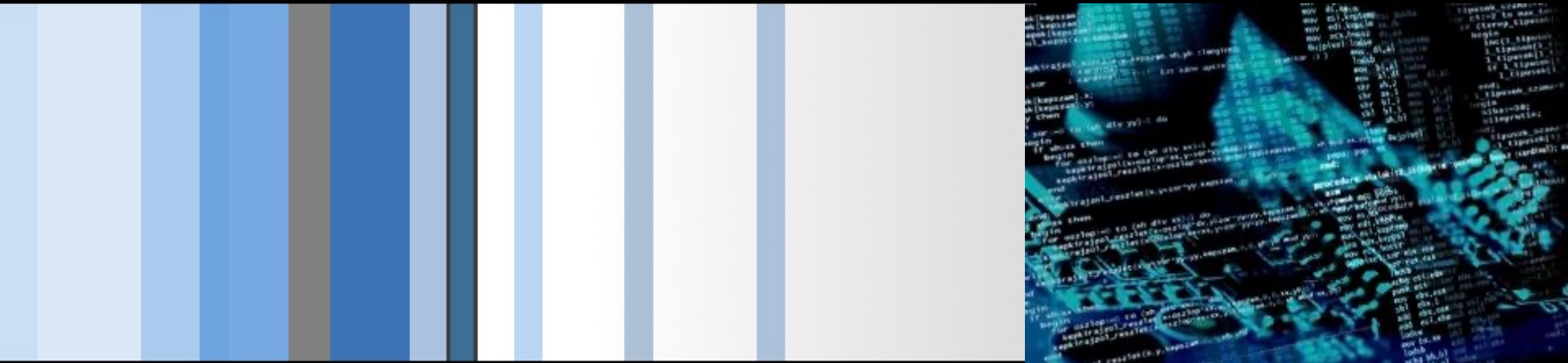


CSC 472 Software Security

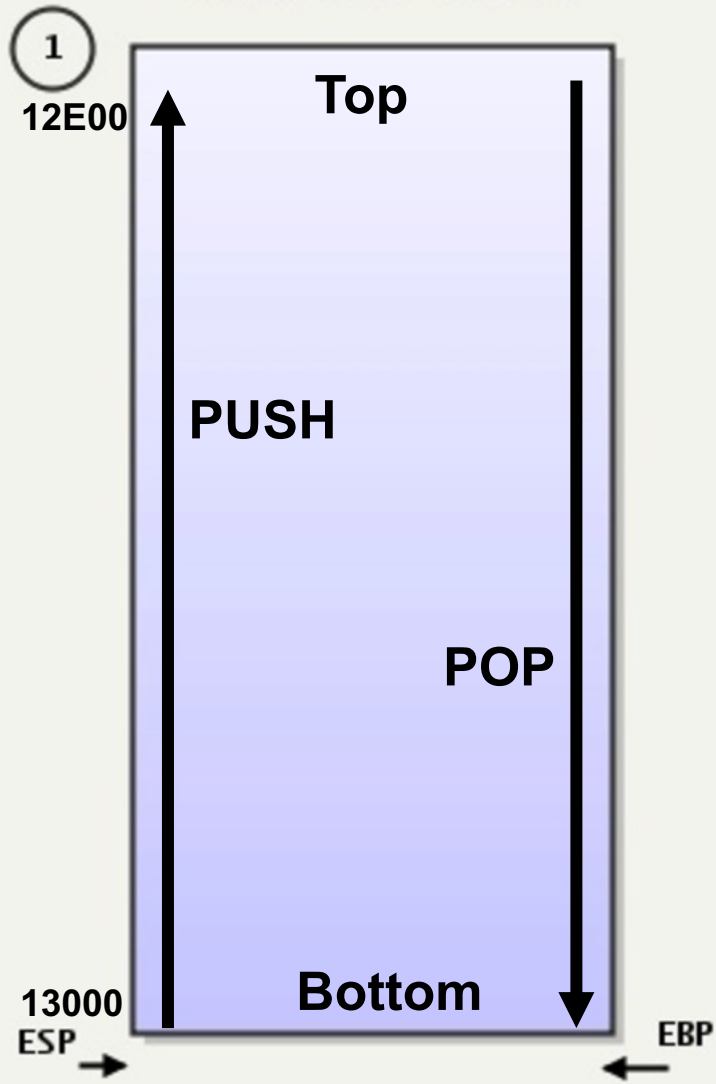
Stack & Stack Frame

Dr. Si Chen (schen@wcupa.edu)



The Stack

Stack frame details



Stack:

- A special region of your computer's memory that **stores temporary variables** created by each functions
- The stack is a "**LIFO**" (last in, first out) data structure
- Once a stack variable is freed, that region of memory becomes available for other stack variables.

Properties:

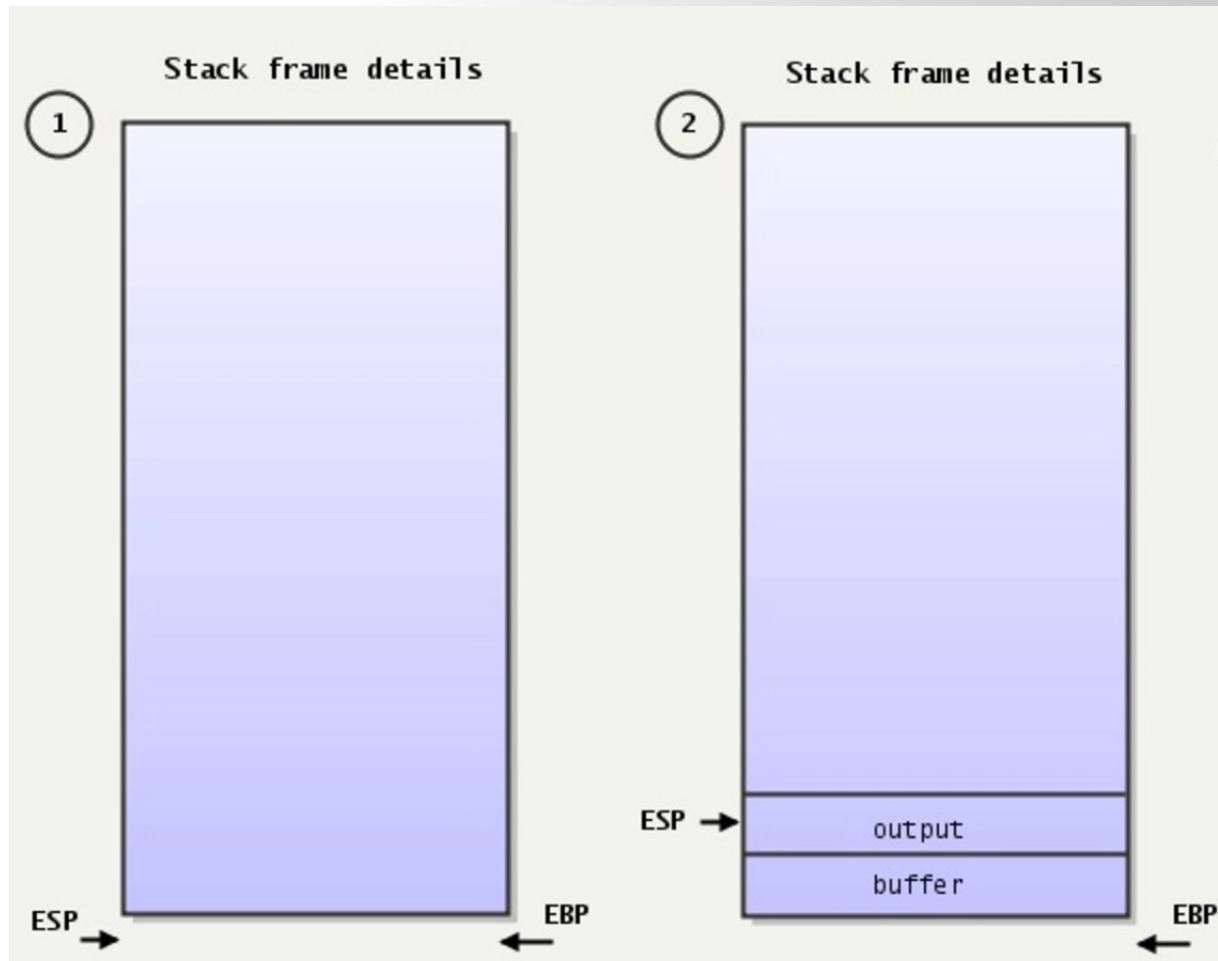
- the stack grows and shrinks as functions **push and pop** local **variables**
- there is no need to manage the memory yourself, variables are allocated and freed **automatically**
- the **stack has size limits**
- stack variables only exist while the function that created them, is running

EBP—Pointer to data on the stack
ESP—Stack pointer

The Stack

Stack:

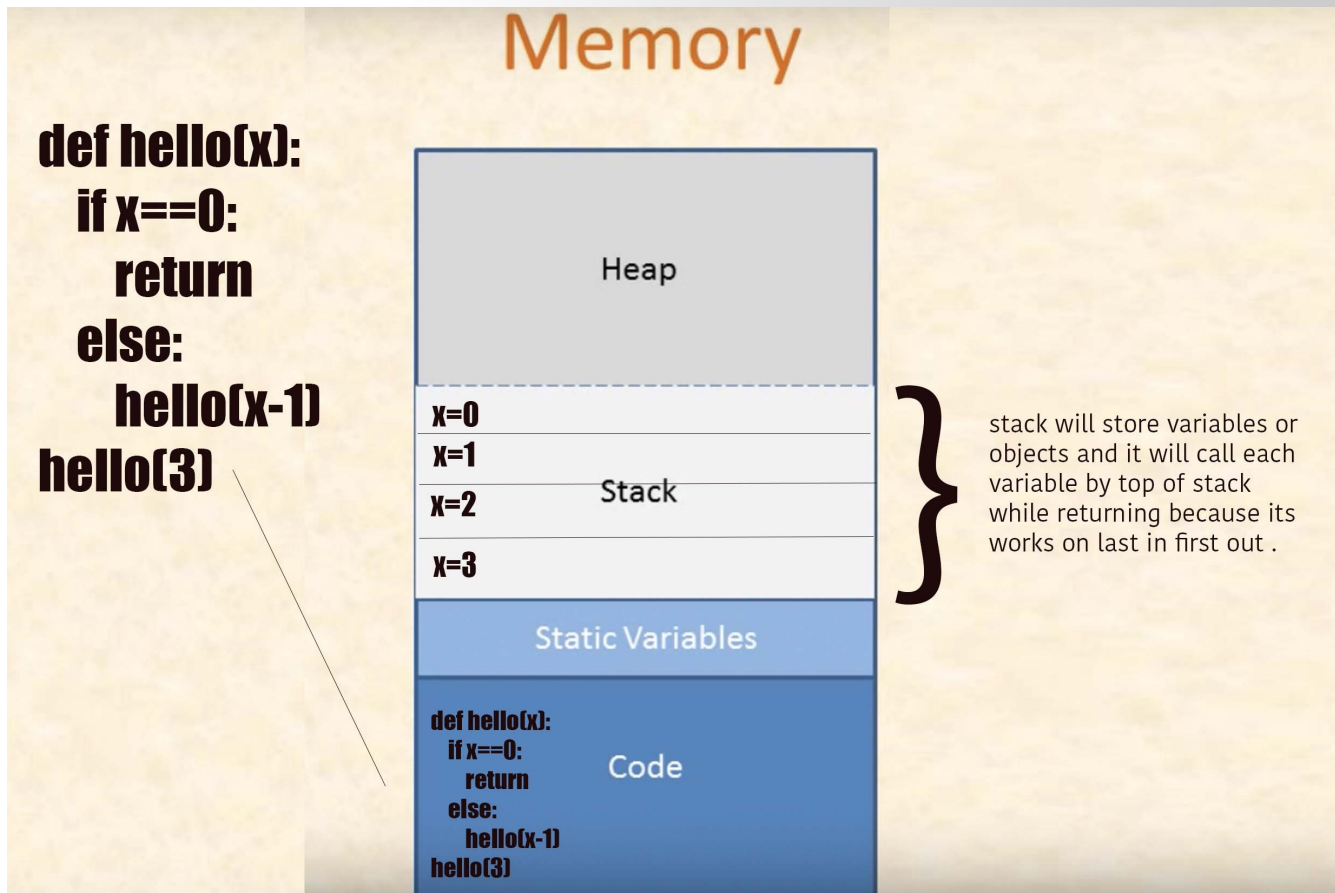
- A special region of your computer's memory that **stores temporary variables** created by each functions
- The stack is a "**LIFO**" (last in, first out) data structure
- Once a stack variable is freed, that region of memory becomes available for other stack variables.



Stack Frame

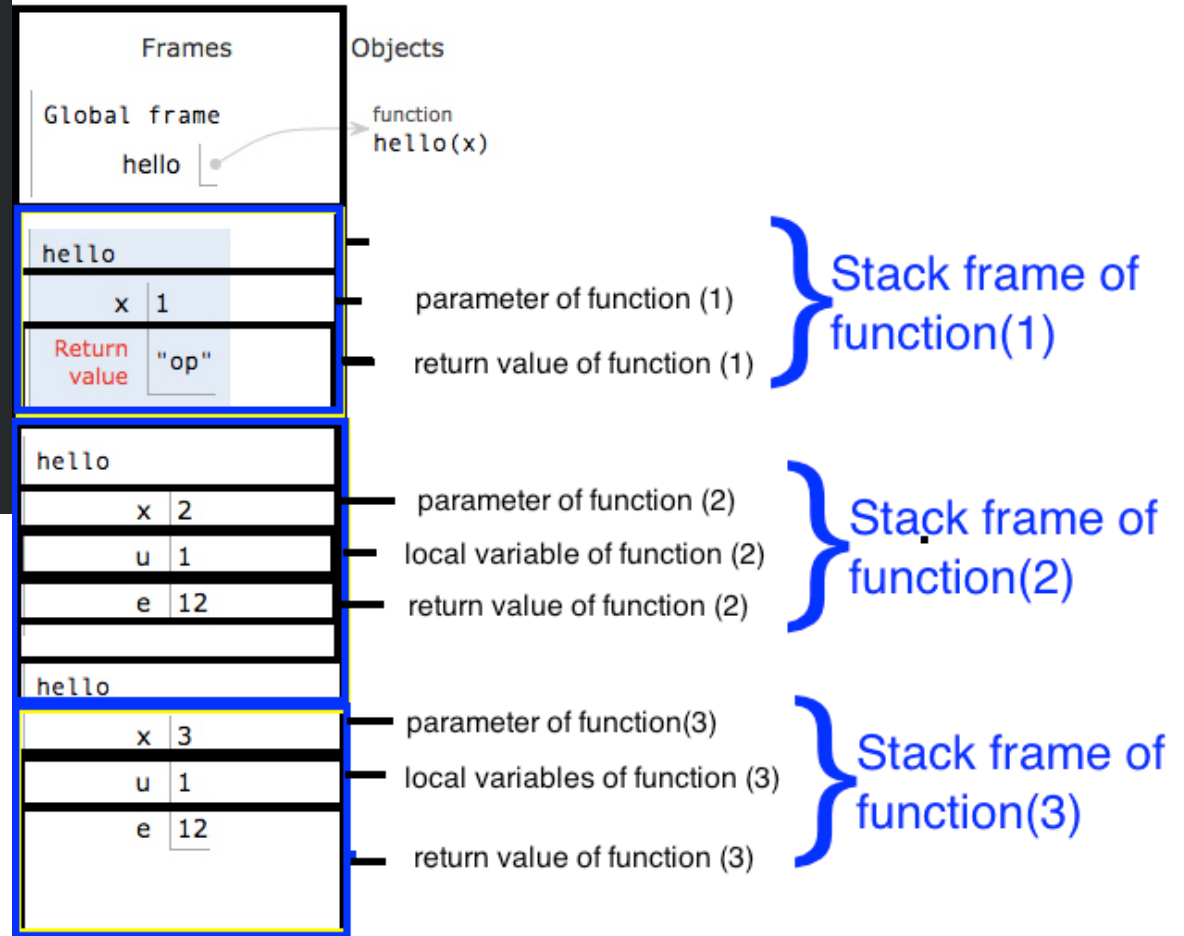
Stack Frame

- A stack frame is **a frame of data that gets pushed onto the stack.**
- In the case of a **call stack**, a stack frame would represent **a function call and its argument data.**



Stack Frame

```
1 def hello(x):  
2     if x == 1:  
3         return "op"  
4     else:  
5         u = 1  
6         e = 12  
7         s = hello(x - 1)  
8         e += 1  
9         print(s)  
10        print(x)  
11        u += 1  
12    return e  
13  
14  
15 hello(3)
```



Functions and Frames

Each function call results in a new frame being created on the stack.

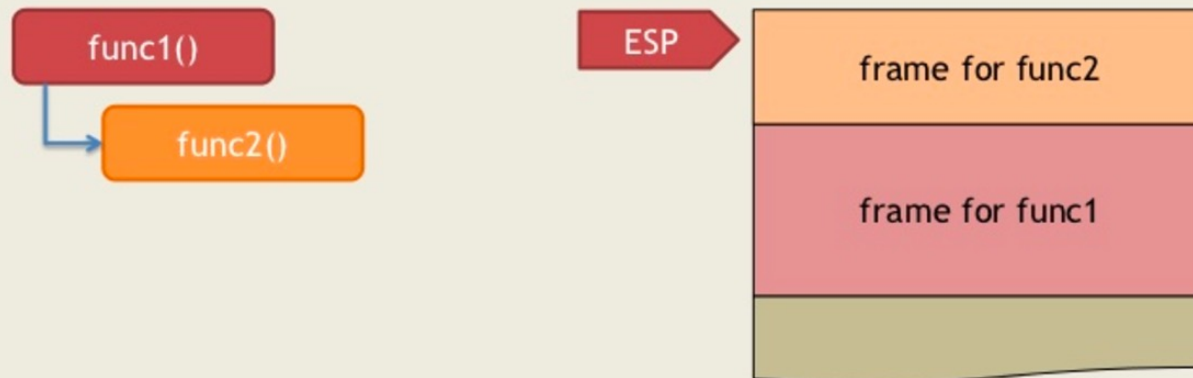
func1()

ESP

frame for func1

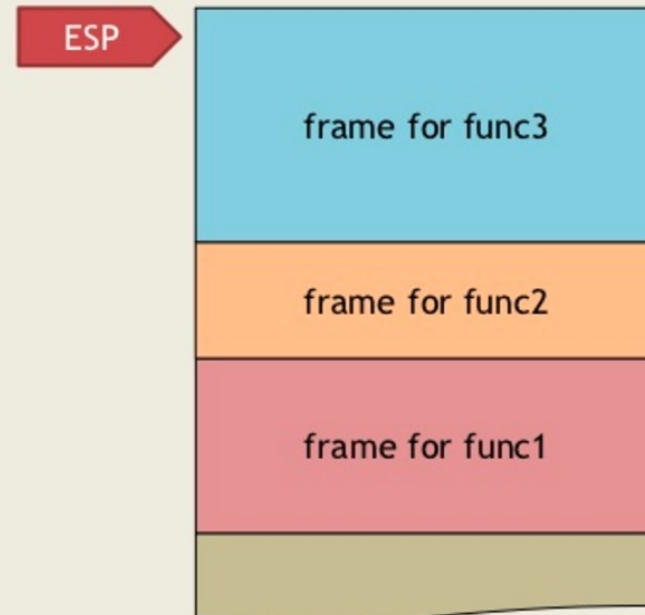
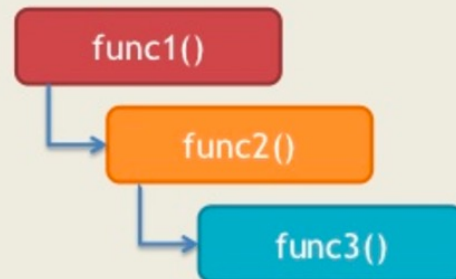
Functions and Frames

Each function call results in a new frame being created on the stack.



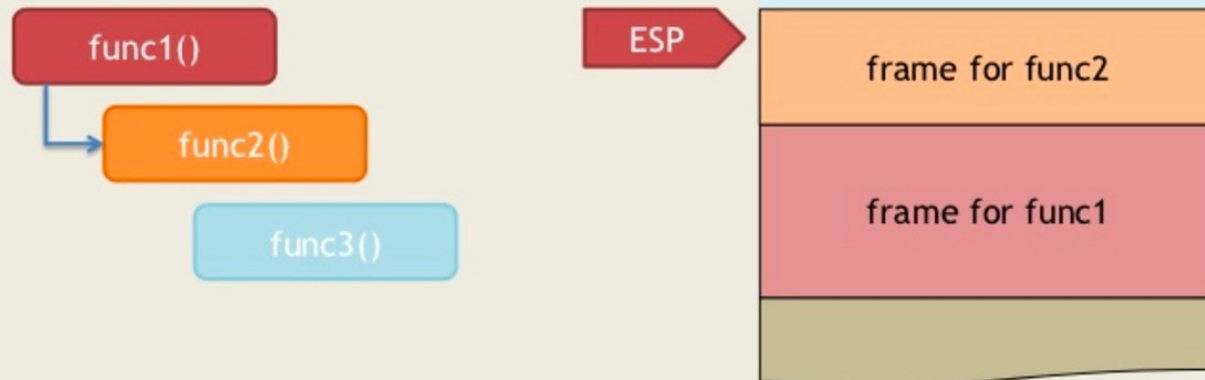
Functions and Frames

Each function call results in a new frame being created on the stack.



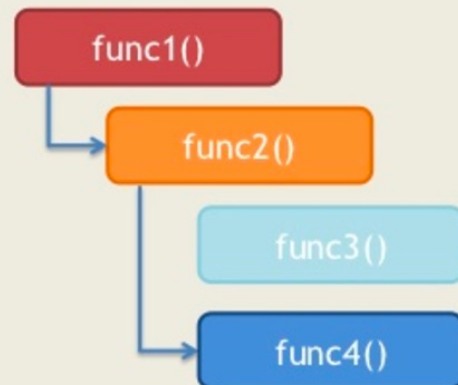
Functions and Frames

When a function returns, the frame is "unwound" or "collapsed".

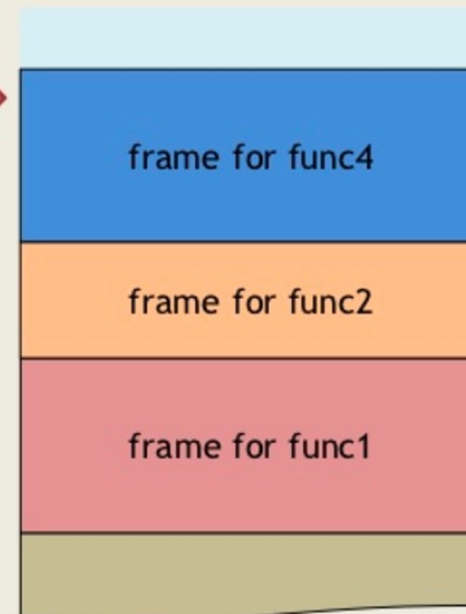


Functions and Frames

And as new functions get invoked, new frames get created.



ESP



Stack Frame

File Edit View Terminal Tabs Help

```
PUSH EBP      ; start of the func (save current EBP to stack)
MOV EBP, ESP  ; save current ESP to EBP

.....      ; function body
              ; no matter how ESP changes, the EBP remains unchanged

MOV ESP, EBP  ; move the saved function start addr back to ESP
POP EBP       ; before return the func, pop the stored EBP
RETN          ; end of the func
```

-- INSERT --

12,1

All

StackFrame.c

```
1 StackFrame.c +
  1 #include "stdio.h"
  2
  3 long add(long a, long b)
  4 {
  5     long x = a, y = b;
  6     return (x + y);
  7 }
  8
  9 int main(int argc, char* argv[])
10 {
11     long a = 1, b = 2;
12     printf("%d\n", add(a,b));
13     return 0;
14 }
15
```

Q & A

