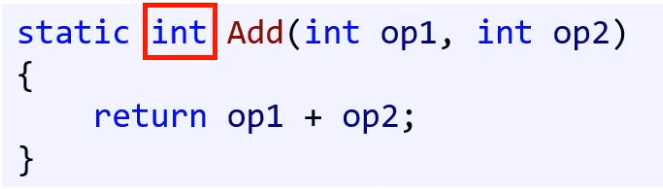
**1. 함수**

* **프로그래밍 세계에서의 함수**
* 함수에 **입력**이 들어가면 어떤 결과가 **출력**됨
* 중복되는 코드를 줄이는데 사용됨  
  - 수정 및 관리가 용이
* 자주 사용하는 코드의 재활용성이 올라감  
  - 함수 호출로 처리 가능
* **함수의 정의**

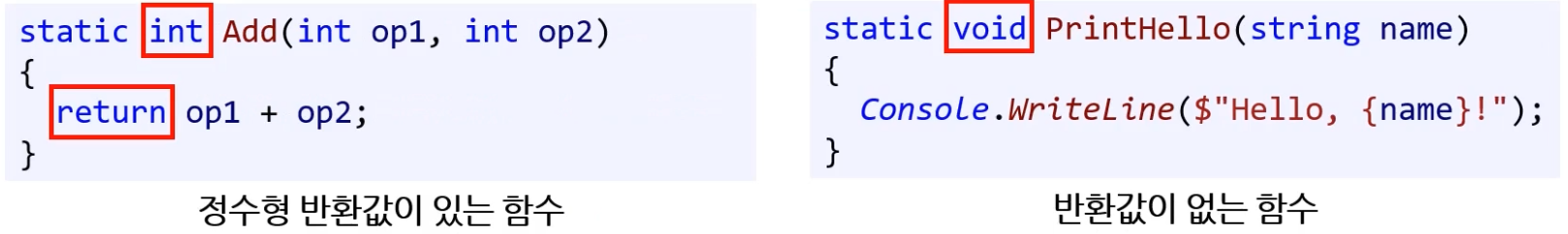
텍스트이(가) 표시된 사진

자동 생성된 설명

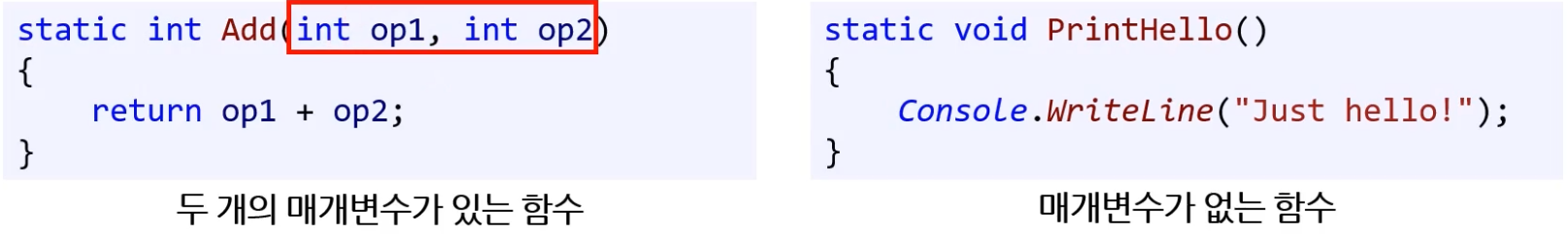
* 함수를 정의할 때는 다음의 것들을 포함해야 함  
  - 함수 시그니쳐  
   - 함수명(필수), 매개변수 목록(선택)  
  - 반환형(필수)  
  - 함수 바디(필수)
* **함수 – 반환형**



* 프로그래밍 세계에서의 **출력(output)**
* 반환형은 반드시 선언해야 함  
  - 안 하면 컴파일 오류
* 반환형이 없을 수도 있음  
  - 반환 값이 없다면 반환형에 void를 사용
* 반환형이 void가 아닐 경우  
  - 함수 바디에 return 키워드를 이용해 데이터를 반환해야 함  
  - 데이터를 반환하지 않으면 컴파일 오류



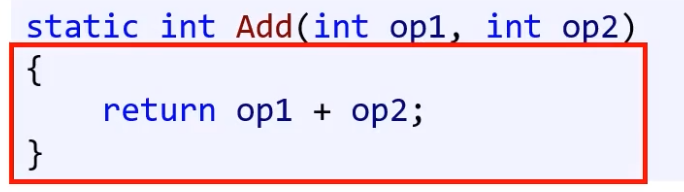
* **함수 – 매개변수**



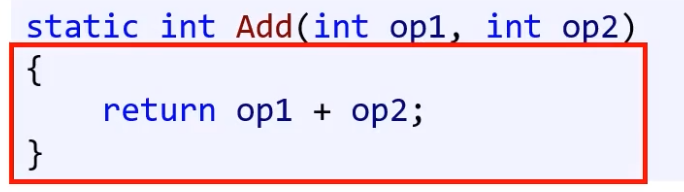
* 프로그래밍 세계에서의 입력
* 이를 매개변수(parameter)라고 부름
* 기본 자료형 뿐만 아니라 배열도 매개변수로 사용 가능
* 매개변수는 필수가 아님  
  - 함수는 0개 이상의 매개변수를 가짐
* 다른 용어로 인자(argument)라고도 함
* 엄밀히 말하면 매개변수와 인자는 다른 것  
  - 매개변수: 함수를 정의할 때 함수의 입력 값을 선언하는 것을 말함  
  - 인자: 함수를 호출할 때 함수로 전달되는 실제 값을 말함



* **함수 – 함수 바디**



* 함수의 기능을 구현한 코드 블록
* 중괄호를 이용하여 범위 표시
* 함수가 void가 아닌 반환형을 가지고 있다면 반드시 return 키워드를 사용하여 데이터를 반환해야 함
* **함수 – 함수명**



* 함수의 이름
* 엄밀히 말하면 함수 이름은 어떻게 짓든지 상관 없음
* 그러나 함수가 어떤 기능을 가졌는지 알 수 있게 지으면 가독성 증가  
  - 입력을 넣으면 내부구조(작동원리)를 알지 못하더라도 어떤 출력이 나오는지 알 수 있어야 함
* **함수는 반드시 블랙박스여야 함**
* 가독성을 위해
* 함수는 함수 호출자에 대해서 알아서는 안 됨
* 호출자가 함수 내부를 알 필요가 없게 **함수명을 명확하게** 지어야 함

**2. 코딩표준: 함수 명명 방법**

* **함수 - 정확하게 어떤 기능을 하는지 알려주는 단어 사용**
* 동사로 시작할 것 -> 함수는 행동(Action)이니까
* 제일 첫 글자는 대문자로
* 여러 단어를 연결한다면 두 번째 단어부터는 첫 글자를 대문자로  
  - 파스칼 표기법
* **변수 - 정확하게 어떤 정보를 담는지 알려주는 단어를 사용**
* 명사를 사용할 것 -> 함수는 행동(Action)이니까
* 제일 첫 글자는 소문자로
* 여러 단어를 연결한다면 두 번째 단어부터는 첫 글자를 대문자로  
  - 카멜 표기법
* **선조건과 후조건**
* 함수가 무슨 일을 하는지에 대한 약속
* 선조건  
  - 함수 실행 시작 전에 참으로 가정한 조건 – ex) Divide() 함수는 분모가 0이 아니어야 한다.  
  - 함수 이름이나 매개변수로 유추 가능하나, 부족하면 주석으로 추가 설명
* 후조건  
  - 함수 실행 후에 보장되는 조건 – ex) 두 정수를 더하면 정수의 결과가 나온다.  
  - 보통 함수 이름과 반환형으로 유추 가능
* 선조건을 만족하지 못하면 후조건을 보장할 수 없음
* **함수 시그니쳐가 약속하는 것**텍스트이(가) 표시된 사진

  자동 생성된 설명
* 함수명  
  - “두 수를 나눈 결과를 얻을 수 있는 함수구나”
* 매개변수  
  - “첫 번째 매개변수가 분자고, 두 번째 매개변수는 분모구나”
* 반환형  
  - “부동소수점형이 반환되는 구나”

**3. 함수와 범위**

* **프로그래밍 세계에서의 함수**
* 함수에 **입력**이 들어가면 어떤 결과가 **출력**됨
* 중복되는 코드를 줄이는데 사용됨  
  - 수정 및 관리가 용이
* 자주 사용하는 코드의 재활용성이 올라감  
  - 함수 호출로 처리 가능