

09. 파이썬과 COM

여러분들은 '1 부 파이썬 프로그래밍편'을 통해 파이썬의 기본 문법을 배웠습니다. 특히 8 장에서는 파이썬을 이용하여 간단한 프로그램도 직접 만들어 보았습니다. 그러나 지금까지 배웠던 내용은 아직 실제 만들어야 할 알고리즘 트레이딩과는 다소 거리가 있었습니다.

2 부 '증권사 API 기초편'에서는 알고리즘 트레이딩 시스템의 기본이 되는 증권사 API(Application Programming Interface)를 배워보도록 하겠습니다. 여기서 API란 파이썬의 함수나 메서드와 같은 것입니다. 증권사가 제공하는 API를 사용하면 시세 정보를 얻거나 프로그램을 통한 매매와 같은 기본 동작을 수행할 수 있습니다.

그러나 여기에도 한 가지 문제가 있습니다. 국내 증권사들이 제공하는 API를 이용하면 프로그램을 쉽게 작성할 수 있는 것은 맞지만, 국내 증권사의 API는 파이썬으로 구현되어 있지 않습니다. 파이썬으로 작성된 함수 또는 클래스라면 앞서 배운 내용을 바탕으로 어떻게 이용해볼 수는 있을 것 같은데, 다른 프로그래밍 언어로 작성된 함수들은 파이썬에서 어떻게 호출할 수 있을까요?

위의 질문에 대한 답이 될 수 있는 기술이 바로 이번 장에서 다룰 마이크로소프트(사)의 COM(Component Object Model)입니다. 이번 장을 통해 COM에 대한 간단히 알아보고 파이썬에서 COM을 사용하는 방법을 배워봅시다.

1) COM 과 Python

COM 이란 마이크로소프트에서 개발한 기술입니다. Component Object Model 이라는 용어가 의미하듯이 여러 컴포넌트 객체를 이용하여 프로그램을 개발하는 모델을 의미합니다. 컨테이너를 이용하여 집을 지으면 쉽고 빨리 집을 지을 수 있는 것처럼 프로그램을 개발할 때도 특정 기능을 위해 개발된 Component Object 들을 이용하면 프로그램을 더 쉽고 빠르게 개발할 수 있습니다.

참고로 컴포넌트 객체란 파이썬으로 치면 클래스로부터 만들어지는 객체와 같은 것이라 이해하면 되겠습니다. 여기까지만 본다면 COM 이라는 기술의 개념은 기존의 객체 지향 프로그래밍과 크게 다르지 않습니다. 그러나 COM 은 여기서 더 나아가 프로그래밍 언어와 상관없이 개발된 객체를 사용할 수 있도록 해줍니다. 예를 들어, COM 을 이용하면 C++과 같은 프로그래밍 언어에서 개발된 객체도 파이썬에서 사용할 수 있게 됩니다.

별써 클래스, 객체, 인스턴스라는 용어가 기억이 나지 않는 분들을 위하여 간단한 예를 통해서 설명해 드리겠습니다. 대신증권은 C/C++로 구현된 CpStockCode 클래스를 제공하는데 이 클래스와 비슷한 기능을 제공하는 클래스를 파이썬으로 구현해 보겠습니다. 참고로 대신증권에서 실제로 제공하는 API 는 10 장에서 보다 자세히 다룰 예정입니다.

```
class CpStockCode:
    def __init__(self):
        self.stocks = {'유한양행': 'A000100'}
    def GetCount(self):
        return len(self.stocks)
    def NameToCode(self, name):
        return self.stocks[name]
```

CpStockCode 클래스는 생성자 (`__init__`)와 `GetCount`, `NameToCode` 라는 메서드를 갖고 있습니다. 여기서 `GetCount` 메서드는 주식 종목수를 리턴해주는 메서드고, `NameToCode` 메서드는 종목명을 입력하면 해당 종목에 대한 종목 코드를 리턴해주는 메서드입니다.

참고로 그림 9.1 과 같이 KRX 웹페이지(www.krx.co.kr)에서 제공하는 주식 종목 검색기에서 종목명을 입력하면 종목코드를 확인할 수 있습니다. 이와 비슷하게 증권사의 API 에도 종목명을 입력하면 종목 코드를 반환해주는 메서드가 존재합니다. 여러분들은 이러한 API 를 프로그램 내에서 호출함으로써 종목 코드를 프로그램 내에서도 쉽게 얻을 수 있습니다. www.krx.co.kr/pop_kor/popup/JHPKOR13008_12.jsp



그림 9.1 KRX 종목 검색기

6장에서 클래스를 정의하는 것은 붕어빵을 만드는 과정에서 붕어빵 틀을 만드는 것과 비슷하다고 설명했습니다. 붕어빵 틀에 반죽을 부은 후 붕어빵을 굽는 것처럼 여러분은 클래스(붕어빵 틀)를 이용하여 인스턴스 (붕어빵)를 만들 수 있었습니다.

```
instCpStockCode = CpStockCode()
```

위 코드는 CpStockCode 클래스를 이용하여 instCpStockCode 객체를 생성한 예입니다. 즉, instCpStockCode 객체는 CpStockCode 클래스의 인스턴스입니다.

참고로 객체와 인스턴스라는 용어는 같은 의미를 갖기 때문에 혼용해서 사용해도 큰 무리는 없습니다. 다만, instCpStockCode와 같이 클래스를 통해서 생성된 인스턴스 자체를 지칭할 때는 '무슨 무슨 객체'라는 표현을 주로 사용합니다. 이와 달리 'instCpStockCode 객체는 CpStockCode 클래스의 인스턴스'라는 표현에서처럼 객체가 어떤 클래스로부터 생성된 것인지를 의도해서 말하고자 할 때는 인스턴스라는 용어를 사용합니다.

객체와 인스턴스에 대한 차이는 이쯤 설명을 하기로 하고 instCpStockCode 객체를 사용하여 GetCount와 NameToCode 메서드를 호출해 보겠습니다. GetCount 메서드는 self 이외에는 인자가 없으므로 호출 시에는 인자를 넘겨주지 않아도 됩니다. NameToCode에는 종목 코드를 확인하고자 하는 종목의 종목 이름을

입력하면 됩니다. 참고로 NameToCode 메서드는 현재 '유한양행'이라는 종목에 대해서만 정상적으로 종목 코드를 리턴해 주도록 구현되어 있습니다.

```
print(instCpStockCode.GetCount())
```

```
print(instCpStockCode.NameToCode('유한양행'))
```

앞서 파이썬으로 구현된 CpStockCode 클래스에 대한 인스턴스를 생성할 때는 단순히 'instCpStockCode = CpStockCode()'와 같이 클래스 이름에 ()를 붙여 주었습니다. 그러나 파이썬이 아닌 다른 프로그래밍 언어로 구현된 클래스에 대한 인스턴스는 파이썬에서 어떻게 생성할 수 있을까요?

파이썬에서 다른 프로그래밍 언어로 작성된 COM 객체를 생성하기 위해서는 win32com.client 라는 모듈의 Dispatch() 메서드를 사용하면 됩니다. 예를 들어, 마이크로소프트의 인터넷 익스플로러에 대한 객체를 생성하기 위해서는 다음과 같이 구현하면 됩니다.

```
import win32com.client
```

```
explore = win32com.client.Dispatch("InternetExplorer.Application")
```

```
explore.Visible = True
```

먼저 win32com.client 라는 모듈을 사용하기 위해 모듈을 import 합니다. 그리고 win32com.client 라는 모듈 내의 Dispatch 함수를 호출하는데 이때 함수의 인자로 "InternetExplorer.Application"라는 문자열을 사용했습니다. Dispatch 함수가 호출되면 객체가 생성되는데 explore 라는 변수가 그림 9.2 와 같이 생성된 COM 객체를 바인딩합니다.



그림 9.2 생성된 COM 객체에 대한 바인딩

COM 객체가 생성되었으니 COM 객체의 메서드를 호출하거나 속성을 변경할 수 있습니다. 'explore.Visible = True'는 Visible 이라는 속성의 값을 True 로 변경한 것입니다. 이 속성값을 True 로 변경하면 그림 9.3 과 같이 화면에 익스플로러가 나타납니다.

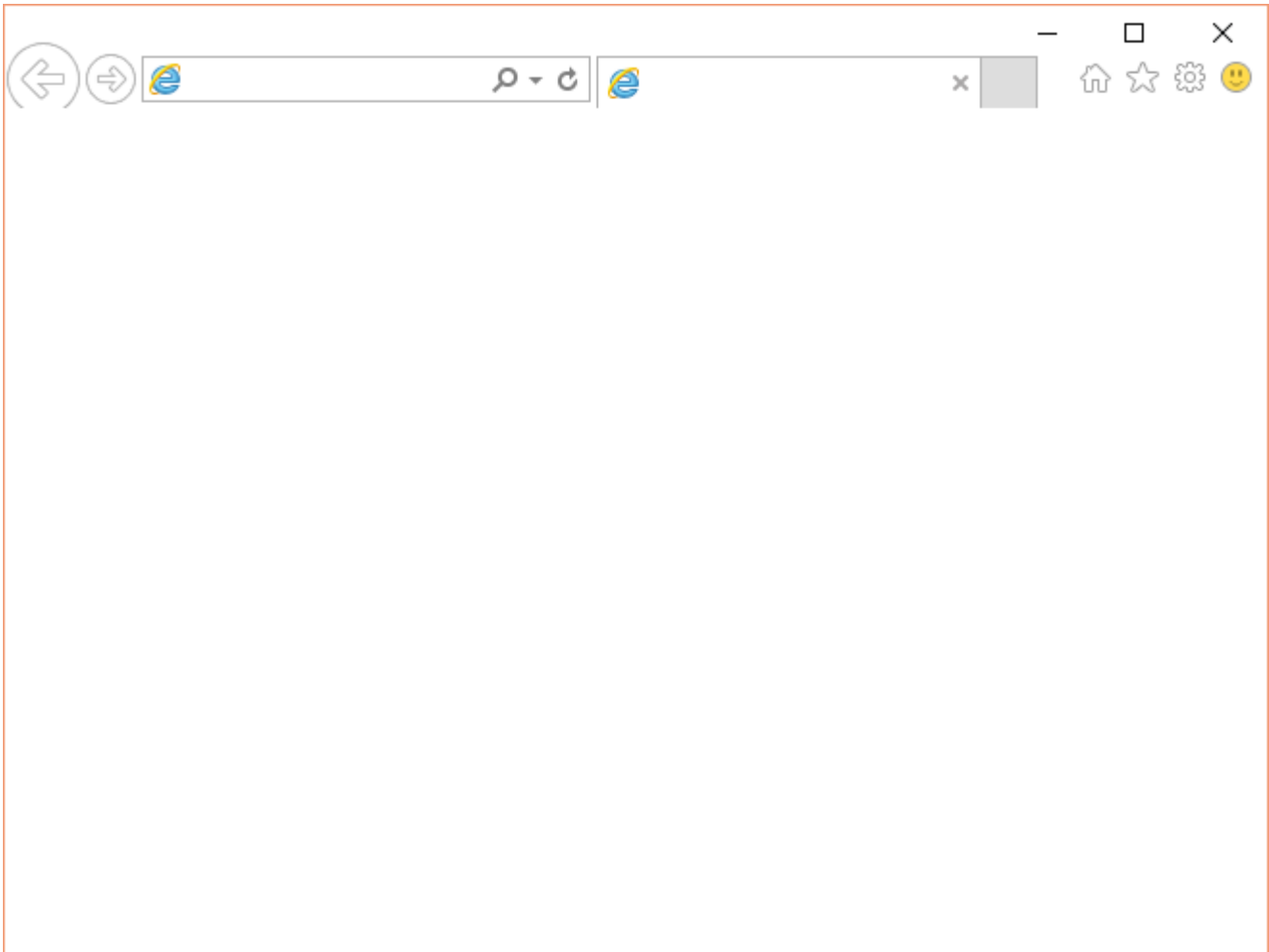


그림 9.3 COM 객체를 이용한 익스플로러 실행

어떤가요? 마이크로소프트의 COM 이라는 기술을 사용해서 인터넷 익스플로러를 파이썬 코드에서도 실행할 수 있다니 신기하지 않습니까?

컴퓨터에 마이크로소프트의 워드가 설치된 분들은 다음 코드도 실행해보기 바랍니다.

```
import win32com.client
```

```
word = win32com.client.Dispatch("Word.Application")
```

```
word.Visible = True
```

이번에는 마이크로소프트 워드 프로그램이 자동으로 실행되는 것을 확인할 수 있습니다. 앞서 실행했던 코드와 마찬가지로 win32com.client 모듈의 Dispatch 함수를 이용하여 "Word.Application"에 대한 COM 객체를 생성한 후 객체의 Visible 속성값을 True 로 변경했습니다.

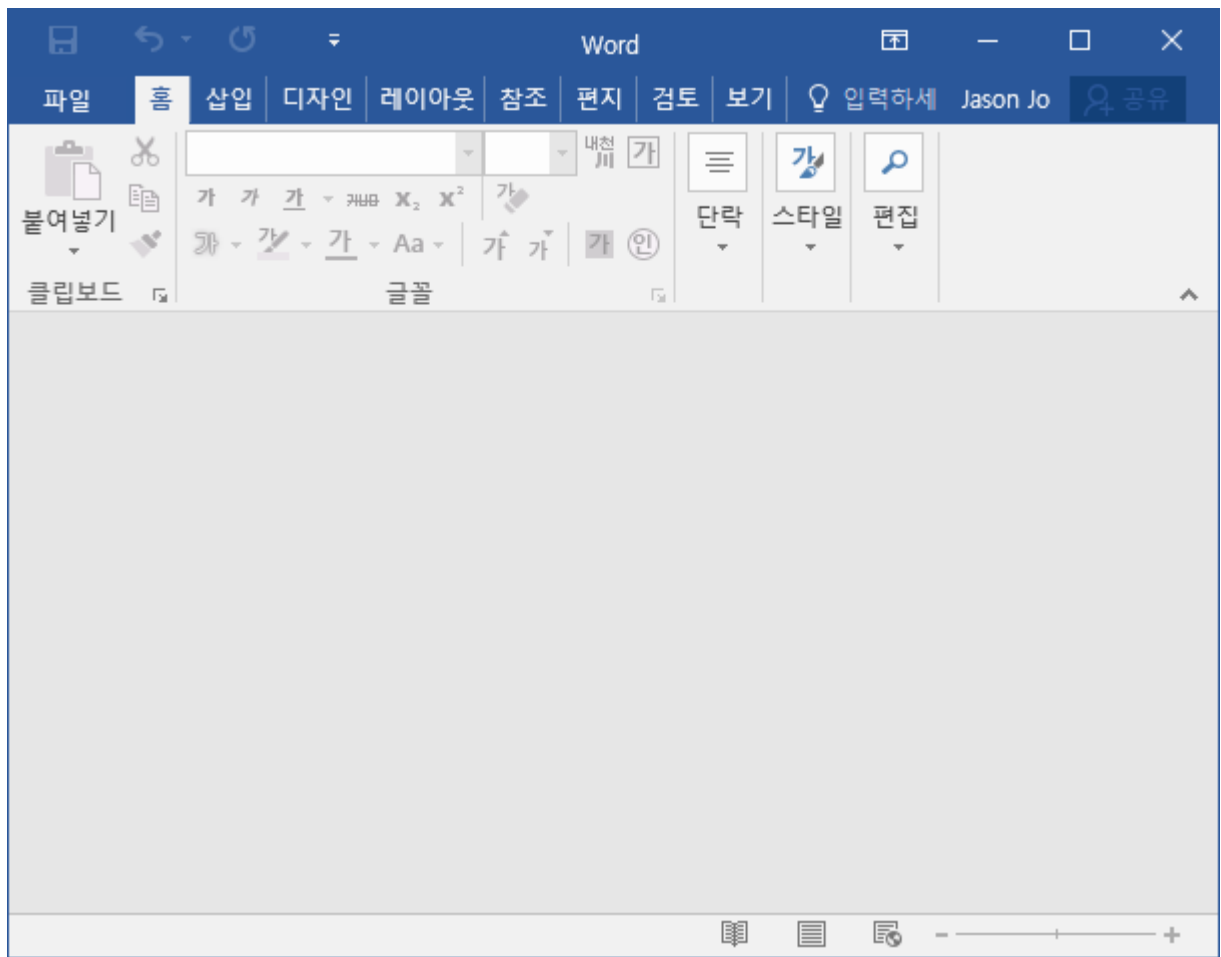


그림 9.4 COM 객체를 이용한 Word 실행

2) 파이썬으로 엑셀 다루기

여러분은 9.1 절에서 파이썬에서 COM 객체를 생성하는 방법과 생성된 객체를 통해 메서드를 호출하거나 속성값을 변경하는 것을 배웠습니다. 이번 절에서는 알고리즘 트레이딩을 위한 도구로 많이 사용되고 있는 마이크로소프트사의 엑셀을 파이썬에서 다루보도록 하겠습니다.

앞서 워드나 인터넷 익스플로러에 대한 COM 객체를 생성한 것과 마찬가지로 엑셀에 대한 COM 객체를 생성해보기 바랍니다. 물론 다음 코드가 정상적으로 실행되려면 여러분의 PC 에 마이크로소프트사의 엑셀이 미리 설치되어 있어야 합니다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

엑셀도 워드나 익스플로러와 마찬가지로 Visible 이라는 이름의 속성이 있습니다. 다음과 같이 Visible 속성의 값을 True 로 변경하면 화면에 엑셀이 나타납니다. 사실 Dispatch 함수를 호출하여 Excel.Application 에 대한 인스턴스를 생성했을 때부터 엑셀은 이미 실행 되어 있었습니다. 다만 화면에 보이지는 않는 상태였을 뿐입니다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

```
excel.Visible = True
```

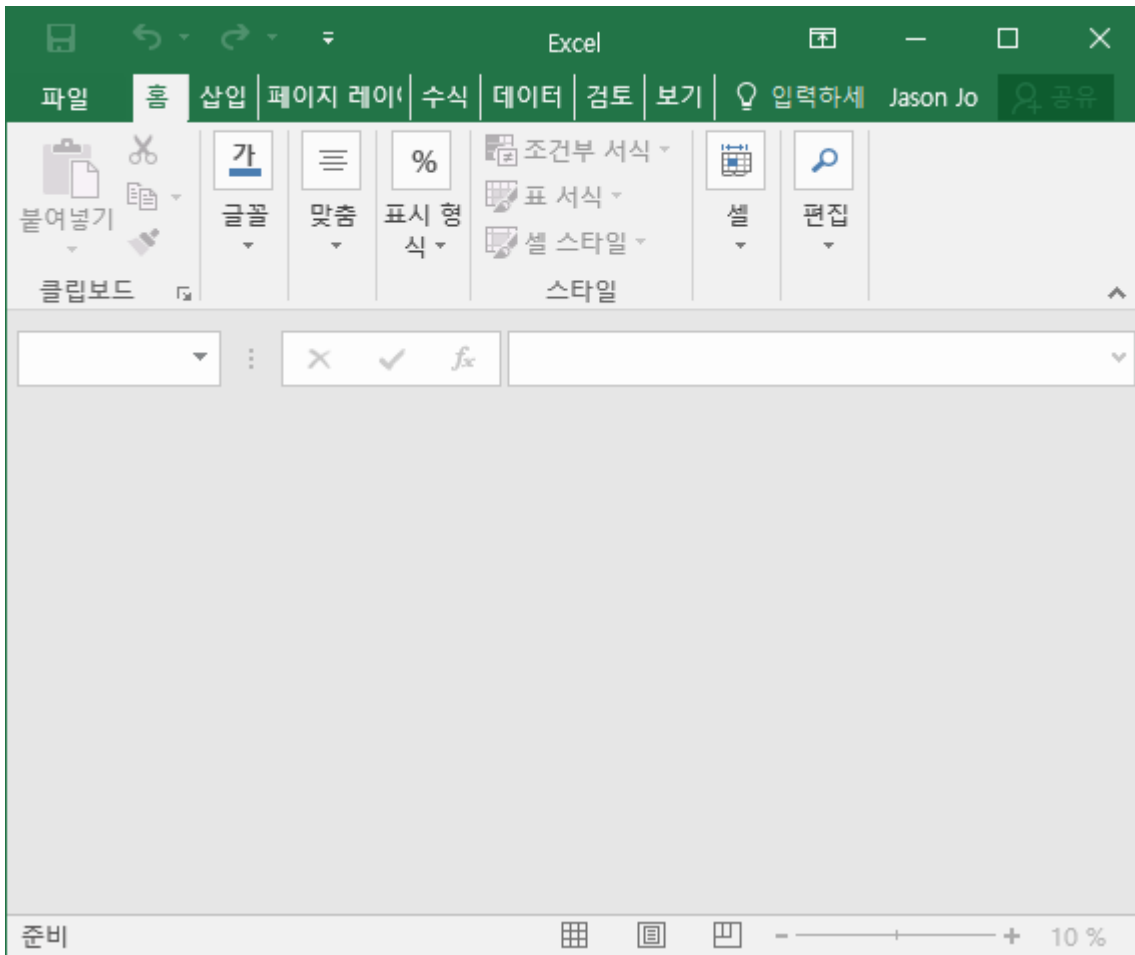


그림 9.5 엑셀 실행 화면

엑셀이 화면에 나타났으니 이제 엑셀에 글자를 출력해 봅시다. 아마도 엑셀을 사용할 때 항상 Worksheet 에서부터 시작했을 것입니다. 엑셀에는 여러 개의 Worksheet 를 포함하는 상위 개념으로 Workbook 이 있습니다. 즉, 엑셀 프로그램을 실행시키면 Workbook 이 자동으로 생성되고 'Sheet1'이라는 이름의 Worksheet 가 자동 생성이 됩니다.

COM 을 사용하여 파이썬 코드로 엑셀을 직접 컨트롤 할 때도 Workbook 과 Worksheet 의 관계를 고려해야 합니다. 먼저 Workbook 을 추가한 후 해당 Workbook 에서 'Sheet1'이라는 Worksheet 를 선택합니다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

```
excel.Visible = True
```

```
wb = excel.Workbooks.Add()
```

```
ws = wb.Worksheets("Sheet1")
```

위 코드를 실행시키면 그림 9.6 과 같이 우리가 평상시에 사용하는 엑셀의 '새 통합 문서' 상태가 됩니다.

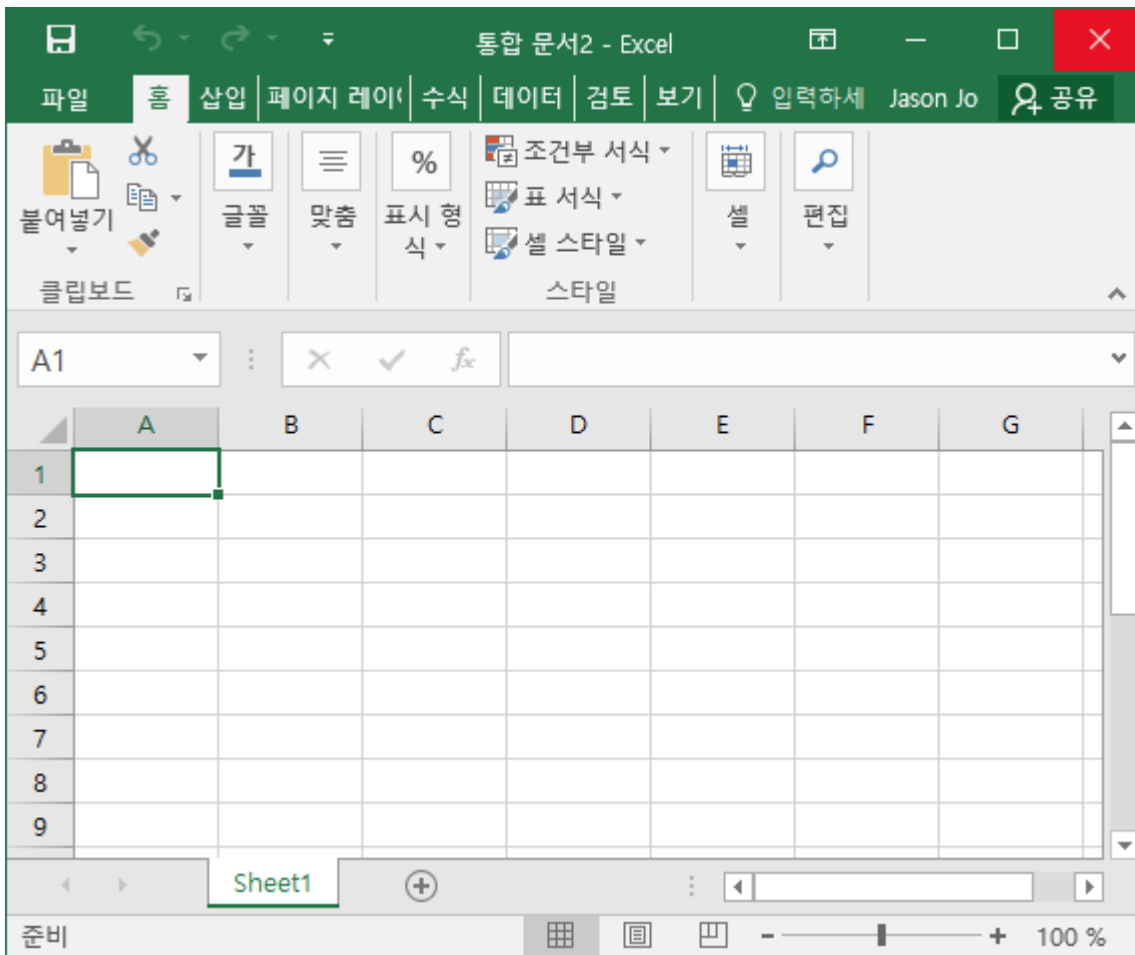


그림 9.6 파이썬을 이용한 엑셀 Workbook 생성

엑셀의 Worksheet에는 많은 수의 셀(Cell)이 있는데 각 셀은 행과 열의 인덱스 값을 통해서 구분됩니다. Worksheet의 특정 셀에 값을 입력하기 위해서는 Cells 속성을 사용하면 됩니다. Cells이라는 키워드 다음에는 행과 열에 대한 인덱스 값을 적어 줌으로써 특정 셀에 접근할 수 있습니다.

예를 들어, 다음 코드에서 'Cells(1, 1)'이라는 표현은 첫 번째 행, 첫 번째 열에 해당하는 셀을 의미합니다. 즉, 그림 9.6 Sheet1에서 A1 셀이 됩니다. 셀이 선택됐다면 셀의 Value 속성에 'hello world'를 입력합니다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

```
excel.Visible = True
```

```
wb = excel.Workbooks.Add()
```

```
ws = wb.Worksheets("Sheet1")
```

```
ws.Cells(1, 1).Value = "hello world"
```

위 코드를 실행하면 그림 9.7과 같이 'Sheet 1'의 A1 Cell에 'hello world'라는 문자열이 입력된 것을 확인할 수 있습니다.

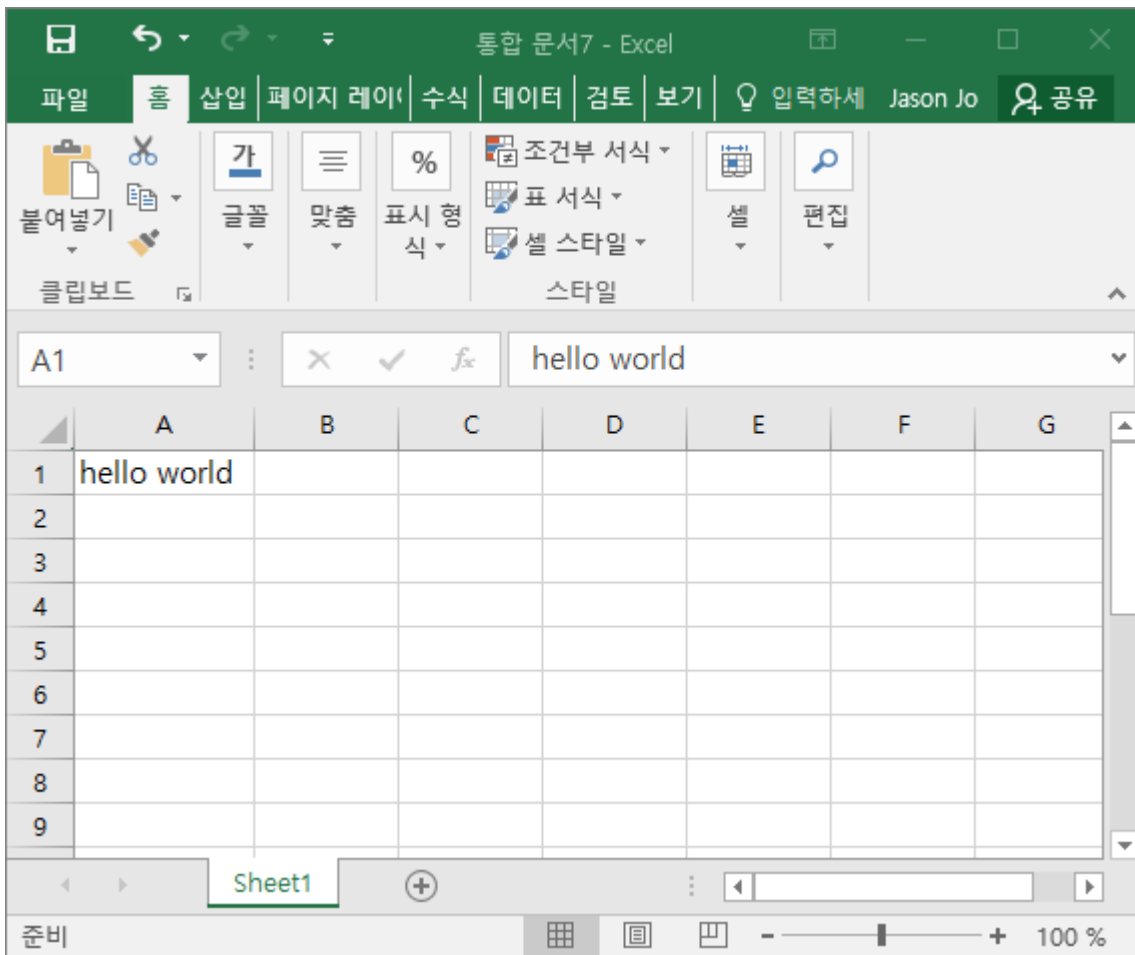


그림 9.7 A1 Cell 에 값 입력하기

엑셀에 정상적으로 데이터가 입력됐으니 해당 파일을 하드 디스크에 저장하고 프로그램을 종료해 봅시다. 엑셀 프로그램 자체에도 '다른 이름으로 저장'이라는 저장 메뉴가 있듯이 엑셀 COM 객체에도 SaveAs 라는 메서드가 있습니다. 다음과 같이 SaveAs 메서드에 파일을 저장할 경로를 인자로 넘겨주면 해당 경로에 엑셀 파일이 저장됩니다. SaveAs 메서드 호출을 통해 정상적으로 파일로 저장했다면 Quit 메서드를 호출하여 엑셀을 종료시켜 줍니다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

```
excel.Visible = True
```

```
wb = excel.Workbooks.Add()
```

```
ws = wb.Worksheets("Sheet1")
```

```
ws.Cells(1, 1).Value = "hello world"
```

```
wb.SaveAs('c:\\Users\\Jason\\Desktop\\test.xlsx')
```

```
excel.Quit()
```

위 코드를 실행하면 그림 9.8 과 같이 윈도우의 바탕화면에 엑셀 파일이 저장됩니다. 참고로 SaveAs 함수의 인자로 입력되는 저장 경로는 여러분의 환경에 맞춰 적절히 변경해 줘야만 합니다.

이처럼 파이썬에서 COM 기술을 사용하면 엑셀 프로그램을 직접 실행시킬 필요 없이 엑셀 실행, 데이터 입력, 저장 및 종료가 모두 프로그램 내에서 가능합니다. 평상시에도 엑셀로 작업을 많이 한다면 파이썬을 이용하여 자동으로 엑셀 문서를 만드는 프로그램을 만들어 보기 바랍니다.



그림 9.8 바탕화면에 저장된 엑셀 파일

2-1) 엑셀 파일 읽기

이번에는 하드 디스크에 이미 존재하는 엑셀 파일로부터 데이터를 읽어 오는 파이썬 프로그램을 작성해보겠습니다. 프로그램에서 사용할 샘플 엑셀 파일을 만들기 위해 그림 9.9 와 같이 엑셀을 실행한 후 A1 셀에 python 을 입력합니다.

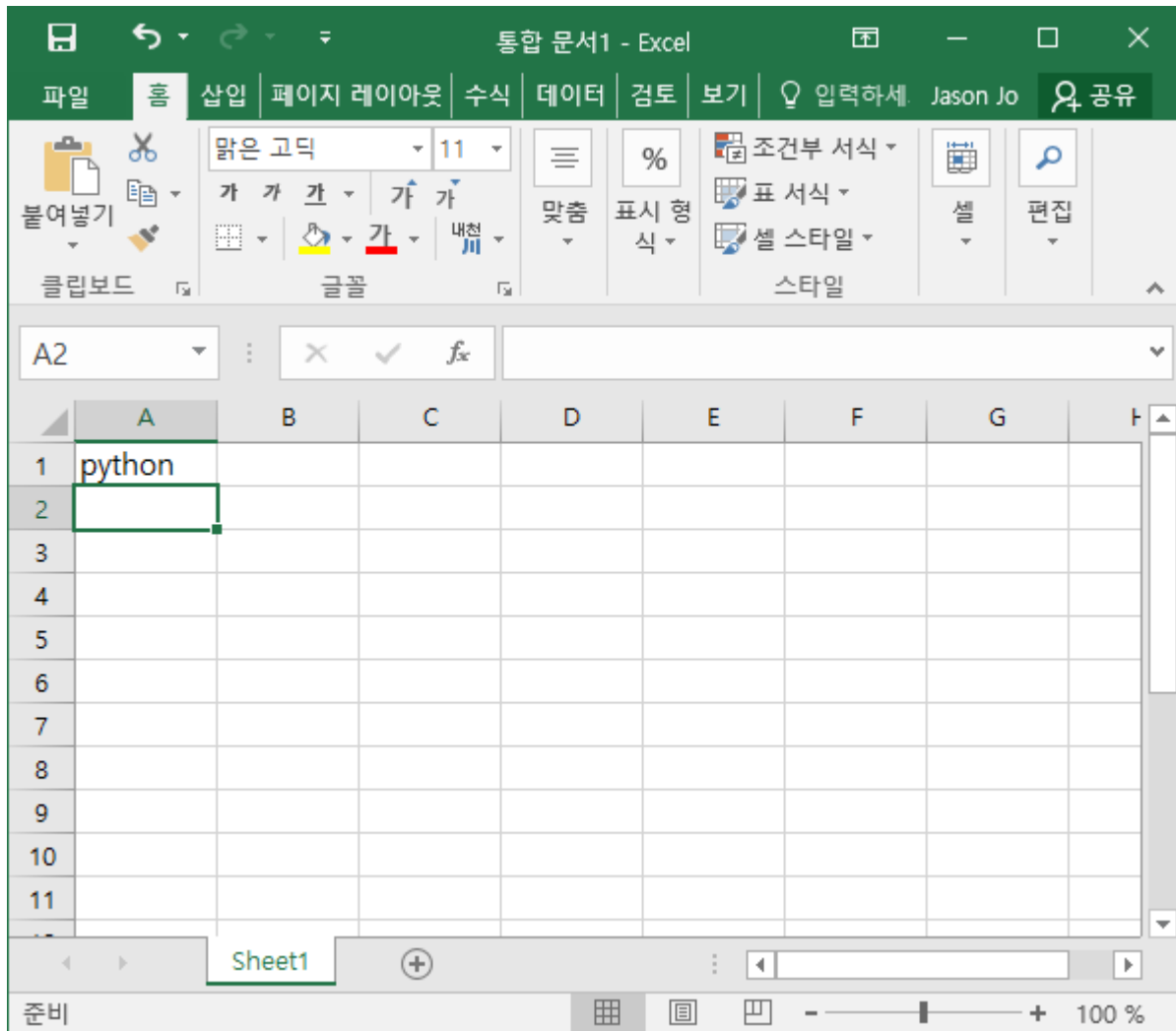


그림 9.9 입력 엑셀 파일 생성

엑셀의 메뉴에서 파일 -> 다른 이름으로 저장 기능을 사용하여 윈도우의 바탕화면에 input.xlsx 라는 이름으로 저장하기 바랍니다.

파이썬에서 하드디스크에 있는 엑셀 파일을 읽는 것도 앞서 엑셀 파일을 쓰는 것과 비슷합니다. 다만 Workbook 과 Worksheet 이 이미 존재하기 때문에 기존에 있는 것 중 선택해서 사용해야 합니다. 먼저 다음코드를 실행해 보기 바랍니다. Open 함수의 인자는 여러분의 윈도우 바탕화면의 경로 및 저장한 파일의 이름에 맞추어 적절히 수정해줘야 합니다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

```
excel.Visible = True
```

```
wb = excel.Workbooks.Open('C:\\Users\\Jason\\Desktop\\input.xlsx')
```

```
ws = wb.ActiveSheet
```

```
print(ws.Cells(1,1).Value)
```

```
excel.Quit()
```

위 코드에서 Open 함수가 호출되면 바로 Workbook 객체를 얻는 것을 확인할 수 있습니다. Workbook 객체를 통해서 다시 Worksheet 객체를 얻을 수 있습니다. 이때 Worksheet 이름('Sheet1')을 사용하여 여러 Worksheet 중 하나를 선택할 수도 있고 위 코드와 현재 Active 상태인 Worksheet 를 선택할 수도 있습니다.

Worksheet 을 얻었다면 Worksheet 을 통해 해당 Worksheet 에 있는 셀에 접근해서 저장된 데이터 값을 읽어올 수 있습니다.

2-2) 셀에 컬러 입히기

이번에는 엑셀 Worksheet 에서 특정 셀의 색상을 변경해 보겠습니다. 엑셀 COM 객체에는 엑셀 프로그램상에서 우리가 사용했던 기능이 거의 그대로 메서드나 속성 형태로 제공됩니다. 먼저 앞서 사용했던 입력 파일인 input.xlsx 파일을 COM 객체를 통해 읽어 봅시다.

```
import win32com.client
```

```
excel = win32com.client.Dispatch("Excel.Application")
```

```
excel.Visible = True
```

```
wb = excel.Workbooks.Open('C:\\Users\\Jason\\Desktop\\input.xlsx')
```

```
ws = wb.ActiveSheet
```

B1 셀과 C1 셀에 'is'와 'good'이라는 문자열을 출력해 봅시다. 이를 위해서 다음과 같이 Cells 와 Range 를 이용했습니다. 그리고 C1 셀의 색상을 변경하기 위하여 Interior 객체의 ColorIndex 속성값을 변경했습니다.

```
ws.Cells(1,2).Value = "is"
```

```
ws.Range("C1").Value = "good"
```

```
ws.Range("C1").Interior.ColorIndex = 10
```

지금까지 작성한 코드를 실행하면 그림 9.10 과 같이 B1 과 C1 셀에 데이터가 입력되고 C1 셀의 색상이 변경된 것을 확인할 수 있습니다.

그림 9.10 엑셀 cell 색상 변경

참고로 ColorIndex 속성은 0~56 까지의 값을 가질 수 있으며 숫자별로 색상이 정해져 있는데 색상 값은 마이크로소프트사에서 제공하는 MSDN 을 참조하면 됩니다. 저자 또한 이러한 정보는 그림 9.11 과 같이 그때그때 Google 검색을 통해서 찾고 있습니다.

검색결과 약 264,000개 (0.41초)

관련검색: 엑셀 vba application 엑셀 application run excel app excel program
excel program for mac

Color Palette and the 56 Excel ColorIndex Colors

dmcritchie.mvps.org/excel/colors.htm ▼ 이 페이지 번역하기

2011. 4. 6. - Excel Color Index, coloring of fonts, cell interiors. ... Each Microsoft Excel workbook has a palette of 56 colors that you can apply to cells, fonts, ...

Adding Color to Excel 2007 Worksheets by Using the ...

[https://msdn.microsoft.com/.../cc296089\(v=office.12...](https://msdn.microsoft.com/.../cc296089(v=office.12...) ▼ 이 페이지 번역하기

Summary: Learn how to use the ColorIndex property to programmatically ... In these instances, Excel tries to randomly apply a color that corresponds to the ...
이 페이지를 15. 12. 5에 방문했습니다.

ColorIndex Property - MSDN - Microsoft

<https://msdn.microsoft.com/en-us/.../ff840443.aspx> ▼ 이 페이지 번역하기

Excel · Excel VBA reference · Microsoft Graph Visual Basic reference · Properties · Application Property · ApplyPictToEnd Property The following illustration shows the color-index values in the default color palette. colorin_ZA06050819 ...

그림 9.11 검색을 통한 COM 객체의 속성 값 확인

앞서 Worksheet의 셀을 선택할 때 Cells와 Range라는 속성을 각각 이용했는데 둘 다 셀을 선택하는 데 사용됩니다. 다만 Range 속성은 이름 자체가 의미하는 것처럼 여러 셀을 선택할 때 주로 사용합니다. 예를 들어 A2:C2 범위의 셀에 대한 색상을 노란색으로 변경하기 위해서는 다음과 같이 작성하면 됩니다.

```
ws.Range("A2:C2").Interior.ColorIndex = 27
```

지금까지 COM을 사용하여 마이크로소프트의 인터넷 익스플로러, 워드, 엑셀 등을 조작해 보았습니다. 이를 통해 COM을 사용하면 다른 프로그램 언어로 작성된 객체도 쉽게 사용할 수 있음을 확인했습니다. 국내 증권사 API들도 COM을 사용하면 지금까지 설명한 것과 유사한 방식으로 사용할 수 있습니다.