

챕터2. 나의 첫 번째 C 프로그램

1절. 개발 환경 구축

핵심 키워드	Visual Studio
여기서는 무얼 배울까	
1챕터에서 우리는 C 언어로 프로그램을 만들기 위해서 컴파일러가 필요함을 배웠다. 물론 컴파일러 하나만으로도 프로그래밍할 수 있지만, 컴파일러 이외에도 프로그래머를 돕기 위한 에디터, 프로파일러, 디버거와 같은 프로그램들을 사용하면 더욱 편하게 프로그램을 만들 수 있을 것이다. 이러한 다양한 도구들을 사용하기 편하도록 한데 모은 프로그램을 통합 개발 환경이라고 한다. 이번 절에서는 C 언어를 위한 통합 개발 환경 중 하나인 Xcode를 설치하고 실행하는 시간을 갖는다.	

DELL2023/03/13 22:30

기초 용어 정리

통합 개발 환경

프로그래머를 돕기 위한 에디터, 프로파일러, 디버거, 컴파일러 등의 프로그램을 하나에 모아둔 프로그램과 개발 환경

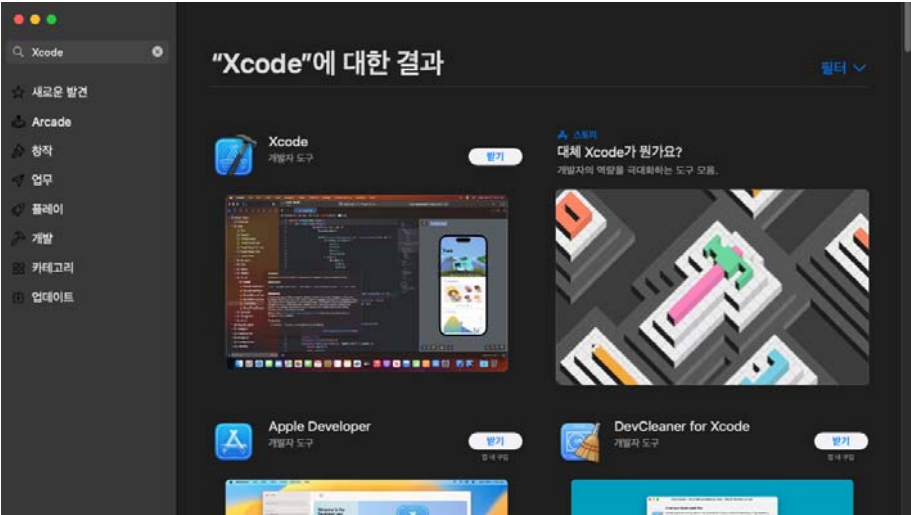
DELL2023/09/18 21:05

코멘트

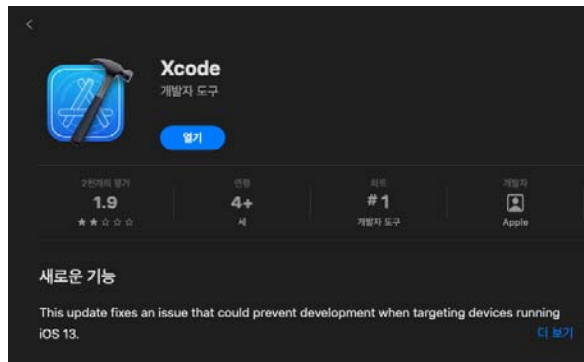
다른 프로그램을 사용한다 하더라도 Apple 환경의 소프트웨어를 개발 및 컴파일하기 위해서는 Xcode가 필요합니다.

1. Xcode 설치하기

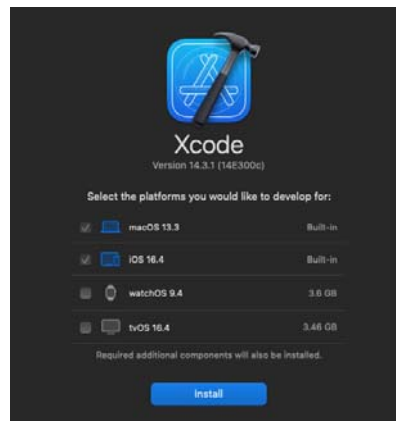
MacOS는 다양한 언어로 프로그램을 개발할 수 있도록 Xcode라는 이름의 통합 개발 환경을 제공하고, C 언어 프로그램 역시 Xcode를 통해 개발할 수 있다. 아래 과정은 Xcode가 설치되지 않았을 경우 App Store에서 Xcode를 설치하는 과정을 설명한다.



① App Store에서 Xcode를 검색하여 받기 및 설치를 누른다.



② Xcode가 설치된 후 열기를 눌러 프로그램을 실행한다.



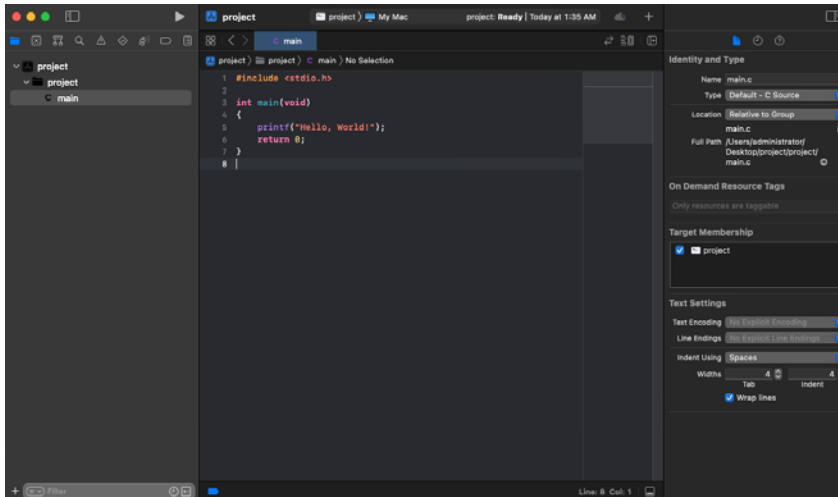
③ 약관에 동의하고 **Install**을 누른다.



④ 설치가 끝나면 Xcode를 사용할 수 있다.

2. Xcode 구성과 로그인

Xcode가 잘 설치되었다면 이를 이용하여 C 언어 프로그래밍을 시작할 수 있다. Xcode로 C 프로그램을 만들기에 앞서 먼저 Xcode의 화면 구성을 살펴보도록 하겠다.



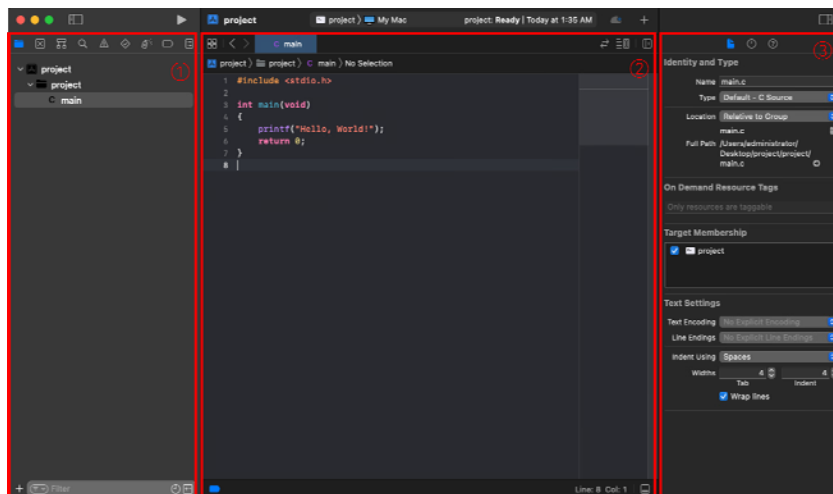
DELL

2023/08/31 14:58

TIP

화면의 구성은 버전과 설정에 따라 상이할 수 있습니다.

프로젝트를 생성하고 실행시키면 Xcode의 화면이 뜬다. 실제 프로젝트를 생성하는 과정은 다음 절에서 다루도록 하고, 먼저 프로그램의 각 부분이 어떤 역할을 하는지 살펴보자.



① 내비게이터 : 프로젝트를 구성하는 파일과 리소스들이 존재하는 영역이다.

- ② 에디터 : 프로젝트의 파일을 열고 편집하는 역할을 하는 영역이다. C 언어 프로그래밍은 주로 에디터에서 C 코드를 작성하는 과정이 주를 이룬다.
- ③ 인스펙터 : 객체의 속성을 관리하는 영역이다. 파일의 이름과 경로, 수정 히스토리 등과 함께 속성과 크기의 설정, 객체와 코드의 연결 등을 할 수 있다. 이 책에서는 인스펙터를 다루지 않는다.

2절. Hello, World!

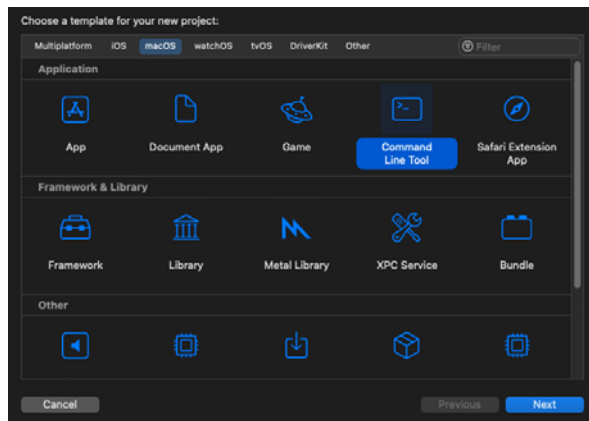
핵심 키워드	소스 파일, 실행 파일, 콘솔
여기서는 무얼 배울까	
<p>세상에서 가장 많은 사람이 만들어본 프로그램은 무슨 프로그램일까? 진지하게 그런 프로그램이 어떤 것인지 조사한 사람은 없겠지만, 단언컨대 지금의 모든 개발자는 이 질문에 Hello, World! 프로그램이라고 대답할 것이다. 그들은 모두 그 프로그램을 만들어봤고, 한 번이 아닌 수십, 수백 번도 만들어봤을 것이며, 그들이 본 다른 모든 개발자도 그 프로그램을 만들었을 것이기 때문이다. 이번 절은 바로 이 Hello, World 프로그램을 작성하고 실행시키는 것이다. 이 과정에서 소스 파일과 실행 파일에 대하여 이해하고, 최종적으로 Hello, World! 프로그램을 만들어 본 수많은 사람 중 한 사람이 되는 것이 목표이다.</p>	

1. 프로젝트 생성하기

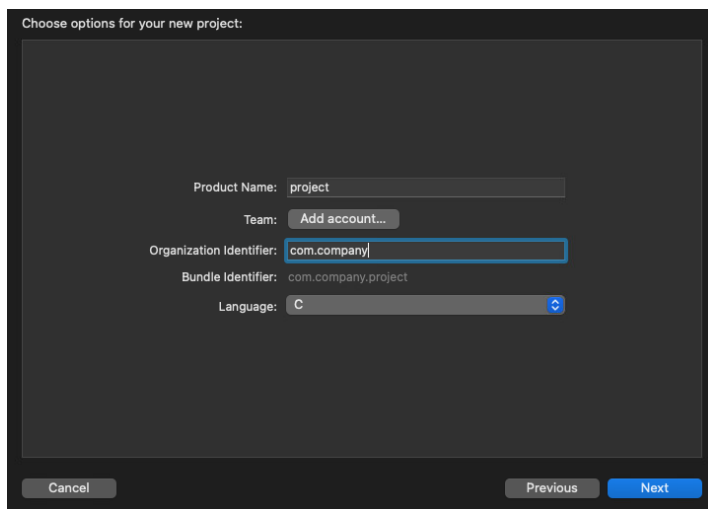
C 언어로 프로그램을 만들기 위한 첫 번째 걸음을 내딛어보자. 이번 절에서 여러분들이 처음으로 만들게 될 C 언어 프로그램은 검은색 화면에 흰색 글씨로 Hello, World! 라는 글자가 표시되는 프로그램이다. 프로그램을 만들기 위해선 먼저 그 프로그램을 위한 프로젝트를 만들어야 하기에 우선 프로젝트를 먼저 생성한다.



- ① Xcode를 실행시키면 기존의 프로젝트를 실행하거나 새 프로젝트를 만들 수 있다. Create a new Xcode project를 눌러 새 프로젝트를 만든다.



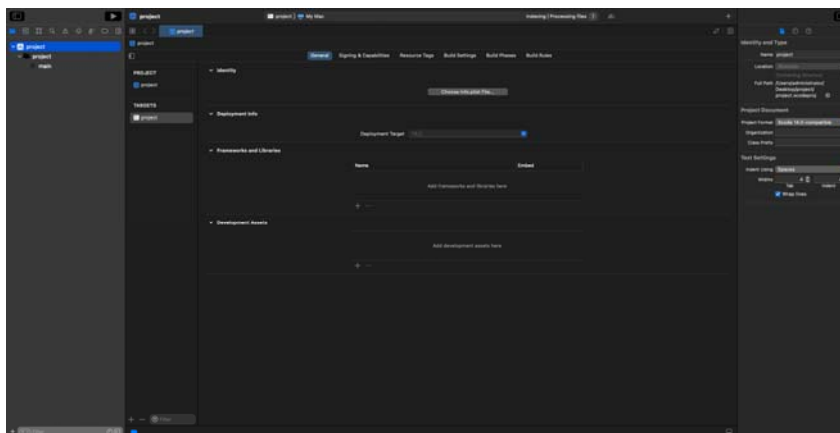
② 그 다음 프로젝트의 템플릿을 선택하는데, 여기서는 macOS 플랫폼의 Command Line Tool을 선택하여 Next를 누른다.



③ 템플릿을 선택하면 프로젝트의 이름과 회사 또는 팀의 ID를 입력한다. 여기서 ID는 주로 회사 또는 팀에서 사용하는 도메인을 뒤집어서 작성하지만, 프로그램을 배포하는 것이 아니라면 어떠한 것으로 지정해도 괜찮다.



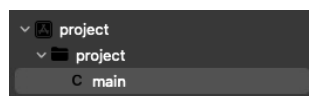
④ 이후 프로젝트가 저장될 위치를 지정한다. 중앙의 create Git repository on my Mac은 버전 관리로 깃을 사용한다는 의미이지만, 이 책에서는 깃에 대한 내용은 다루지 않으므로 체크를 풀고 생성한다.



⑤ 프로젝트를 생성하면 해당 프로젝트에서 프로그램을 만들 수 있도록 창이 뜬다. 이로써 Xcode에서 C 언어 프로그램을 만들기 위한 사전 준비가 끝났다.

2. Hello, World! 프로그램 작성하기

프로그램을 만들기 위한 프로젝트가 준비되었으니 이제 본격적인 C 언어로 프로그램을 만들어보도록 하자.



① 내비게이터에서 main이라는 이름의 C 언어 파일을 연다. 만약 아무런 파일이 없다면 프로젝트를 우클릭하여 **New File...**에서 C 언어 파일을 만든다.

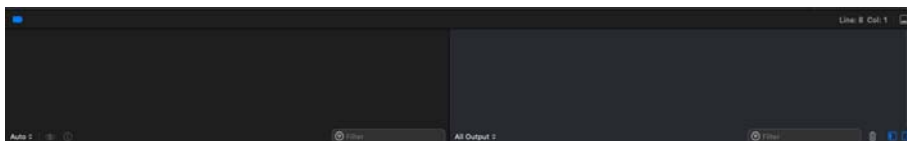
② main.c 파일에 다음의 코드를 작성한다.

손으로 익히는 코딩
<pre>#include <stdio.h> int main(void) { printf("Hello, World!"); return 0; }</pre>

실행 결과
Hello, World!

3. 프로그램 실행하기

앞서 코드를 작성한 main.c 파일은 C 언어 파일, 즉 사람이 이해하기 쉽도록 만들어진 고급 프로그래밍 언어로 작성된 파일이다. 그렇기에 이 파일에 담긴 내용을 컴퓨터가 이해하고 실행하기 위해서는 컴파일 과정이 필요하다. 이처럼 고급 프로그래밍 언어로 작성되어 컴파일되기 전의 파일을 **소스 파일**이라고 한다. 그리고 소스 파일을 컴파일하여 실제 컴퓨터가 실행할 수 있도록 기계어로 구성된 파일을 **실행 파일**이라고 한다. 앞서 만든 소스 파일을 실행시키기 위해선 컴파일 과정을 통해 실행 파일로 만드는 것이 선행되어야 한다.



① **Command + Shift + Y**를 눌러 커맨드 창을 연다. 커맨드 창에선 프로그램의 계산 결과를 확인하거나 프로그램이 계산해야 하는 값을 입력하는 역할을 수행한다.

DELL2023/02/12 22:15

기초 용어 정리
소스 파일
고급 프로그래밍 언어로 작성되어 컴파일되기 전의 파일

DELL2023/02/12 22:15

기초 용어 정리
실행 파일
컴퓨터가 실행할 수 있도록 기계어로 구성된 파일

```
Hello, World!Program ended with exit code: 0
```

② **Command + R**을 눌러 컴파일과 실행을 동시에 수행한다. 정상적으로 컴파일에 성공하였다면 앞서 실행한 커맨드 창에 Hello, World!라는 메시지를 볼 수 있다.

여러 예제를 실행하고자 하는 경우, 한 프로젝트에 main()이 한 개만 존재해야 한다. 만약 둘 이상의 main()이 있다면 프로그램이 실행되지 않으므로 새로운 프로젝트를 만들거나 main()이 존재하는 파일을 지워야 한다.

3절. C 언어의 구성 요소

핵심 키워드	서술문, 단일 서술문, 복합 서술문
---------------	---------------------

여기서는 무얼 배울까

프로그래밍 언어, 특히 고급 프로그래밍 언어는 그 언어라는 이름처럼 사람이 사용하는 자연어와 비교했을 때 비슷한 점이 많다. 그 쓰임새 때문에 자연어보다 더 형식적이고 체계적으로 설계되어있지만, 결국 사용하는 것은 사람이기 때문에 자연어에서 사용하던 익숙한 형태가 고급 프로그래밍 언어에서 나타나는 경우가 많다. 이번 절에서는 고급 프로그래밍 언어인 C 언어의 구조를 그와 대응되는 자연어와 함께 살펴보는 시간을 갖는다.

1. 서술문

고급 프로그래밍 언어는 사람이 사용하는 자연어와 비교하여 설명할 수 있다. 이러한 관점에서 볼 때 사람이 읽을 수 있도록 작성된 글은 컴퓨터 프로그램이고, 글이 여러 문단과 문장으로 이루어지듯 프로그램은 **서술문**이라고 하는 요소로 이루어진다. C 언어에서의 서술문은 단일 서술문과 복합 서술문으로 나눌 수 있다.

(1) 단일 서술문

단일 서술문은 하나의 문장에 대응되는 개념이다. 우리가 글이나 말을 할 때 완전한 문장을 단위로 이야기하듯 프로그램도 완전한 형식을 갖춘 단일 서술문의 단위를 사용한다. 따라서 **단일 서술문**은 프로그램의 최소 실행 단위이다.

```
printf("Hello, World!");
```

위의 코드는 하나의 단일 서술문을 의미한다. 이 서술문은 화면에 Hello, World!를 출력하라는 명령을 뜻한다. 단일 서술문은 문장의 온점과 비슷한 역할인 세미콜론(;)으로 끝나며, 세미콜론으로 단일 서술문을 구분한다.

DELL

2023/03/09 17:12

기초 용어 정리

서술문

C 언어를 서술하기 위한 기본 요소

DELL

2023/03/09 17:18

기초 용어 정리

단일 서술문

프로그램의 최소 실행 단위로 단일 서술문의 끝은 세미콜론으로 표시한다.


```
printf(
    "Hello, World!"
);
```

이처럼 길이가 길어지거나 가독성을 위해 여러 줄에 걸쳐서 단일 서술문을 작성할 수는 있지만 **단일 서술문을 나누는 기준은 세미콜론**이기에 여전히 하나의 단일 서술문이다.

(2) 복합 서술문

여러 문장이 묶여서 하나의 문단을 이루는 것과 비슷하게 서술문들이 모여 복합 서술문을 이룬다. **복합 서술문** 또는 블록은 0개 이상의 서술문을 묶은 서술문이다. 단일 서술문이 최소 실행 단위인 짧은 코드이기에 복합 서술문은 이러한 코드의 묶음이라고 볼 수 있다.

```
{
    printf("Hello, World!");
    system("pause");
    return 0;
}
```

위의 코드는 3개의 서술문을 묶은 복합 서술문의 예이다. 위의 코드처럼 복합 서술문은 어떤 서술문을 묶을지를 중괄호 안에 넣어서 나타낸다.

```
{
}
```

위의 복합 서술문은 내부에 서술문이 존재하지는 않는다. 복합 서술문은 0개 이상의 서술문을 묶은 것이기에 서술문이 없음에도 오류가 발생하지 않는다.

```
{ { } }
```

복합 서술문 안에 또 다른 복합 서술문이 들어갈 수도 있다. 위의 예는 바깥에 복합 서술문 하나를 포함하는 복합 서술문이 있고, 안에는 **서술문을 하나도 묶지 않은 빈 복합 서술문**이 있는 것이다.

(3) 주석

C 언어를 비롯한 여러 프로그래밍 언어에서는 그 프로그램에 대한 설명을 달 수 있는 주석이 있다. 주석은 컴파일 과정에서 무시되는 문장이지만 프로그램을 작성할 때 이 코드가 어떠한 기능을 하는지, 추가해야 하는 부분이 무엇인지 등 설명을 남기기 위해서 주로 사용된다.

DELL 2023/08/31 15:14

TIP

C 언어는 세미콜론에 엄격합니다. 단일 서술문이 세미콜론으로 끝나지 않는다면 오류가 발생합니다.

DELL 2023/03/09 17:22

기초 용어 정리

복합 서술문

0개 이상의 서술문을 묶은 서술문, 코드의 묶음

DELL 2023/08/31 15:15

TIP

일반적으로 위처럼 서술문을 작성하지는 않습니다. 이렇게 사용해도 오류가 없다는 것을 설명하기 위한 예입니다.

DELL 2023/07/10 12:09

기초 용어 정리

주석

코드에 대한 설명으로 컴파일할 때는 무시되지만 프로그램을 작성하는 과정 중에 확인할 수 있음

코드 소개
// 주석

한 줄로 주석을 작성할 때는 // 이후에 주석으로 달 문장을 작성한다.

```
#include <stdio.h>

int main()
{
    // comment 1
    printf("Hello, World!"); // comment 2
}
```

슬래시부터 같은 줄의 오른쪽에 해당하는 모든 글자는 주석으로 처리되어 컴파일할 때 무시된다. 슬래시 왼쪽에 있는 글자는 주석이 아니기에 정상적으로 컴파일 과정에서 번역된다.

코드 소개
/* 주석 */

두 줄 이상으로 주석을 작성할 때에는 /*와 */ 사이에 주석으로 사용할 문장을 작성한다.

```
#include <stdio.h>

int main()
{
    printf("Hello, World!");
    /*
        comment 1 ...
        comment 2 ...
    */
}
```

/*와 */ 사이에 있는 글자들은 주석으로 처리되어 컴파일 과정에서 무시되고, 그 외에는 컴파일러가 번역한다. 적절한 주석의 사용은 프로그램의 가독성을 높인다는 장점이 있다.

2. Hello, World! 프로그램 분석하기

C 언어를 서술하기 위한 요소인 서술문에 대해서 알아보았으니 이제 처음 작성한 Hello, World! 프로그램이 어떻게 구성되어 있는지를 이해할 시간이다. 프로그램 전체를 모두 이해할 수는 없지만, 각각의 부분이 어떠한 역할을 파악하는 정도면 충분하다.

```
#include <stdio.h>

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

지난 절에서 만들었던 Hello, World! 프로그램이다. 이 코드를 한 줄씩 보면서 어떤 역할을 하는지 알아보자.

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

이 프로그램에서 표준 입출력을 사용하겠다는 의미이다. 여기서 표준 입력이란 대개 사용자의 키보드 입력을, 표준 출력이란 콘솔에 글자를 출력하는 것을 의미한다. Hello, World! 프로그램은 해당 문장을 화면에 표준 출력을 하는 프로그램이기에 이 부분이 필요하다.

```
int main(void)
```

프로그램의 본문이 여기서부터 시작이라는 의미이다. 이 위에 있는 #include는 이 프로그램이 어떤 기능을 사용하는지를 알려주는 부분이기에 본문과는 큰 관련이 없다. 따라서 프로그램이 어떤 연산을 어떤 절차에 맞게 수행하는지는 본문, 즉 main에서 작성한다. 이러한 프로그램의 본문을 특히 **main 함수**라는 이름으로 부른다.

```
{
    printf("Hello, World!");
    return 0;
}
```

main 함수가 어디서부터 어디까지인지, 즉 프로그램의 본문의 범위를 나타내야 한다. 이는 int main(void) 뒤에 복합 서술문의 형태로 나타나게 되며 중괄호 안에 프로그램의 본문에

DELL

2023/07/03 23:43

기초 용어 정리

main 함수

C 언어 프로그램의 본문

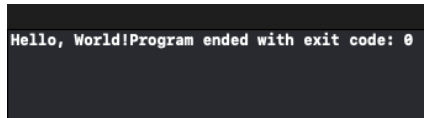
해당하는 코드를 작성한다. Hello, World! 프로그램의 본문은 2개의 단일 서술문으로 이루어진다.

```
printf("Hello, World!");
```

콘솔에 Hello, World! 라는 문장을 출력하는 부분이다. 큰따옴표로 묶인 부분에 다른 글을 적으면 Hello, World! 대신 그 문장이 출력된다.

```
return 0;
```

프로그램이 해야 하는 모든 작업이 끝난 이후에 프로그램이 끝났음을 알려주는 부분이다. 여기서 특히 0이라는 숫자가 있는데, 이는 프로그램이 정상적으로 종료가 되었음을 나타낸다. 만약 0이 아닌 숫자가 있다면 프로그램에 문제가 생겨 정상 동작이 되지 않았음을 나타낸다.

A screenshot of a terminal window with a dark background. The text 'Hello, World!Program ended with exit code: 0' is displayed in a light-colored monospace font.

Xcode에서 Hello, World! 프로그램을 작성한 이후에 실행하면 이처럼 맨 마지막에 0이라는 숫자가 출력된다. 만약 프로그램에서 return 0; 이 아닌 return 1; 로 수정하고 실행하면 0 대신 1이라는 숫자가 출력된다. 이를 통해 프로그램을 개발하거나 사용할 때 어디서 문제가 생겼는지 대략 파악할 수 있다.

예러에서 배우기
프로그래밍 언어를 이용하여 프로그램을 처음 만들었을 때 쉽게 실수할 수 있는 부분은 문법을 제대로 따르지 않았을 때이다. 사람은 어느 정도 문법에 맞지 않는 문장을 보아도 해석하는 데에 큰 문제가 없지만, 컴퓨터는 사소한 문법의 실수만으로도 오류가 발생할 수 있다. 아래는 Hello, World! 프로그램에서 발생할 수 있는 몇 가지 실수를 소개한다.
<pre>int Main(void)</pre>
(LNK1120) 1개의 확인할 수 없는 외부 참조입니다.
프로그램의 본문을 나타내는 부분은 그 이름이 main이어야 한다. 이는 다른 단어여서는 안 되고 대소문자 또한 모두 소문자여야 한다. 만약 다른 이름을 사용하면 오류가 발생한다.

<code>printf('Hello, World!');</code>
(C2015) 상수에 문자가 너무 많습니다.
printf()에서 문장을 출력하기 위해선 출력하고자 하는 문장을 큰따옴표로 묶어야 한다. 작은따옴표로 묶는다면 오류가 발생한다.
<code>return 0</code>
(E0065) ':'가 필요합니다.
단일 서술문은 항상 세미콜론으로 끝나야 한다. 세미콜론으로 끝나지 않고 곧바로 다른 서술문이나 프로그램 코드가 끝나면 세미콜론이 필요하다는 오류가 발생한다.
이 외에도 Hello, World! 프로그램에서 발생할 수 있는 오류는 많다. 그렇지만 발생할 수 있는 오류 대부분은 써야 하는 단어가 아닌 다른 단어를 쓰거나 잘못된 기호를 쓰는 등 문법과 관련된 부분으로 발생하므로 자신이 작성한 코드가 제대로 작성되었는지 확인하는 연습이 필요하다.

연습문제
OX 퀴즈
C 언어를 개발하기 위해선 Xcode를 필수적으로 사용해야 한다.
Xcode는 C 언어가 아닌 다른 프로그래밍 언어를 지원한다.
약술형
통합 개발 환경이란?
모두 고르기
다음 중 문제가 없는 서술문의 형태를 모두 고르시오.
<code>printf("Hello, World!");</code>
<code>scanf("%f", &height)</code>
<pre> { int a = 5; int b = 10; int c = a + b; } </pre>

```
{
    { { } } {
        { }
    } }
}
```

```
{
    printf("Hello, C!");
}
```

단답형

다음 프로그램은 몇 개의 단일 서술문과 몇 개의 복합 서술문으로 구성되어 있는가?

```
#include <stdio.h>

int main(void)
{
    {
        printf("A");
        printf("B");
        {
            printf("C");
            {
                printf("D");
            }
        }
    }
    printf("E");
}
```

한 장으로 요약 정리

키워드로 정리하기

- 프로그래밍을 돕기 위한 여러 프로그램을 한 데 모아놓은 프로그램을 통합 개발 환경이라고 하며, Xcode는 맥에서 C 언어 개발을 위한 통합 개발 환경이다.
- Xcode에는 컴파일러, 에디터, 디버거, 전처리기, 링커, 프로파일러 등 C 언어 개발을 하는 데에 도움이 되는 다양한 프로그램이 내장되어 있다.
- 고급 프로그래밍 언어로 작성된 파일을 **소스 파일**, 소스 파일을 컴파일하여 컴퓨터가 실행할 수 있는 상태의 파일을 **실행 파일**이라고 한다.
- **콘솔**은 화면에 글자들을 출력하고 사용자로부터 키보드 입력을 받을 수 있는 창을 의미한다.
- C 언어의 최소 실행 단위는 **단일 서술문**으로 세미콜론으로 구분하며, 0개 이상의 단일 서술문을 중괄호로 묶어 **복합 서술문**으로 만들 수 있다.

예제 훑아보기

```
#include <stdio.h> // (1)

int main(void) // (2)
{
    printf("Hello, World!"); // (3)
    return 0; // (4)
}
```

- (1) 이 프로그램에서 표준 입출력을 사용하겠다는 의미이다.
- (2) 프로그램 본문의 시작이다.
- (3) 콘솔에 Hello, World라는 글자를 출력한다.
- (4) 프로그램이 문제없이 실행되었음을 알린다.

연습 문제 해설과 답

OX 퀴즈

C 언어를 개발하기 위해선 Xcode를 필수적으로 사용해야 한다.

- X, C 언어를 개발하기 위한 프로그램은 다양하므로 다른 프로그램을 사용해도 괜찮다.

Xcode는 C 언어가 아닌 다른 프로그래밍 언어를 지원한다.

- O, Xcode는 C++, Swift, Objective-C 등의 프로그래밍 언어도 지원한다.

약술형

통합 개발 환경이란?

- 프로그래밍에 필요한 컴파일러, 에디터, 런처, 디버거 등을 모은 프로그램이다.

모두 고르기

다음 중 문제가 없는 서술문의 형태를 모두 고르시오.

```
printf("Hello, World!");
```

O, 완전한 형태의 단일 서술문이다.

```
scanf("%f", &height)
```

X, 단일 서술문은 세미콜론으로 끝나야 한다.

```
{  
    int a = 5;  
    int b = 10;  
    int c = a + b;  
}
```

O, 3개의 단일 서술문을 묶은 복합 서술문이다.

```
{  
    { { } } {  
        { }  
    } }  
}
```

X, 중괄호는 한 쌍이 복합 서술문을 이루지만 여기서는 닫는 중괄호 하나가 쌍이 없다.

```
{  
    printf("Hello, C!");;  
}
```

O, 2개의 단일 서술문을 묶은 복합 서술문입니다. 마지막 세미콜론은 빈 단일 서술문을 의미합니다.

단답형

다음 프로그램은 몇 개의 단일 서술문과 몇 개의 복합 서술문으로 구성되어 있는가?

```
#include <stdio.h>

int main(void)
{
    {
        printf("A");
        printf("B");
        {
            printf("C");
            {
                printf("D");
            }
        }
    }
    printf("E");
}
```

중괄호가 4쌍이므로 복합 서술문 4개, 세미콜론이 5개이므로 단일 서술문 5개이다.

[1쪽] [메모:6] DELL 2023/03/13 22:30

기초 용어 정리

통합 개발 환경

프로그래머를 돕기 위한 에디터, 프로파일러, 디버거, 컴파일러 등의 프로그램을 하나에 모아둔 프로그램과 개발 환경

[1쪽] [메모:12] DELL 2023/09/18 21:05

코멘트

다른 프로그램을 사용한다 하더라도 Apple 환경의 소프트웨어를 개발 및 컴파일하기 위해서는 Xcode가 필요합니다.

[3쪽] [메모:9] DELL 2023/08/31 14:58

TIP

화면의 구성은 버전과 설정에 따라 상이할 수 있습니다.

[7쪽] [메모:1] DELL 2023/02/12 22:15

기초 용어 정리

소스 파일

고급 프로그래밍 언어로 작성되어 컴파일되기 전의 파일

[7쪽] [메모:2] DELL 2023/02/12 22:15

기초 용어 정리

실행 파일

컴퓨터가 실행할 수 있도록 기계어로 구성된 파일

[8쪽] [메모:3] DELL 2023/03/09 17:12

기초 용어 정리

서술문

C 언어를 서술하기 위한 기본 요소

[8쪽] [메모:4] DELL 2023/03/09 17:18

기초 용어 정리

단일 서술문

프로그램의 최소 실행 단위로 단일 서술문의 끝은 세미콜론으로 표시한다.

[9쪽] [메모:10] DELL 2023/08/31 15:14

TIP

C 언어는 세미콜론에 엄격합니다. 단일 서술문이 세미콜론으로 끝나지 않는다면 오류가 발생합니다.

[9쪽] [메모:5] DELL 2023/03/09 17:22

기초 용어 정리

복합 서술문

0개 이상의 서술문을 묶은 서술문, 코드의 묶음

[9쪽] [메모:11] DELL 2023/08/31 15:15

TIP

일반적으로 위처럼 서술문을 작성하지는 않습니다. 이렇게 사용해도 오류가 없다는 것을 설명하기 위한 예입니다.

[9쪽] [메모:8] DELL 2023/07/10 12:09

기초 용어 정리

주석

코드에 대한 설명으로 컴파일할 때는 무시되지만 프로그램을 작성하는 과정 중에 확인할 수 있음

[11쪽] [메모:7] DELL 2023/07/03 23:43

기초 용어 정리

main 함수

C 언어 프로그램의 본문