

그런데 이렇게 간단한 한개의 함수만을 만들어 쓴다면 하나의 cpp 파일에 이정도로 만들어도 되겠지만 실제 프로그래밍을 할 때에는 수많은 함수를 만들게 됩니다. 저렇게 만들어 쓴다면 가독성에도 매우 좋지 못하고 비효율적이 되죠. 때문에 .h 헤더 파일을 새로 만들게 되는 것입니다. 헤더파일과 헤더파일을 설명하는 cpp파일을 만들어보도록 하겠습니다.

main.cpp	head.h	head.cpp
<pre>#include "head.h" int main(void) { int result = sum (3, 4); return 0; }</pre>	<pre>int sum (int a , int b);</pre>	<pre>int sum (int a , int b) { return (a + b); }</pre>

하나의 cpp파일로 되어있던 프로그램을 3개의 cpp 파일로 나누었습니다. 기능별로 확실하게 파일이 나누어져있으며 기능을 매우 정확하게 보기 좋습니다. 또한 파일이 따로따로 나누어져 있기 때문에 여러사람이 분업을 해서 만들기 더욱더 좋아졌습니다. 헤더파일과 메인파일을 제외하고 만든 세번째 cpp파일은 헤더파일에 선언되어있는 함수의 기능을 구현한 파일입니다. 여러가지 이유로 헤더파일에 선언한 함수들은 그 기능을 다른 cpp에 구현합니다. 절대로 head.h안에 적어주면 안됩니다. 이유는 나중에 컴파일러가 이런 파일들을 어떻게 처리하는지 알아보면서 설명드리겠습니다. 또한 우리가 만든 헤더파일을 불러올때는 include " "를 사용합니다.

extern으로 선언하면.
 ↳ include를 하면, 해당 헤더 파일 내 함수 선언문이 스킵으로 복사된다!

이렇게 헤더파일을 만들면 이점이 많습니다. 마치 메뉴판과 같습니다. 메인에서 실제 프로그램을 작동하도록 만드는 사람은 단순히 기능을 새로 추가할때 헤더파일만을 불러오면 됩니다. 헤더파일에 선언되어있는 함수들을 그냥 불러오면 되기 때문에 매우 편리합니다. 또한 기능을 만드는 사람은 단순히 헤더파일과 그에따른 cpp파일을 만든다면 기능을 만들수 있는 것이기때문에 여러가지 프로그램을 만드는데 사용할수 있습니다.