

Data structure

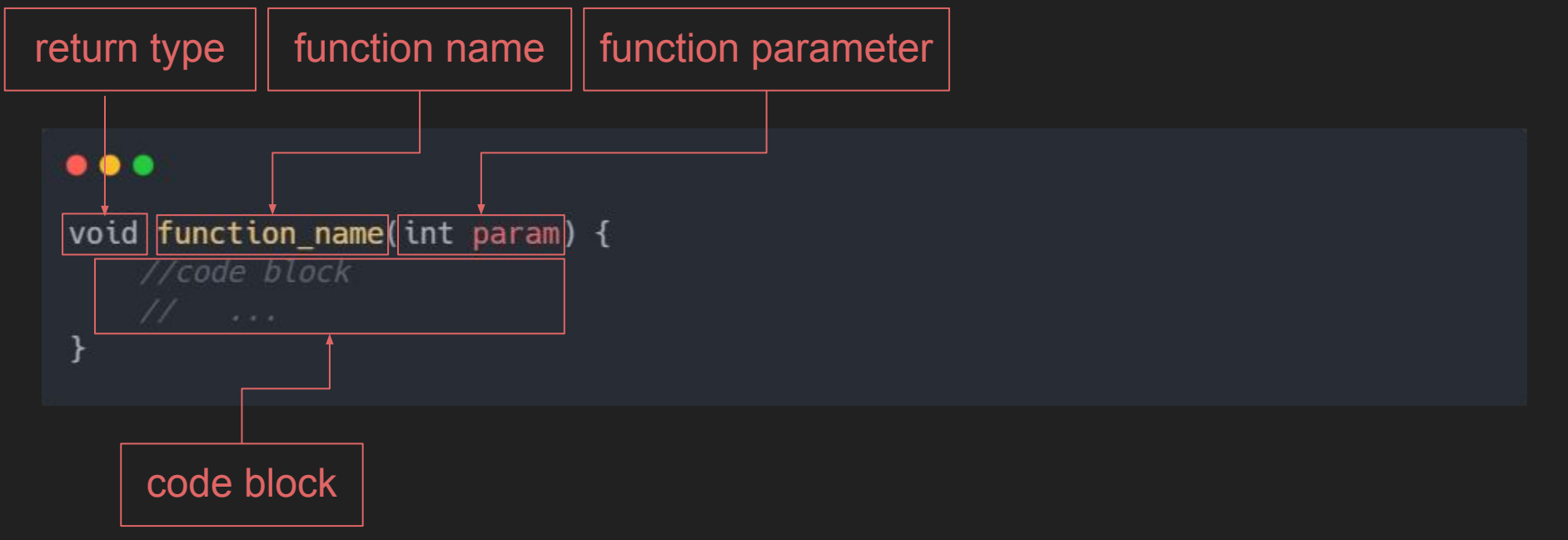
Function Pointer

1. 함수 구조

return type

function name

function parameter



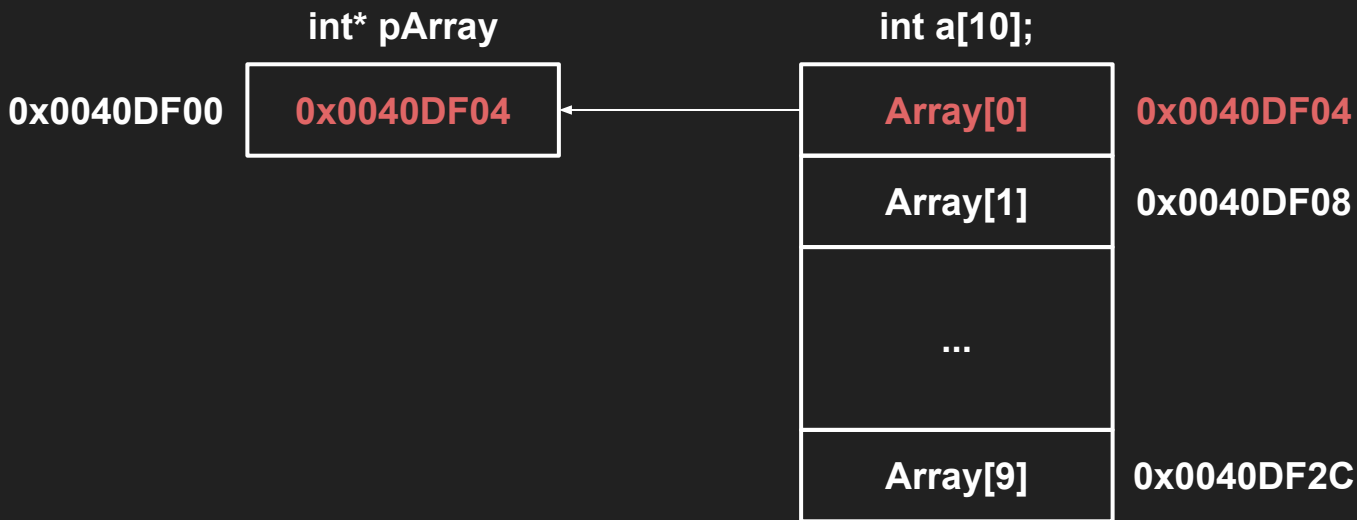
```
void function_name(int param) {  
    //code block  
    // ...  
}
```

The diagram illustrates the structure of a C++ function. It shows a code snippet with four labels and arrows pointing to specific parts: 'return type' points to 'void', 'function name' points to 'function_name', 'function parameter' points to '(int param)', and 'code block' points to the body of the function enclosed in curly braces. The code block contains two lines of comments: '//code block' and '// ...'.

code block

2. 포인터

- **포인터(pointer)** : 프로그래밍 언어에서 다른 변수, 혹은 그 변수의 메모리 공간주소를 가리키는 변수를 말한다.
- 포인터가 **가리키는 값**을 가져오는 것을 **역참조**라고 한다.



3. 함수의 주소

- 함수의 이름은 메모리에 올라간 함수의 시작 주소를 가리키는 포인터 상수 (constant pointer)가 된다.

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

```
void function() {  
    printf("called function");  
}
```

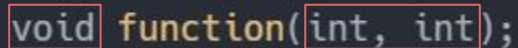
```
int main() {  
    printf("%p", function);  
}
```

함수의 주소 값

```
// answer : 0022102D
```

4. 함수 포인터

- 함수 포인터(function pointer) : 함수의 시작 주소를 가리키는 포인터 상수
- 함수 포인터의 포인터 타입은 함수의 반환값과 매개변수에 의해 결정



```
void function(int, int);
```

```
void (*pFunction)(int, int);
```

함수의 반환 형식

매개변수

5. 표기법 단순화

- C Language



```
typedef int (*functionPointer)(int, int);  
//함수 포인터에 functionPointer이라는 새로운 이름을 붙임.
```

```
functionPointer pFunction = function;
```

- C++ Language



```
auto pFunction = function;
```