

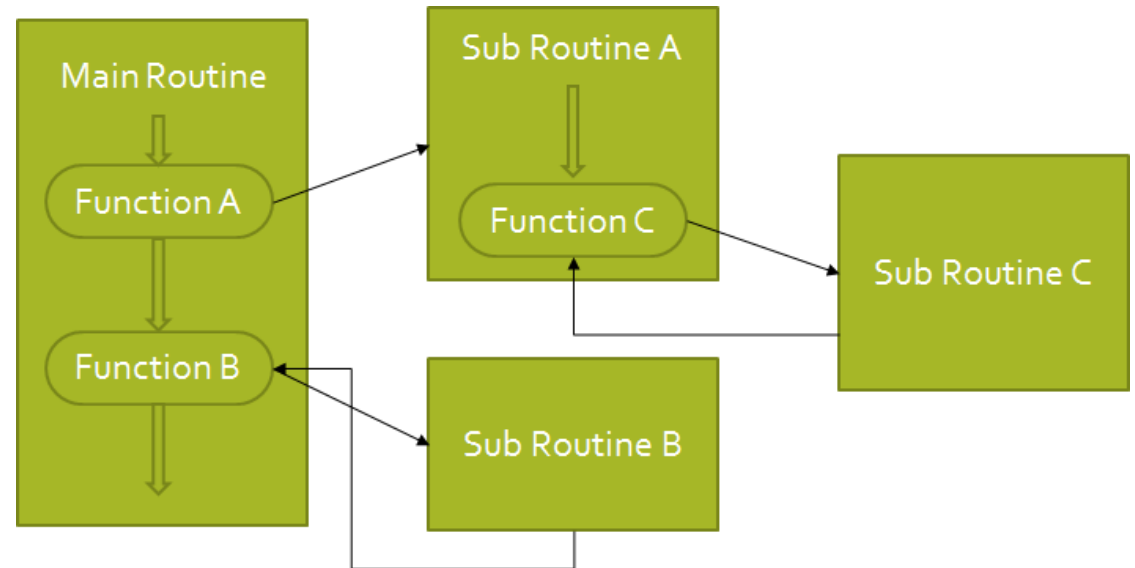
# C# 양보와 반복

---

# 루틴

- 루틴(Routine)

- 프로그램이 수행하는 일련 처리 순서
- 처리의 우선 순위가 존재
- 메인 루틴
  - 프로그램의 수행 시 가장 상위의 처리 순서
  - C 언어의 main.c 함수의 처리 순서
- 서브 루틴
  - 메인 루틴에서 호출한 함수의 처리 순서



메인 루틴과 서브루틴의 개념

# 코루틴

- **코루틴(Coroutine)**

- 프로그램에서 서로 다른 루틴이 처리 권한을 주고 받으며 동시에 작업을 수행
- 쓰레드는 실제로 프로세스가 두 개의 작업을 동시에 처리하지만 코루틴은 서로 권한을 주고 받으며 작업을 처리하기 때문에 다른 개념임
- 동시에 서로 다른 루틴이 동시에 수행되는 효과

# yield 키워드

- yield(양보)
  - 메소드나 연산자가 반복호출 됨을 의미하는 키워드
  - 현재 루틴의 수행을 멈추고 대기 상태에 들어감

```
yield return <expression>;  
yield break;
```

# 반복기

- 반복자(Iterable, Iterator)
  - 반복의 대상이 될수 있는 행위
  - 내부의 데이터를 순방향으로만 접근할 수있도록 지원

# 유니티에서 사용한 반복기

코루틴용 데이터	엔진이 수행하는 기능
<code>yield return null</code>	다음 프레임까지 대기
<code>yield return new WaitForSeconds(float)</code>	지정된 초 만큼 대기
<code>yield return new WaitForFixedUpdate()</code>	다음 물리 프레임까지 대기
<code>yield return new WaitForEndOfFrame()</code>	모든 렌더링작업이 끝날 때까지 대기
<code>yield return StartCoroutine(string)</code>	다른 코루틴이 끝날 때까지 대기
<code>yield return new WWW(string)</code>	웹 통신 작업이 끝날 때까지 대기
<code>yield return new AsyncOperation</code>	비동기 작업이 끝날 때까지 대기 ( 씬로딩 )

유니티에서 사용가능한 `yield return` 반복기