



C

next page →

C +

- qq 2820047809 (Trdthg)

---

☺ For

IO Club

---

🦊 Presenter

20

---

🐧 QQ

2820047809

---

👾 Github

trdthg



盲目自学只会毁了  
完整300集 这还学  
技大 逃出编程

【清华大学】C语言课程，  
通C语言程序设计！C语言  
秃头美少年坤坤 · 9-14

## 是2022最好的 C语言视频

7.5万 10.6万 142:52:38

FishC Studio

## 带你学C带你飞

640.2万 11.1万 17:44:08

编程2021C语言零基础视  
C语言入门初学者小白...

大学计算机... · 2021-8-24

【C语言】《带你学C带你飞》

鱼C-小甲鱼 · 2018-7-26

C

C

C

C++

<https://github.com/search?q=>

OI-Wiki

OJ OJ

言学那么多语法有用吗？

一个小球的编程之旅



球，27分钟，讲清楚所  
语言语法知识

老师 · 5-19



翁恺C语言

carrot-boer · 2018-1-6

## C语言教程

卷不动了  
整整600集都沒

【C语言教程600集】目前最

完整的C语言教程，包含所

编程不加班 · 9-24

语言入门详解  
C Programming Language In Detail

1.8万 2.6万 36:19:58

嵌入式C语言

14.3万 1287 14:27:39

《C#语言入门详解》全

嵌入式C语言

手把手编程  
100道

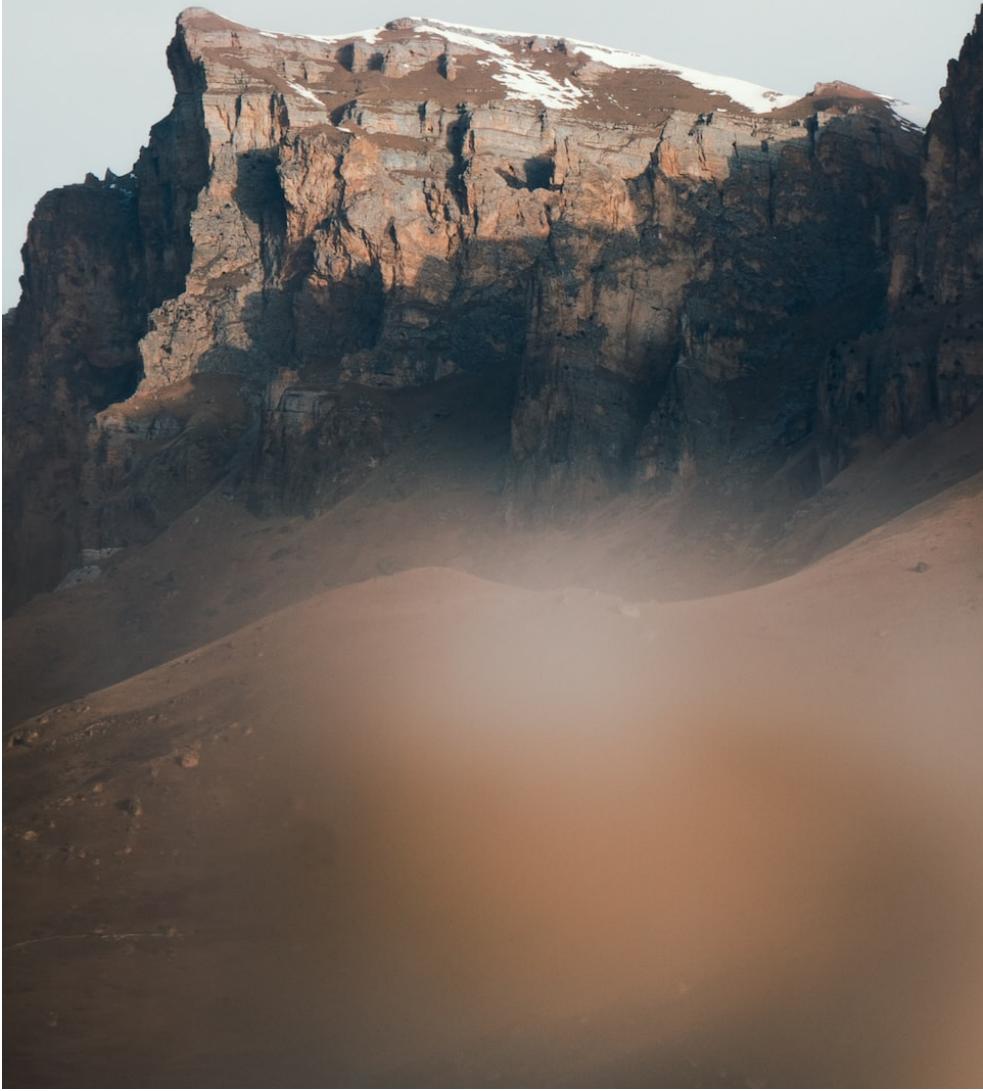
13.2万 969

C语言经典100题（手把手  
）

- C
- IDE /
- C /
- 

```
#include <stdio.h>

int main() {
    printf("Hello World ");
    return 0;
}
```



C

1. C
  - C
2. C
3. C
4. / / ...



C

1.

■

■ gcc

■ CMD

2.

■

■

■

{ nano +

2.

- Dev C++



- VS



Visual C++ 6

- - gcc

C / Cpp & .<sup>[1]</sup>

## Logo

Linux **gcc** GNU  `apt install gcc`

Windows **gcc** MinGW<sup>[2]</sup>  **MinGW-w64**  
A complete runtime environment for GCC & LLVM  
for 32 and 64 bit Windows ` ` or Dev C++

Windows **CL** MSVC  VS

All **clang** Clang & LLVM  `apt install clang` VS

1. Binutils GCC Glibc

2. Minimalist GNU for Windows

1. Dev C++ VS



+ + + +

- IDE
- vscode



2.

MinGW (gcc) MSVC (cl)

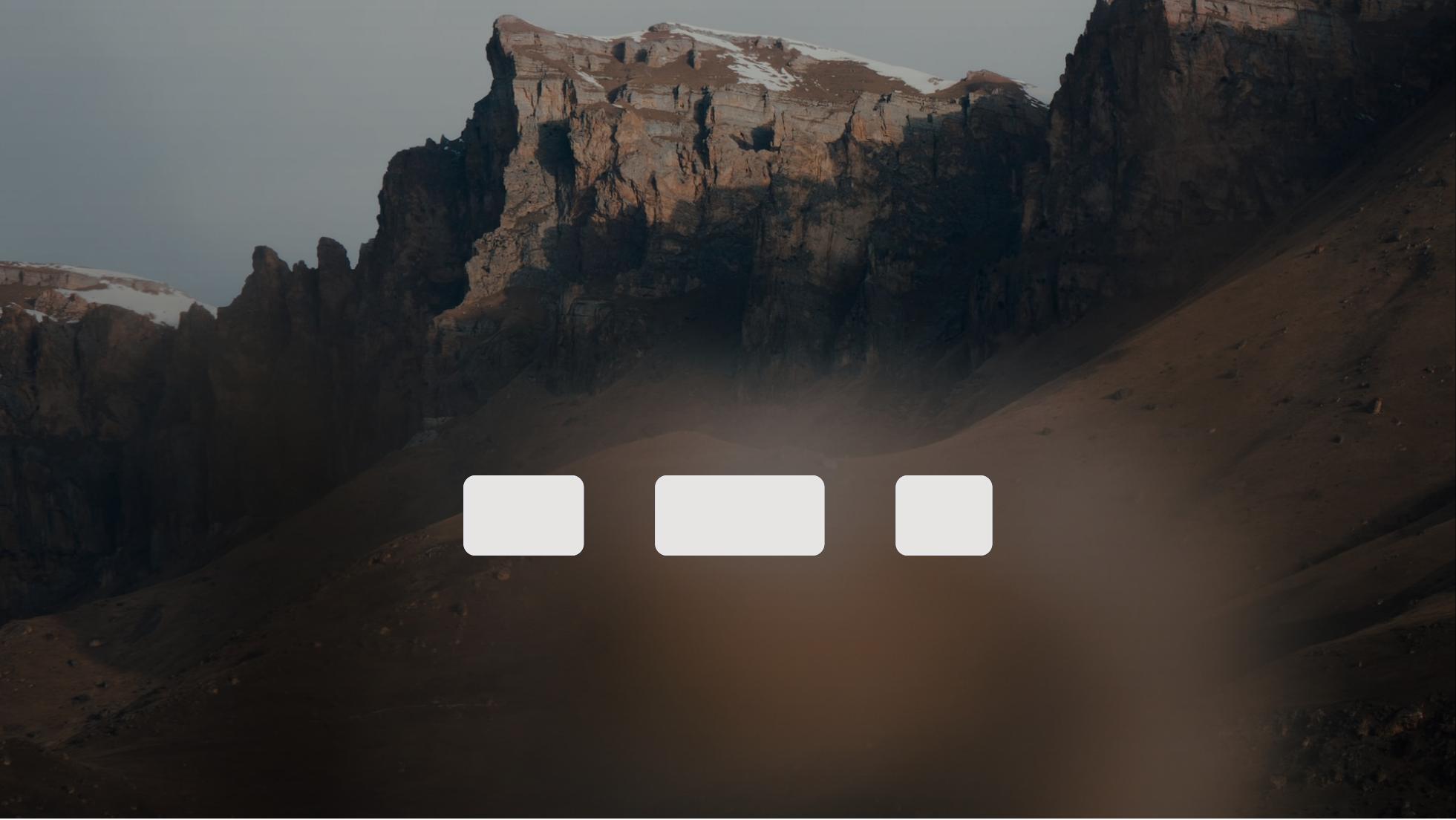
- 
- ( + )

3. gcc

- cph
- c/c++

# C

- 
- 
- 
- |
- IDE /
- &      /      /
- & IO &
- malloc
- Git / Github
- Markdown / Editor



Rust

Go

```
package main

func main() {
    var a8 int8 = 8
    var a16 int16 = 8
    var a32 int32 = 8
    var a64 int64 = 8

    var u8 uint8 = 0
    var u16 uint16 = 0
    var u32 uint32 = 0
    var u64 uint64 = 0

    // byte    uint8
    var c2 uint8 = 'a'
    var c1 byte = 'a' // byte    uint8

    print(a8, a16, a32, a64, u8, u16, u32, u64)
}
```

# C

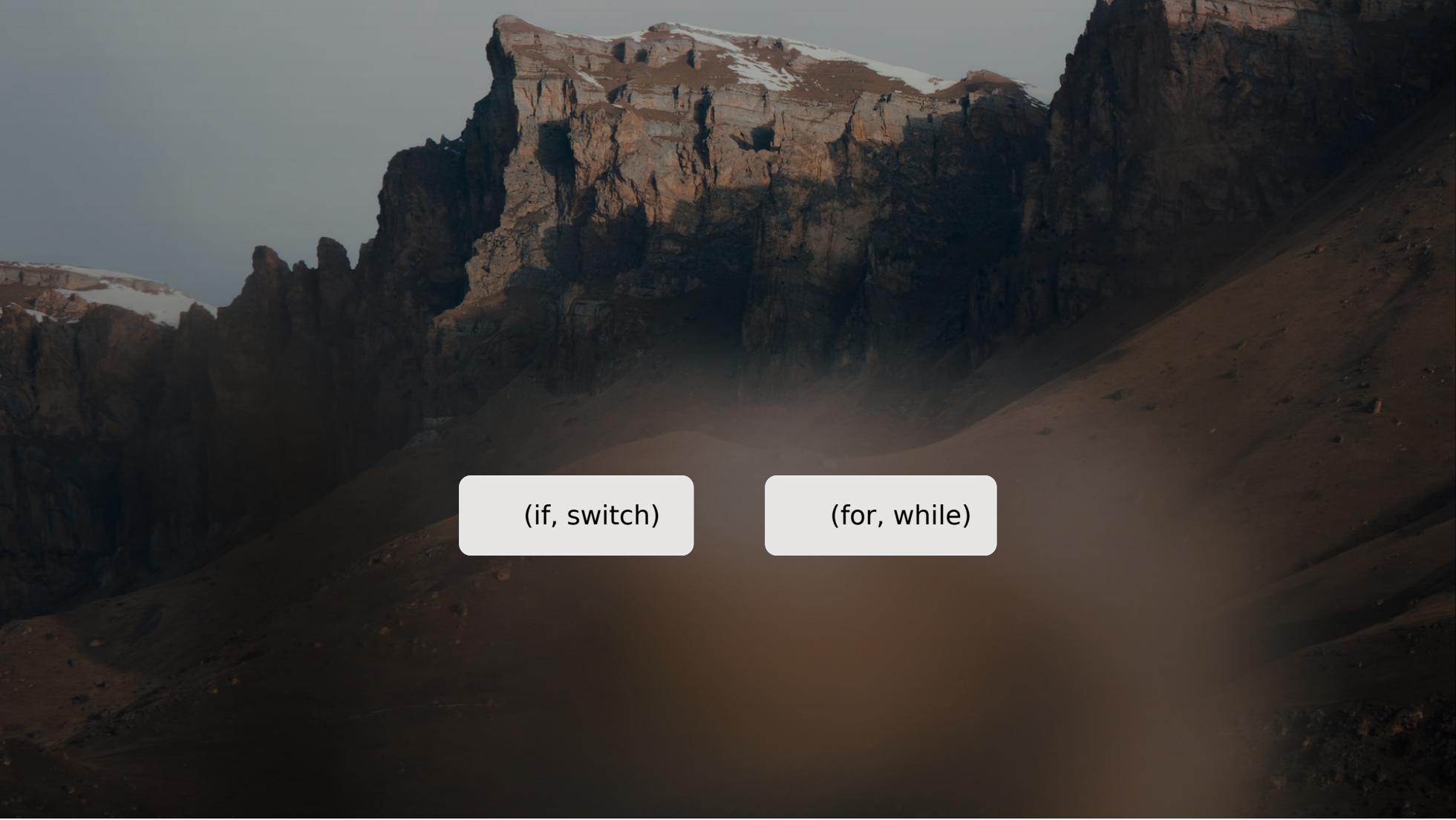
```
char a1 = 127;      // 8 0000 0000 -128 - 127
short a2 = 1;       // 16
int a3 = 0;         // 32
long a4 = 0;        // 64
long long a5 = 0;   // 128 0000 0000 0000 0000 0000 0000 0000

unsigned char b1 = 255; // 8
unsigned short b2 = 1; // 16
unsigned int b3 = 0; // 32
unsigned long b4 = 0; // 64
unsigned long long b5 = 0; // 128
```

```
float c1 = 1.0;      // 32
double c2 = 1.0;     // 64
long double c3 = 1.0; // 128
```

```
char c1 = 'a';
char c2 = '\0';
```

# ASCII

The background of the image is a dark, moody landscape of mountains. A prominent peak in the center-right is heavily shadowed, while its upper slopes and the peaks to its left are partially illuminated by a low sun, casting long shadows and highlighting rocky textures. The foreground is dark and indistinct.

(if, switch)

(for, while)

# if

```
if (xxx) {  
    ...  
} else if (xxx) {  
    ...  
} else {  
    ...  
}
```

```
if (i == 0)  
    printf("      \n");  
else if (i == 1)  
    printf("      \n");  
else  
    printf("      \n");
```

```
#include <stdio.h>  
int main() { // scope 0  
    int i = 0;  
  
    { // scope 1  
        int a = 0;  
        printf("%d", i);  
    }  
  
    printf("%d", a);  
  
    { // scope 2  
        int b = 0;  
        printf("%d", i);  
    }  
  
    return 0;  
}
```

# switch

## FSM

```
switch (i)
{
    case 1:
        statement1;
        statement2;
        break;
    case 2:
        statement;
        break;
    ...
    default:
        statement;
}
```

```
#include <stdio.h>
int main() {
    return 0;
}
```

for

```
for (      ;      ;      ) {  
    ...  
}  
  
for (;;) {  
    //  
}
```

while

```
while (      ) {  
    ...  
}
```

Break Continue

```
#include<stdio.h>  
int main() {  
    for(int i = 0; i < 5; i++) {  
        if (i == 3) {  
            continue; //  
            break;    //  
        }  
        printf("%d\n", i); // 0 1 2  
    }  
}
```

1.

$f(x)$

$y = |x|$

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

```
int main() {
```

```
    return 0;
```

```
}
```



2.

```
int add(int a, int b) {  
    return a + b;  
}
```

# void ?

## void

1.

```
void printSomeSentence()
{
    printf("=====\\n");
    printf("Company *****\\n");
    printf("Fax ***** \\n");
    printf("Email ***** \\n");
    printf("=====\\n");
}
```

2.

```
void printSomeSentence(void)
{
    printf("=====\\n");
    printf("Company *****\\n");
    printf("Fax ***** \\n");
    printf("Email ***** \\n");
    printf("=====\\n");
}
```

# main

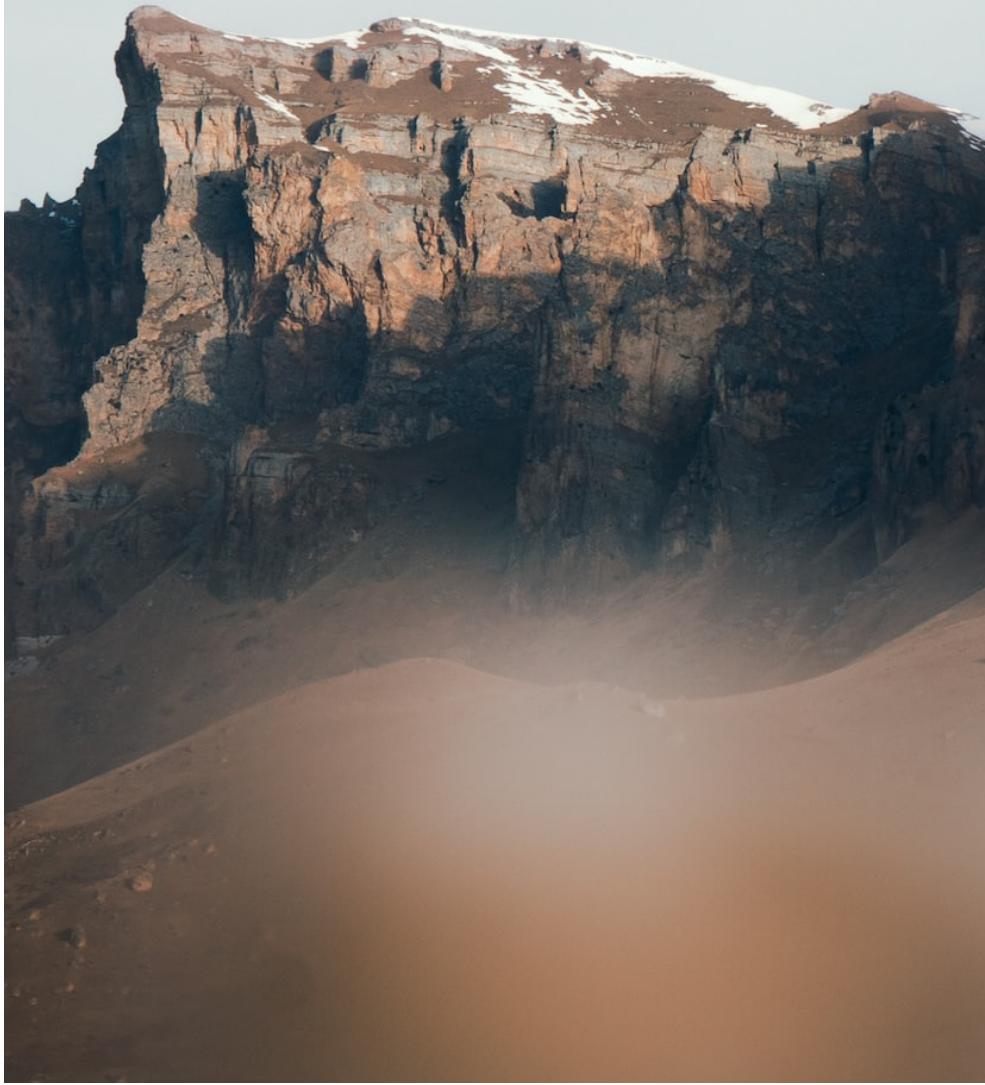
```
#include<stdio.h>
int main(int argc, char* argv[]) {
    // argc:
    // argv:

    return 0;
}
```

C/C++

```
int a[6];  
  
char b[6];  
  
double c[6];
```

```
#include <stdio.h>  
int main() {  
    char chars[5] = {'a', 'b', 'c', 'd', 'e'};  
    //          0   1   2   3   4  
  
    char chars_first = chars[0];  
    char chars_last = chars[4];  
  
    for (int i = 0; i < 5; i++) {  
        printf("%c\n", chars[i]);
```





```
char chars[10] = {'a', 'b', 'c', 'd', 'e', '\0'};  
printf("%s", chars); // abcde
```

`\0`

```
char chars[10] = {'a', 'b', 'c', 'd', 'e', '\0', 'a'  
printf("%s", chars); // abcde
```

```
char chars[] = {"abcde"};  
char chars[] = "abcde";
```



# C string.h

1.

```
#include <string.h>
char str[] = "abcde";
int len = strlen(str);
```

2.

```
char str1[] = "abcde";
char str2[10];
strcpy(str2 , str1);
```

3.

```
char str1[] = "aaa";
char str2[] = "bbb";

int res = strcmp(str1, str2);

if (res == 0)
    printf("str1      str2\n");
else
    printf("str1      str2      \n");
```

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int arr[10][10] = {};
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

# main

```
#include<stdio.h>
int main(int argc, char* argv[]) {
    return 0;
}
```

# `#define`

```
#define LIGHT 1
#define DARK 1

enum Week {
    Monday = 0,
    Tuesday = 1,
    Wednesday = 2,
};

int main(int argc, char const *argv[]) {
    printf("%d\n", LIGHT);
    return 0;
}
```

1. (Preprocessing)
  - #include
2. (Compilation)
  -
3. (Assemble)
  -
4. (Linking)
  - (.so / .dll)

templete