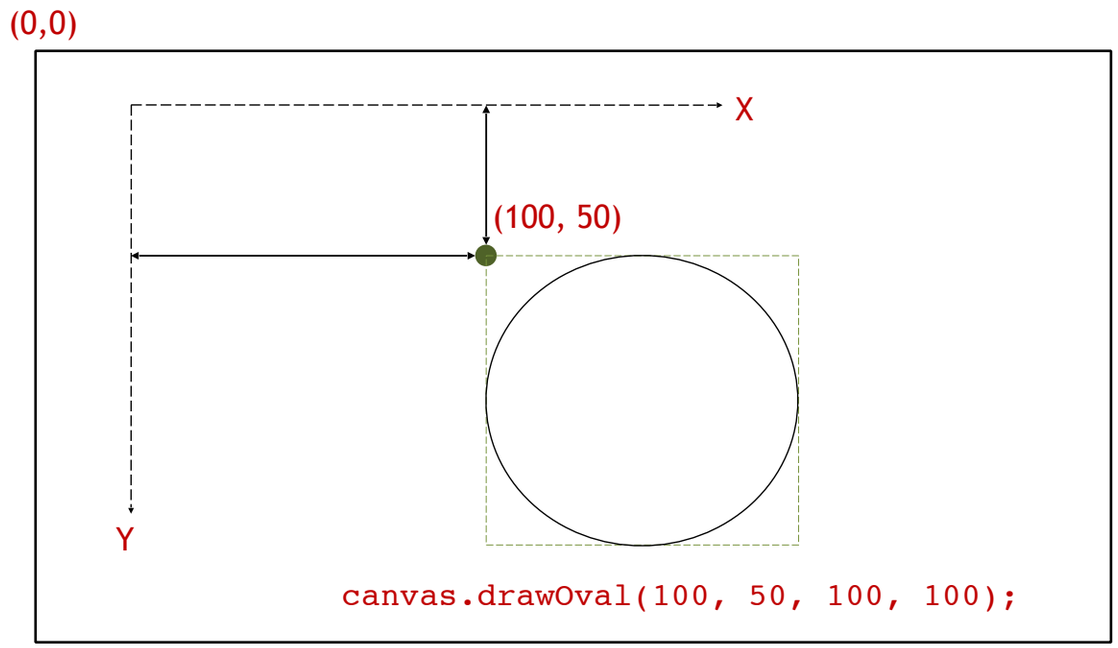


The coordinate system used for drawing graphics is one with the origin placed at the top left corner of the screen. This is a bit unconventional for most, but you do get used to it quickly.

The positive x-axis is to the right of the screen while the positive y-axis is toward the bottom of the screen.

The current screen resolution will dictate the x,y coordinates of the bottom right corner.

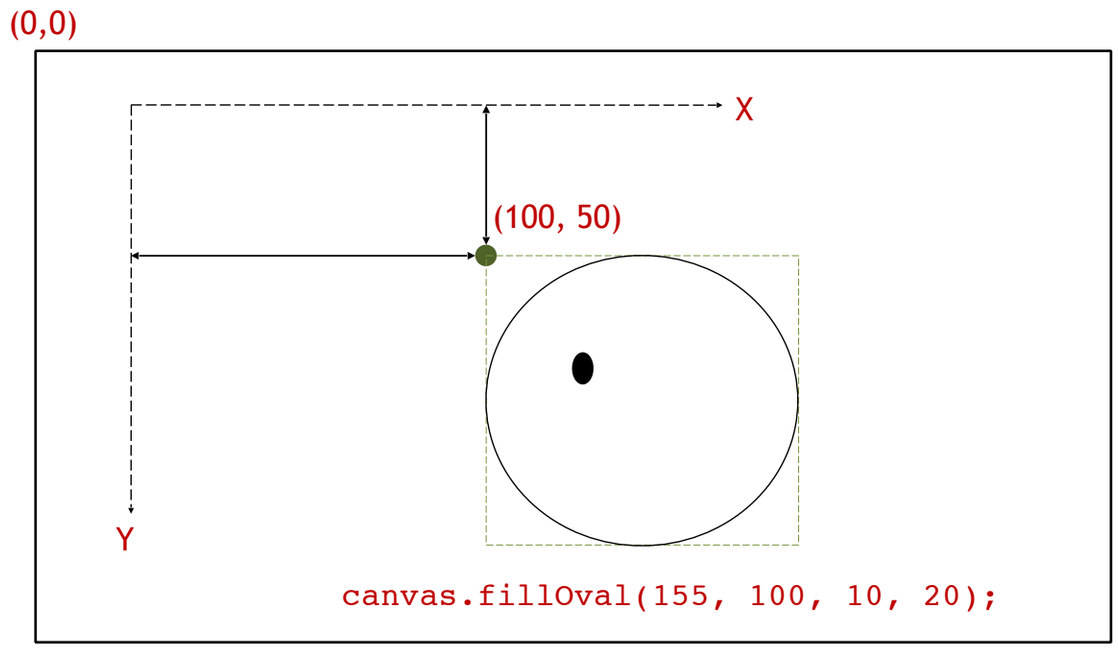
Drawing an Oval



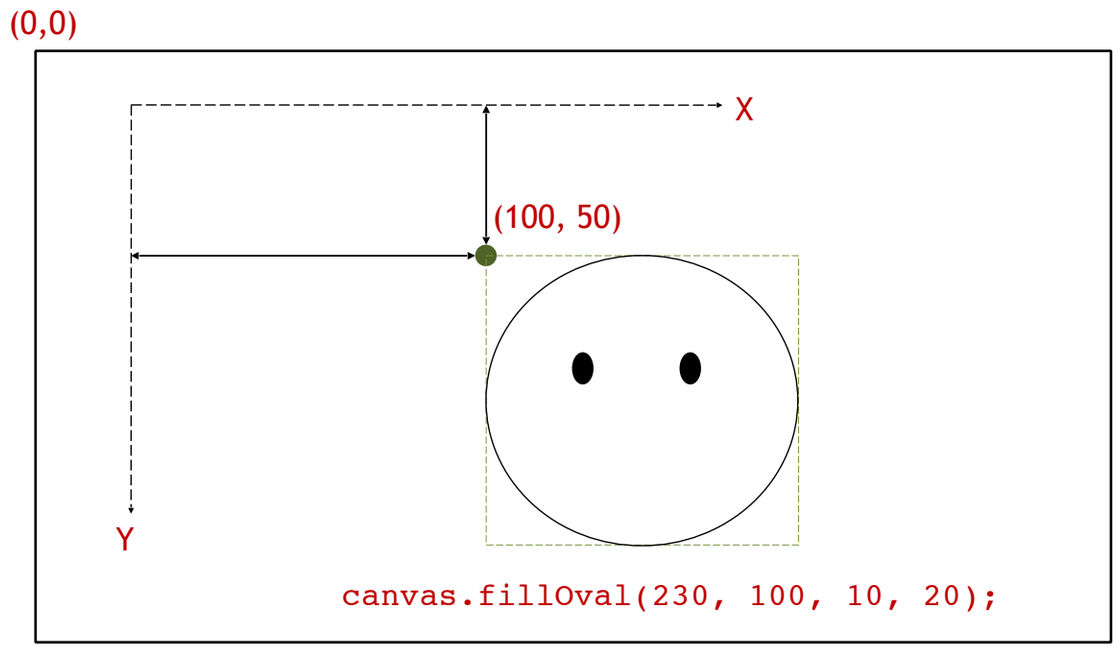
Each time you need to draw a graphics object, you will do so by specifying the subscribing rectangle. This rectangle is defined in terms of the coordinates of the top left corner, which defines its origin and its width and length.

The actual object is drawn within this rectangle, based on additional geometric attributes needed by the specific command.

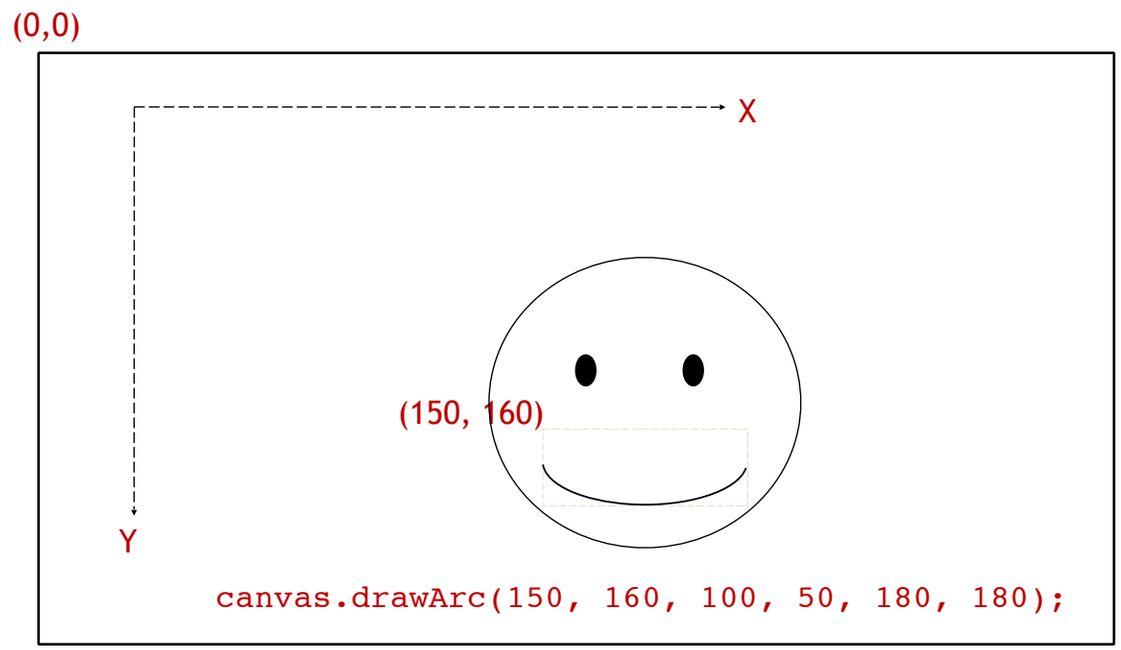
Filling an Oval



Filling an Oval



Drawing an Arc



Arcs are drawn using an angle coordinate system with 0 degrees on the right, 90 degrees up, 270 left and 360 back right.

When an arc is swept counterclockwise the angle is positive and when clockwise it is negative.