



Microcontrollers & Applications

Lecture 2.2: General Structures & Libraries & File Names & Indenting

General Structure (C & Python)

```
int main(void) {
```

```
    printf("Hello world!\n");
```

```
    return 0;
```

```
}
```

```
// After finishing all codes, it go back to the OS.
```

```
print("Hello world!\n")
```

```
# After finishing all codes, it go back to the OS.
```

General Structure (Arduino & CircuitPython)

```
void setup() {
```

```
    // Whatever you code here,
```

```
    // it runs first but once only
```

```
}
```

```
void loop() {
```

```
    // Whatever you code here,
```

```
    // it runs until electricity is down
```

```
}
```

```
# Whatever you code here,
```

```
# it runs first but once only
```

```
while (True):
```

```
    # Whatever you code here,
```

```
    # it runs until electricity is down
```

Library Use (C & Python)

```
#include <stdio.h>
```

to use all commands and definitions given under the standard library of stdio.h

```
#include "lcd.c"
```

to use all commands and definitions given under the library saved under the given folder or the same folder with the main file of lcd.c

```
import matplotlib.pyplot as plt
```

to use all commands and definitions related to pyplot class given under the library of matplotlib using the abbreviation of plt

```
import numpy
```

to use all commands and definitions given under the library of numpy using its full name of numpy

```
from math import sqrt, pi
```

to use sqrt command and pi definition directly given under the library of math

Library Statements Use (Python-specific case)

```
import math
```

```
x = math.sqrt(64)
```

```
print(x)
```

```
y = sqrt(64) # gives error
```

```
print(y)
```

```
from math import sqrt
```

```
x = math.sqrt(64) # gives error
```

```
print(x)
```

```
y = sqrt(64)
```

```
print(y)
```

File Naming (C & Python)

- `.c` **code file**
- `.ino` **Arduino code file**
- `.h` **definition file**
- File names have the same rule with variable names:
 - `main.c`
 - `constants.h`
- `.py` **code file**
- File names have the same rule with variable names:
 - `code.py` # CircuitPython main code file
 - `constants.py`

Indenting (C & Python)

```
int main(void) {
```

```
    while (true)
```

```
        printf("Hello world!\n");
```

```
    return 0;
```

```
}
```

```
---
```

```
int main(void) {
```

```
    while (true) printf("Hello world!\n");
```

```
    return 0; }
```

```
---
```

```
int main(void) { while (true) printf("Hello world!\n"); return  
0; }
```

```
while (True):
```

```
    print("Hello world!")
```

```
---
```

```
while (True):
```

```
    print("Hello world!")
```

Indenting with Grouping (C & Python)

```
int main(void) {
    while (true) {
        printf("Hello world!\n");
        printf("Hello world!\n");
    }
    return 0;
}
---
```

```
int main(void) {
    while (true) { printf("Hello world!\n");
                    printf("Hello world!\n");}
    return 0;
}
---
```

```
int main(void) { while (true) {printf("Hello world!\n"); printf("Hello world!\n"); } return 0; }
```

```
while (True):
    print("Hello world!")
    print("Hello world!")
---
```

```
while (True):
    print("Hello world!")
    print("Hello world!")
---
```

```
while (True):
    print("Hello world!"); print("Hello world!")
---
```

```
while (True): print("Hello world!"); print("Hello world!")
---
```

```
while (True):
    print("Hello world!")
    print("Hello world!")
```




Thanks for
listening 😊

YALÇIN İŞLER

Assoc. Prof.