C与C++中的逗号

March 21, 2017

在C 与 C++中，逗号有两层含义：

1. 逗号充当运算符。

   逗号运算符为一元运算符，先计算第一个操作数并舍弃之，然后计算第二个操作数并返回该值。逗号运算符具有最低优先级，并且是一个顺序点。

   ```
   /* comma as an operator */
   int i = (5, 10);  /* 10 is assigned to i*/
   int j = (f1(), f2());  /* f1() is called (evaluated) first followed by f2().
                          The returned value of f2() is assigned to j */
   ```

2. 逗号充当分隔符

   逗号作为分隔符，通常用于函数调用与定义，函数宏，变量声明，enum声明以及结构体中。

   ```
   /* comma as a separator */
   int a = 1, b = 2;
   void fun(x, y);
   ```

```
/* Comma acts as a separator here and doesn't enforce any sequence.
    Therefore, either f1() or f2() can be called first */
void fun(f1(), f2());
```

```
// comma1.c:
#include<stdio.h>
int main()
{
  int x = 10;
  int y = 15;
  printf("%d\n", (x, y));
  return 0;
}
```

```
// comma2.c:
#include<stdio.h>
int main()
{
  int x = 10;
  int y = (x++, ++x);
  printf("%d", y);
  return 0;
}
```

```
// comma3.c:
#include<stdio.h>
int main()
{
```

```
  int x = 10, y;
  // The following is equavalent to y = x++
  y = (x++, printf("x␣=␣%d\n", x), ++x, printf("x␣=␣%d\n", x), x++);
  // Note that last expression is evaluated
  // but side effect is not updated to y
  printf("y␣=␣%d\n", y);
  printf("x␣=␣%d\n", x);
  return 0;
}
```