

Summary of Lesson 1

Introduction

What's C?

- Almost all computer languages are categorized into **Interpreter language** and **Compiler language**¹
- **C** is a **Compiler language**
- Generally, Compiler languages are **faster** than **Interpreter languages** but need to be **compiled** with a **Compiler** before run the program.
- **Interpreter languages** need **Interpreter application** to run the software. Max/MSP is also an **Interpreter application** of its own language

What kind of software can I program with C

- **Almost all**, Unix is in fact written in C language.
- There are huge amount of **libraries** for all kinds of application, like OpenGL, CSound, Quicktime, Quickdraw, and so on
- For making a MacOSX Application, You can utilize **Carbon** library.

What's Compile?

- **To Compile** means **to translation** a **Source code** into **Native Code(Object Code)**
- Computer only understands its **Native Code**
- **C Compiling** consist of **two steps**
- **C Compiler** translate **C Source Code** into **Assembly language**
- **Assembler** translate **Assembly language** into **Native Code**

What's XCode?

- **XCode** is an **IDE(Integrated Development Environment)**²
- One can **edit, compile, link, orgnize, debug** source codes and making **GUI** at ease
- Several software can be managed in **one project file**
- **C, C++, Java, Objective-C** are supported

What's the defference between C and Max/MSP?

- Max/MSP is a pseudo **OOP** (Object Oriented Programming)language,
- C is a **procedure** language, the relationship between **functions** are all **parallel**.

What's function?

- Functions are set of tasks like a **patch, subpatch**.
- There are 3 kinds of functions, **Main function**,(sub) **Function**, and **Library function**
- Allmost all C programs consists of **one** Main Function and several (sub)**Functions**
- Programmer can program only **Main** and (sub)**Functions**.
- **Library functions** are **ready-made functions**, Programmer can **call(use)** them but can not read the source codes of them (it's already compiled)
- To **call**, to **invoke** means to use a function

¹ Compiler languages: C, C++, Objective-C Interpreter languges: Basic, Perl, PHP Lisp, Ruby, Javascript, Max/MSP, Python Others:Java, C#, Super Collider

² Another famous IDE:Visual Basic, Visual C+,C++ Builder,Eclipse,Code Warrior, Real Basic

- Normally ,Functions need **parameters** to execute its content, but some functions do not need them
- Normally, Functions return a **value**, but **some functions**(void function) doesn't return it
- Functions are always with **parenthesis ()**, in the parenthesis you can put parameteres.
- After a function excuted, all data used in the function disappears
- Functions are **black box**, a function dosen't know how the other functions work. they can know only return value, after invoking

lesson1_1.c

```
#include <stdio.h>

int main(void)
{
    printf("Hello Computer, I am Chikashi");
    return 0;
}
```

lesson1_2.c

```
#include <stdio.h>

/* This is my first C programming */
int main(void)
{
    printf("Hello Computer, I am Chikashi"); // call library function, printf
    return 0; // return 0 means, "This program finished correctly"
}
```

lesson1_3.c

```
#include <stdio.h>

/* output with two lines */
int main(void)
{
    printf("Hello Computer, I am Chikashi.\nI am 26 years old.");
    return 0;
}
```

lesson1_4.c

```
#include <stdio.h>

/* printf automatically format its first and second parameters*/
int main(void)
{
    printf("Hello Computer, I am Chikashi\nI am %d years old.", 26);
    return 0;
}
```