

엑셀
X
파이썬
업무 자동화



매크로,
VBA는
이제 낡았다!

카네히로 카즈미 지음
서수환 옮김

Excel



Python

인사이트

엑셀 × 파이썬 업무 자동화



EXCEL × PYTHON SAISOKU SHIGOTOJUTSU

by Kazumi Kanehiro

Copyright © 2019 by Kazumi Kanehiro.

All rights reserved.

Originally published in Japan by Nikkei Business Publications, Inc.

Korean translation rights arranged with Nikkei Business Publications, Inc. through Botong Agency.

이 책의 한국어판 번역권은 Botong Agency를 통한 저작권자와의 독점 계약으로 인사이트가 소유합니다.
신 저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단전재와 무단복제를 금합니다.

엑셀 × 파이썬 업무 자동화 : 매크로, VBA는 이제 낡았다!

초판 PDF 1.0 2020년 12월 22일 지은이 카네히로 카즈미 옮긴이 서수환 펴낸이 한기성 펴낸곳 인사이트 편집 신승준
등록번호 제2002-000049호 등록일자 2002년 2월 19일 주소 서울특별시 마포구 연남로5길 19-5 전화 02-322-5143
팩스 02-3143-5579 블로그 <http://blog.insightbook.co.kr> 이메일 insight@insightbook.co.kr ISBN 978-89-6626-292-2

엑셀 × 파이썬 업무 자동화



매크로, VBA는 이제 낡았다!

카네히로 카즈미 지음 | 서수환 옮김

인사이트

저자 서문	viii
역자 서문	ix

1장 파이썬이란? 1

01 파이썬의 특징	3
언어 사양이 단순하다	3
들여쓰기가 문법이다	4
라이브러리가 풍부해서 다양한 용도로 사용할 수 있다	6
02 VBA와 파이썬의 차이점	8
입문자를 위한 VBA 추가 설명	11
03 파이썬 프로그래밍 환경 설정하기	13
파이썬 설치하기	13
파이썬을 실행해서 동작 확인	20
비주얼 스튜디오 코드 설치	21
프로그래밍이 편리해지는 확장 기능 추가	23

2장 파이썬과 프로그래밍 기본 31

01 파이썬 문법	33
변수와 자료형	33
변수명 짓기	37
산술 연산자	38
비교 연산자, 복합 대입 연산자, 논리 연산자	39
함수	43
자신만의 함수도 작성 가능	45
객체 지향	49

3장 엑셀 워크시트 다루기

53

01 엑셀 데이터를 한꺼번에 읽는 프로그램	55
지정한 범위의 셀을 읽어서 목록 작성	58
02 파이썬 핵심 정리	73
외부 라이브러리 설치하기	73
if 문으로 조건 분기	76
같은 처리를 반복할 때 사용하는 for 구문	80
엑셀 파일에서 읽을 범위	86
데이터의 한 형태인 리스트(list)	90
CSV 형식으로 출력하기	93
엑셀 파일이 서버나 NAS에 있을 경우	95
실행했더니 ‘퍼미션 에러?’	96

4장 집계

101

01 매출 집계, 수주 교차 집계 프로그램	103
키와 값을 쌍으로 갖는 데이터형인 사전(dictionary)	106
파이썬으로 매출 목록표에서 담당자별, 거래처별로 집계하기	107
수주 목록표를 상품 분류와 사이즈로 교차 집계하는 프로그램	119
02 파이썬 핵심 정리	128
자료 구조	128
2차원 리스트의 초기화	130
엑셀 함수 포함하기	132

5장 서식 설정과 인쇄 141

01 집계표에 서식 설정하는 예제 프로그램	143
02 파이썬 핵심 정리	149
import 문법 구조	150
행과 열 숨기기	152
셀 서식 설정 방법 정리	152
raw 문자열	157
테두리 스타일	157
셀 범위를 지정하는 방법	160
조건부 서식 설정하기	164
색조 설정법	168

6장 차트 171

01 차트를 작성하는 예제 프로그램	173
막대형 차트	179
누적 막대형 차트	184
깎은선형 차트	187
영역형 차트	189
원형 차트	192
방사형 차트	195
거품형 차트	197
프로그램 실행 방법	201

01 엑셀 문서를 PDF로 만드는 프로그램	210
COM을 이용해 파이썬으로 엑셀 조작하기	210
win32com 패키지 설치	211
매출전표 여러 장을 하나로 모아 PDF로 만들기	214
파이썬으로 PDF 레이아웃 정하기	219
후기	236
찾아보기	238

여러분, 이 책을 선택해 주셔서 감사합니다.

독자 여러분은 분명 여기 서문부터 읽기 시작하셨겠지요. 하지만 책을 쓰다 보면 서문은 마지막에 쓰는 게 보통입니다. 이 책의 서문도 다른 부분을 모두 쓴 다음, 이전에 썼던 것을 다시 돌아보면서 쓰고 있답니다.

이 책은 사무직 종사자라면 누구에게나 익숙한 엑셀을, 어떻게 파이썬으로 프로그래밍할 수 있는지 소개하고 있습니다. 파이썬 프로그래밍 입문의 문턱을 낮춰 독자들이 손쉽게 사용할 수 있도록 닷케이 BP 편집자와 몇 번이 고 머리를 맞대며 고민한 끝에 만들게 된 책입니다.

프로그래머인 제게도 엑셀 기능은 무척 복잡해 보였습니다. 그런데 책을 쓰기 위해 파이썬 코드를 작성하다 보니 엑셀이 어떤 구조로 되어 있는지 알게 되었고, 그래서 엑셀 기능도 한결 이해하기 쉬워졌습니다.

비즈니스 현장에서는 수많은 자료가 엑셀로 만들어집니다. 엑셀만으로 이런 자료를 전부 처리하려면 복잡한 함수를 쓰거나 VBA 매크로를 작성해야 하는데 다음과 같은 문제가 생기기도 합니다.

- 어디서 어떻게 하고 있는지 알 수 없다. 워크시트에 있는 것이 함수 또는 엑셀의 기능인지 VBA의 기능인지 경계선이 불분명하다.
- 사람마다 엑셀에 대한 능력 차이가 있어서 인수인계가 어렵다.

이럴 때 자료는 엑셀로 만들고 복잡한 처리는 파이썬으로 프로그래밍한다면 업무 효율을 훨씬 극대화할 수 있습니다. 이 책이 그런 기회가 되었으면 합니다.

그러면 부디 끝까지 읽어 주시고 괜찮다면 감상을 부탁드립니다.

첫 손주로 Twins가 태어난 날에
카네히로 카즈미(金宏和實)

‘아무것도 안 하고 싶다.

이미 아무것도 안 하고 있지만, 더 격렬하게 아무것도 안 하고 싶다.’

한때 이런 인터넷 유행어가 있었습니다. 바쁘게 돌아가는 하루하루를 살다 보면 문득 마음속에서 흘러나오는 ‘좀 쉬고 싶다’라는 탄식과 같은 말이겠지요.

회사에서 일하다 보면 아무것도 안 했는데 시간이 훌쩍 지나가는 때가 있습니다. 그럴 때 하루 동안 뭘 했나 생각해보면, 이 파일을 열어 저 파일로 복사해서 붙여넣기, 파일 하나하나 열어서 특정 내용이 있는지 찾아보기 같은 단순 반복 작업만 하다 끝나 버린 경험이 제법 있습니다. 여러분도 그런 경험이 있지 않나요? 이 책은 이런 아무것도 아닌 단순 반복 작업을 이제 하고 싶지 않은 여러분에게, 당신 대신에 일해줄 프로그램이 있다고 이야기해 주는 책입니다.

요즘 들어 파이썬의 인기가 뜨겁습니다. 인공 지능 관련 라이브러리는 물론 그 외에도 수많은 확장 기능이 있으며, 〈엔터〉 키만 누르면 바로 결과를 확인해 볼 수 있는 편리함과 문법적 간결함이 파이썬의 인기 비결이라는 생각입니다. 그리고 인터넷에 공개된 수많은 강좌와 소스 코드가 있으니 접근성도 매우 좋습니다. 그래서 프로그래밍이라는 말만 들어도 뒷걸음질하던 분들도 ‘그렇게 쉽다며? 한번 보거나 해볼까?’라는 마음이 들 정도입니다.

이렇게 쉬운 파이썬과 이 책에서 나오는 자동화 아이디어로 여러분의 엑셀 작업 시간을 줄여 봅시다. 그러면 아무것도 안 해도 되는 시간이 늘어서 다음 도약을 준비할 시간을 마련할 수 있지 않을까요? 빈 시간은 그저 비어 있는 게 아니라 힘을 모으는 시간이 될 수 있습니다.

늘 곁에서 함께 해주는 우리 가족 그리고 편집에 신경 써주신 신승준 님에게 감사드립니다.

여러분의 일분일초가 더욱더 소중한 시간이 되는데 이 책이 도움이 되었으면 합니다.

2020년 가을
서수환

1장

파이썬이란?

0122

유비, 매일 밤
파이썬 세미나에 참가하다



은미 하이~ 비, 잘 지냈어?

점심을 먹으러 들어간 회사 근처 식당에서 유비는 영업 담당 김은미 씨와 만났습니다.

유비 김은미 씨! 다른 사람 이름은 제대로 불러야죠. 제 이름은 김유비란 말입니다.

은미 뭐야, 왜 이리 딱딱하게 굴까?

유비 회사원이라면 모름지기 공사를 구분해야죠.

둘은 삼국어패럴 입사 동기로 그룹 연수에서 함께 신입 사원 교육을 받았습니다. 삼국어패럴은 의류업을 하는 중견 기업입니다. 도매업뿐만 아니라, 베트남에 자회사 공장도 두고 독자적인 브랜드 상품을 개발하고 있습니다. 비록 수는 아직 적지만 직영점도 운영하는 견실한 기업입니다.

은미 그리고 보니 얼마 전에 무슨 프로그래밍 세미나에 참여했다고 올리지 않았어? 영업부에 있다가 총무부로 가더니 이젠 IT 기업으로 이직하려고?

유비 그런 거 아니야. 요즘은 초등학생도 학교에서 프로그래밍을 배우는 시대니까, 프로그래밍을 모른다면 뒤처지는 게 아닐까 싶어서 말이야.

은미 에이, 그런 거였구나. 회사를 관두려고 그러나 했는데 괜한 걱정이었네. 그나저나 꽤 열심히 하는 것 같던데 어떤 걸 배웠어?

유비 파이썬이라는 건데, 읽고 쓰는 게 쉬워 초보자가 배우기 좋은 프로그래밍 언어 같아.

은미 뭔가 재미있어 보이네. 유비도 알겠지만, 영업 일이라는 게 결국 엑셀로 이것저것 다하는 거잖아. 조 과장님이 엑셀 VBA 배우면 일이 쉽게 끝난다며 약간 가르쳐 주셨는데, 웬지 어려워서 말이지. 엑셀 VBA랑 파이썬은 뭐가 다른 거야?

유비 그게. 나도 이제 막 공부를 시작한 터라…….

.....

그렇군요. 둘 다 프로그래밍에 흥미가 있어 보이네요. 그렇다면 파이썬 초보인 유비를 대신해서 파이썬이란 어떤 것인지, 프로그래밍이란 과연 무엇인지, 여러분 일에 어떤 도움이 되는지 설명하겠습니다.

01 | 파이썬의 특징

프로그래밍을 배우려면 수많은 언어 가운데 하나를 골라야 합니다. 여러분은 어떤 프로그래밍 언어를 알고 있나요? 이 책에서 소개하는 파이썬 외에도 자바나 자바스크립트, C 언어 등이 널리 쓰이는 프로그래밍 언어입니다.

유비가 파이썬을 선택한 이유가 있습니다. 사무직 종사자가 배우기에는 파이썬만 한 게 없기 때문입니다. 그 이유를 확실히 알기 위해 파이썬의 특징을 살펴봅시다.

언어 사양이 단순하다

파이썬은 유비가 말한 대로 아주 배우기 쉬운 프로그래밍 언어입니다. 그것은 파이썬이 무척 간단한 프로그래밍 언어이기 때문인데, 예약어가 적다는 게 가장 큰 이유입니다. 외워야 하는 예약어가 적으니까 쉽게 친숙해질 수 있습니다.

False	None	True	and	as	assert	async
await	break	class	continue	def	del	elif
else	except	finally	for	from	global	if
import	in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with	yield

표 1-1 파이썬 예약어(파이썬 3.7.4 기준 35개)

예약어는 말 그대로 프로그래밍 언어에서 미리 정해진 특별한 의미가 있는 단어를 말합니다. True나 class가 대표적입니다.

예약어가 많으면 미리 이해할 것도, 기억해야 할 문법도 늘어납니다. 이런 예약어가 파이썬에는 상당히 적어서 그만큼 언어 사양이 단순해집니다.

들여쓰기가 문법이다

유비는 파이썬 프로그램이 읽고 쓰기 쉽다고 했습니다. 그 비밀은 들여쓰기(indent)입니다. C나 자바를 비롯한 대다수 프로그래밍 언어에서 들여쓰기는 사람이 코드를 읽기 쉽게 하려는 표현 방법이지, 동작을 지시하는 문법과는 아무런 관련이 없습니다. 파이썬에서 들여쓰기가 실제로 사용된 다음 쪽 코드 예제를 살펴봅시다. 각각의 코드가 어떤 동작을 하는지는 나중에 자세히 설명합니다. 지금은 잘 몰라도 괜찮습니다. 여기서는 첫 줄의 코드 내용에 따라 둘째 줄 이후의 들여쓰기가 “문법적으로 의미가 있다 = 프로그램 동작과 밀접한 관계가 있다” 정도만 이해하면 충분합니다.

예를 들어 파이썬 프로그램에는 if 문이라고 부르는 조건 분기 식이 자주 등장합니다. if 문에서는 어떤 조건이 성립할 때(참일 때), 순차적으로 실행할 코드를 작성합니다. 이런 코드는 적어도 한 줄, 상황에 따라서는 여러 줄에 걸쳐 작성합니다. 이렇게 하나로 묶어 처리하는 부분을 코드 블록이라고 합니다. 이런 블록을 표현할 때 어떤 언어에서는 중괄호({})로 감싸도록 하는 규칙이 있습니다(자바나 C 등).

그에 비해 파이썬은 if 문 끝에 있는 :(콜론) 다음에 줄바꿈하고, 그 이후 들여쓰기 줄은 if 문 조건을 만족할 때 실행하는 코드입니다. 블록을 표현할 때는(중괄호 같은 별도의 표시 없이) 코드의 시작 위치를 들여쓰기로 일치시켜 사용합니다.

```

if a = 0:
    print(a)
    print("True")

```

그림 1-1 조건 분기 if 문에서 사용하는 들여쓰기 예제

if 문 이외에도 들여쓰기가 필요한 코드는 많습니다. 계속해서 들여쓰기 예를 살펴봅시다.

- 반복 처리하는 for 문

```

for score in test:
    sum += score
    print(sum)

```

- 반복 처리하는 while 문

```

while i < 5:
    print(i)
    i += 1

```

- myname이라는 함수를 정의하는 def 문

```

def myname(name):
    str = "My name is " + name
    print(str)

myname("minho")

```

그림 1-2 들여쓰기를 사용하는 다른 문법 예제

이렇게 들어가고 나오는 들여쓰기가 문법의 일부가 되면, while 문 내부가 어디까지인지, for 문 내부는 어디까지인지, 또 if 문 조건식이 성립할 때 실행하는 코드가 어디서부터 어디까지인지 한눈에 파악할 수 있습니다.

들여쓰기는 탭(Tab) 키나 스페이스(Space) 키로 입력하는데, 탭과 스페이스를 섞어 쓰면 시작 위치가 미묘하게 달라져서 혼동하기 쉬우므로 들여쓰기는 탭 키를 사용한다고 기억해 둡시다. 이 책에서 다루는 파이썬 IDLE¹이나 비주얼 스튜디오 코드(Visual Studio Code)에서는 기본값으로 탭 키가 스페이스 4개에 해당합니다. 들여쓰기는 실제로 들여쓰기가 필요한 코드를 다룰 때 자세히 설명하겠습니다.

라이브러리가 풍부해서 다양한 용도로 사용할 수 있다

예약어가 적어서 언어 사양이 단순한데 다양한 용도로 사용할 수 있다니 모순처럼 들릴 수도 있습니다. 그러나 파이썬을 다양한 목적으로 사용할 수 있는 이유는 사용 가능한 라이브러리(Library)가 풍부하고 쓰기도 편하기 때문입니다. 라이브러리는 특정한 목적을 위해 만든 프로그램을, 다른 프로그램에서도 사용할 수 있게 형식을 적절히 변경해 제공하는 프로그램 모음을 말합니다. 파이썬에는 다양한 용도로 사용할 수 있는 수많은 라이브러리가 있습니다.

라이브러리에는 표준 라이브러리와 외부 라이브러리가 있습니다.

라이브러리명	주요 기능	구분
string	문자열 처리	표준
re	정규 표현식	표준
datetime	날짜, 시간 처리	표준
random	난수 생성	표준

1 IDLE는 (Python's) Integrated Development and Learning Environment의 약어입니다.

pathlib	객체 지향 파일 시스템 경로	표준
sqlite3	sqlite3 데이터베이스 조작	표준
zipfile	zip 압축	표준
Tkinter	GUI	표준
shutil	고수준 파일 조작	표준
NumPy	수치 계산	외부
SciPy	과학 기술 계산	외부
Pandas	자료 분석	외부
Matplotlib	그래프 그리기	외부
Pygame	게임 작성용	외부
simplejson	JSON 인코딩/디코딩	외부
django	웹 프레임워크	외부
Beautiful Soup	스크래핑(HTML에서 정보 추출)	외부
TensorFlow	기계 학습	외부

표 1-2 파이썬 주요 라이브러리

표준 라이브러리는 파이썬을 설치할 때 함께 설치되어 바로 이용할 수 있는 라이브러리입니다. 한편, 외부 라이브러리는 필요에 따라 여러분이 직접 설치해야 합니다. 일단 설치하고 나면 표준 라이브러리와 같은 방법으로 사용할 수 있습니다.

이 책의 주제이기도 한, 파이썬으로 엑셀 작업을 효율적으로 하는 게 가능한 이유는 엑셀 파일을 다루는 라이브러리가 제공되기 때문입니다. 이 책에서는 엑셀용 라이브러리 가운데 기능이 풍부한 `openpyxl`을 사용합니다.

라이브러리	주요 기능
openpyxl	엑셀 파일(.xlsx) 읽고 쓰기 가능
xlrd	엑셀 파일(.xls, .xlsx) 자료 읽기 가능
xlwt	엑셀 파일(.xls) 자료와 서식 쓰기 가능
xlswriter	엑셀 파일(.xlsx) 자료와 서식 쓰기 가능

표 1-3 엑셀 문서를 다루는 주요 파이썬용 라이브러리

이외에도 엑셀 관련해서 VBA를 대신할 라이브러리도 있지만 이용하려면 설정이 복잡하거나 유료라서 이 책에서는 다루지 않습니다.

02 | VBA와 파이썬의 차이점

“엑셀에는 VBA가 있잖아?”라고 생각하는 분도 계실 테지요. 은미 씨의 상사인 조 과장도 VBA 애용자인 모양입니다.

VBA를 잘 모르는 분을 위해 VBA가 무엇인지 설명하자면 VBA는 Visual Basic for Applications의 약어로, 비주얼 베이직은 마이크로소프트(Microsoft)가 만든 범용 프로그래밍 언어입니다. PC에서 사용하는 프로그래밍 언어 가운데 역사가 가장 오래된 언어인 베이직(Basic)에서 발전했습니다. 비주얼 베이직이 입문용 프로그래밍 언어로 자리 잡으면서, 윈도우 환경에서 프로그래밍과 시스템 개발에도 널리 사용되었습니다.

계속해서 for Applications의 애플리케이션(Application)이 무엇인지 설명하겠습니다. 컴퓨터 소프트웨어에는 OS(Operating System = 운영체제)와 애플리케이션 소프트웨어(응용 소프트웨어)가 있습니다. 여러분이 사용하는 PC를 예로 든다면 윈도우 10이 OS입니다. OS가 없다면 컴퓨터는 동작하지 않습니다.

반면, 애플리케이션은 OS에서 동작하는 특정 목적을 지닌 소프트웨어입

니다. 예를 들어 표 계산, 그림 편집, 급여 계산 등을 하는 프로그램을 애플리케이션 소프트웨어라고 합니다. 따라서 VBA에서 A가 뜻하는 애플리케이션이란 엑셀(Excel)이나 워드(Word) 같은 마이크로소프트 오피스 계열의 소프트웨어입니다. 그 외에도 액세스(Access), 파워포인트(PowerPoint), 아웃룩(Outlook) 등이 모두 애플리케이션입니다.

결국 VBA란 엑셀이나 워드 등에 특화된 즉, 그러한 오피스 소프트웨어의 기능을 이용할 수 있는 비주얼 베이직을 기반으로 한 프로그래밍 언어를 말합니다.

VBA를 사용하면 엑셀이나 액세스에 특화된 업무를 자동화하거나, 엑셀이나 액세스의 기능을 확장해 입력 폼을 만든다든지 하여, 전문 애플리케이션이 수행하는 기능과 유사한 기능을 구현할 수 있습니다. 실용적인 애플리케이션을 엑셀이나 액세스의 기능을 써서 비교적 짧은 시간에 만들 수 있다는 게 VBA의 가장 큰 장점입니다. 그런 특징 때문에 실제 업무에서 많이 사용합니다.

하지만 반드시 마이크로소프트 오피스를 써야 한다는 단점이 있습니다. 또한, 업무에 필요한 기능이나 데이터의 양이 엑셀이나 액세스가 처리할 수 있는 범위를 넘게 되면 방법이 없습니다. VBA는 어디까지나 엑셀이나 액세스라는 애플리케이션 안에서 동작하기 때문입니다.

또한 마이크로소프트 오피스의 맥(Mac) 버전도 있지만 기반이 되는 운영체제가 다르기 때문에, 맥용 VBA는 윈도우보다 사용할 수 있는 기능이 적어 윈도우에서 작성한 VBA 프로그램을 맥에서는 사용하지 못할 때가 많습니다. MS 오피스와 호환되는 여타 오피스 프로그램들²에서도 VBA 프로그램을 사용할 수 없습니다. 데이터는 호환되지만, VBA가 호환되지 않기 때문입니다.

이렇듯 VBA는 동작하는 플랫폼이 한정적이고 범용성에 문제가 있습니다.

2 (오픈이) 오픈오피스(OpenOffice)나 리브레오피스(LibreOffice) 등이 있습니다.

반면 범용 프로그래밍 언어인 파이썬은 인터프리터로 동작하므로, 여러 하드웨어나 OS에서 문제없이 사용할 수 있습니다.

용어 해설 인터프리터 언어와 컴파일러 언어

사람이 작성한 프로그래밍 코드를 컴퓨터는 그대로 실행하지 못합니다. 프로그래머는 ‘프로그램을 작성한다’고 말하지만, 실제로는 프로그래밍 언어 문법에 맞춰 키보드로 코드를 입력하는 것에 불과합니다. 컴퓨터가 유일하게 실행할 수 있는 코드는 기계어입니다. 기계어는 사람이 이해하기도 입력하기도 어렵습니다. 따라서 프로그래머가 입력한 코드를 컴퓨터가 이해할 수 있도록 기계어로 변환해야 합니다. 이런 변환 기능을 하는 것이 인터프리터와 컴파일러입니다. 둘은 동작 방법이 서로 다릅니다.

인터프리터는 작성된 프로그램을 한 줄씩 기계어로 변환해 하나씩 실행합니다.

반면에 프로그램 전체를 미리 기계어로 변환하는 것이 컴파일러입니다. 변환된 실행 파일을 열면 프로그램이 실행됩니다.

파이썬에서는 각각의 OS에 대응하는 인터프리터를 파이썬을 설치할 때 함께 설치합니다. 파이썬 프로그램 코드는 이 인터프리터가 해석해서 실행합니다.

프로그래밍 언어에 따라서는 인터프리터와 컴파일러를 함께 사용하는 경우도 있습니다. 예를 들어 자바는 컴파일러를 사용하지만, 컴파일러가 작성하는 것은 ‘중간 코드’라고 부르는 코드입니다. 이 시점에는 아직 기계어가 아닙니다. 이 중간 코드를 각각의 OS(윈도우나 맥 OS, 리눅스)에서 동작하는 자바 가상 머신(Java Virtual Machine)이 인터프리터처럼 기계어로 변환해 실행합니다.

파이썬은 윈도우뿐만 아니라 맥 OS, 우분투(Ubuntu) 같은 리눅스 OS에서도 동작합니다. 일반 PC는 물론 네트워크에 있는 서버에서도 동작합니다. 요즘 자주 듣는 클라우드 서버에서도 사용합니다.

또한 파이썬은 실행에 필요한 메모리나 CPU, 하드디스크의 용량 자원이 적게 든다는 장점이 있습니다. 따라서 라즈베리 파이 같은 저가의 단일 보드

컴퓨터에서도 동작합니다.

이렇듯 파이썬은 광범위한 환경에서 프로그램을 작성, 실행할 수 있습니다.

입문자를 위한 VBA 추가 설명

엑셀을 제법 다뤄본 사람이라면 “그래도 엑셀 하면 VBA 아니겠어?”라고 생각할지도 모릅니다. 그래서 VBA와 파이썬을 좀 더 자세히 비교해 보겠습니다. 때에 따라서는 어려운 이야기라고 느낄 수도 있는데 반드시 완벽히 이해할 필요는 없습니다. 어려운 부분은 그냥 넘어가도 괜찮습니다.

비주얼 베이직은 1990년대에 마이크로소프트에서 개발한 범용 프로그래밍 언어로 역사가 깊은 언어입니다. 베이직이라고 부르는 언어의 맥을 잇는 언어입니다.

베이직(BASIC, Beginner's All-purpose Symbolic Instruction Code)은 PC가 나온 초창기에 알기 쉽고 누구나 사용할 수 있는 프로그래밍 언어로 인기를 끌었습니다. 전문 프로그래머가 아니라 개인도 즐겁게 프로그래밍할 수 있는 시대가 왔으며 지금의 파이썬과 비슷한 인기를 누렸습니다.

그런 베이직 언어의 흐름을 계승한 비주얼 베이직은 지금도 유력한 통합 개발 환경³입니다. 입력 폼을 만들고, 버튼 같은 GUI 부품을 배치하고, 클릭 이벤트를 처리하는 프로그램을 만드는 개발 방법은 RAD(Rapid Application Development)라고 부르며 인기가 있었습니다.

비주얼 베이직의 인기가 높아지는 데 발맞추어 VBA는 1990년대 후반, 엑셀, 액세스에 실리게 되었습니다. 이렇게 해서 엑셀, 액세스는 일반 애플리케이션 소프트웨어의 범주를 넘어 전용 애플리케이션 개발도 가능한 환경이 되었습니다. VBA를 실행 되면서 입력 폼에 버튼을 배치해 워크시트에 있는 값을 처리하거나 엑셀 함수로는 할 수 없던 복잡한 계산도 실행할 수 있게

3 통합 개발 환경(Integrated Development Environment, IDE)은 코딩, 디버깅, 컴파일, 배포 등 프로그램 개발에 필요한 모든 작업을 하나의 프로그램 안에서 처리할 수 있는 소프트웨어를 말한다.

되었습니다.

VBA는 마이크로소프트 오피스 버전의 업그레이드에 따라 기능이 강화되었지만, 기본적인 언어 사양에는 큰 변화가 없었습니다. 베이직은 초보자를 위한 언어라서 이해하긴 쉽지만 입력해야 할 코드가 길어지는 단점이 있습니다.

어떤 식으로 VBA 코드가 길어지는지 다음 예제로 살펴봅시다.

```
1 For i = 1 To 5
2   ___If Cells(i, 1).Value >= 60 Then
3     ___Cells(i, 2).Value = "good"
4   ___Else
5     ___Cells(i, 2).Value = "bad"
6   ___End If
7 Next
```

이 예제는 A열 1행부터 5행 사이에 들어 있는 값을 조사해서 60점 이상이면 그 옆 셀에 good을, 그렇지 않으면 bad라고 출력하는 코드입니다. 조건 분기 if 문에 대응하는 Then이 있어야 하고 반드시 End If로 끝나야 합니다.

하지만 파이썬은 같은 내용이라도 훨씬 짧은 코드로 깔끔하게 작성할 수 있습니다.

다음은 앞에서 작성한 VBA 코드를 삼항 연산자라는 파이썬 문법을 이용해 만든 코드입니다.

```
1 for row in range(1, 6):
2   ___sh.cell(row, 2).value = "good" if sh.
   cell(row, 1).value >= 60 else "bad"
```

이렇게 파이썬은 두세 줄로 끝났습니다. VBA처럼 말이 계속 끊이지 않고 이어지는 방식과는 다르지요.

03 | 파이썬 프로그래밍 환경 설정하기

파이썬이 업무에 도움이 되는 프로그래밍 언어라는 것을 이제 알게 되었으니 파이썬을 본격적으로 사용해 봅시다. 일반 사무 환경에서 사용한다고 가정해서 윈도우 10 환경에 파이썬을 설치하겠습니다. 파이썬을 모두 설치하고 나서 계속해서 코드를 작성하는 에디터도 설치하겠습니다.

파이썬 설치하기

파이썬을 설치하려면 우선 윈도우용 설치 프로그램을 다운로드해야 합니다. 웹 브라우저를 열고 <https://www.python.org/>에 접속합니다.

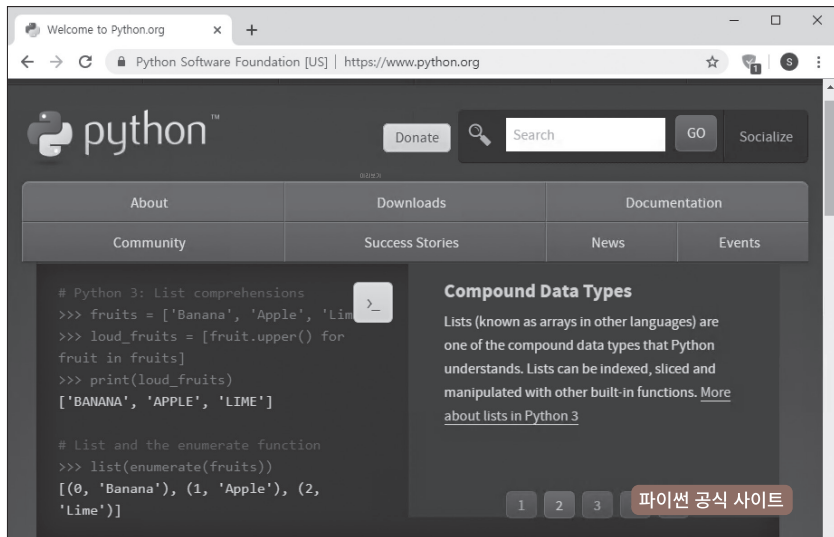


그림 1-3 파이썬 공식 사이트에 접속(<https://www.python.org/>)

홈페이지 메뉴에 있는 [Downloads]를 클릭하고, 나온 메뉴에서 [Windows]를 선택합니다.

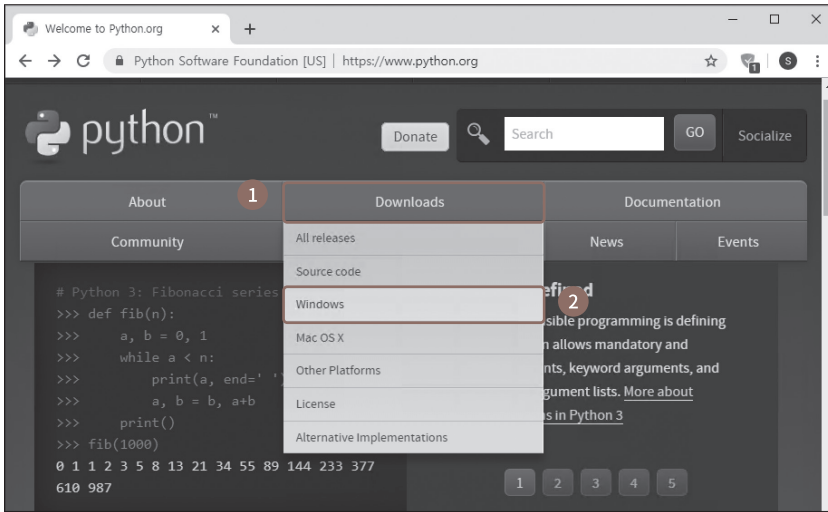


그림 1-4 [Downloads], [Windows] 순서로 클릭

홈페이지 중간에 최신판 <다운로드> 버튼이 있지만, 다운로드 정보를 살펴 보기 위해서 메뉴에서 직접 선택해 보았습니다.

페이지가 바뀌면 윈도우용 파이썬 페이지가 열리는데, 여기에서 최신판뿐만 아니라 현재 개발하고 있는 시험판 버전을 보거나 이전 버전을 다운로드할 수 있습니다.

페이지 상단에 파이썬 최신판(Latest Release)이 두 종류 있는 것을 알 수 있습니다.

이것은 각각 파이썬 3과 파이썬 2 버전의 최신판을 뜻합니다. 파이썬 2는 전통 버전이고 파이썬 3은 새롭게 개발된 버전입니다.

설치할 버전은 최신판 파이썬 3입니다. Latest Python 3 Release - Python 3.x.x 링크를 클릭해서 페이지를 엽니다. 이 책을 집필할 당시 최신 버전

은 3.7.4였습니다. 이 숫자는 달라질 가능성이 있는데 숫자와 상관없이 Latest(최신) 버전을 다운로드하면 됩니다.^{4,5}

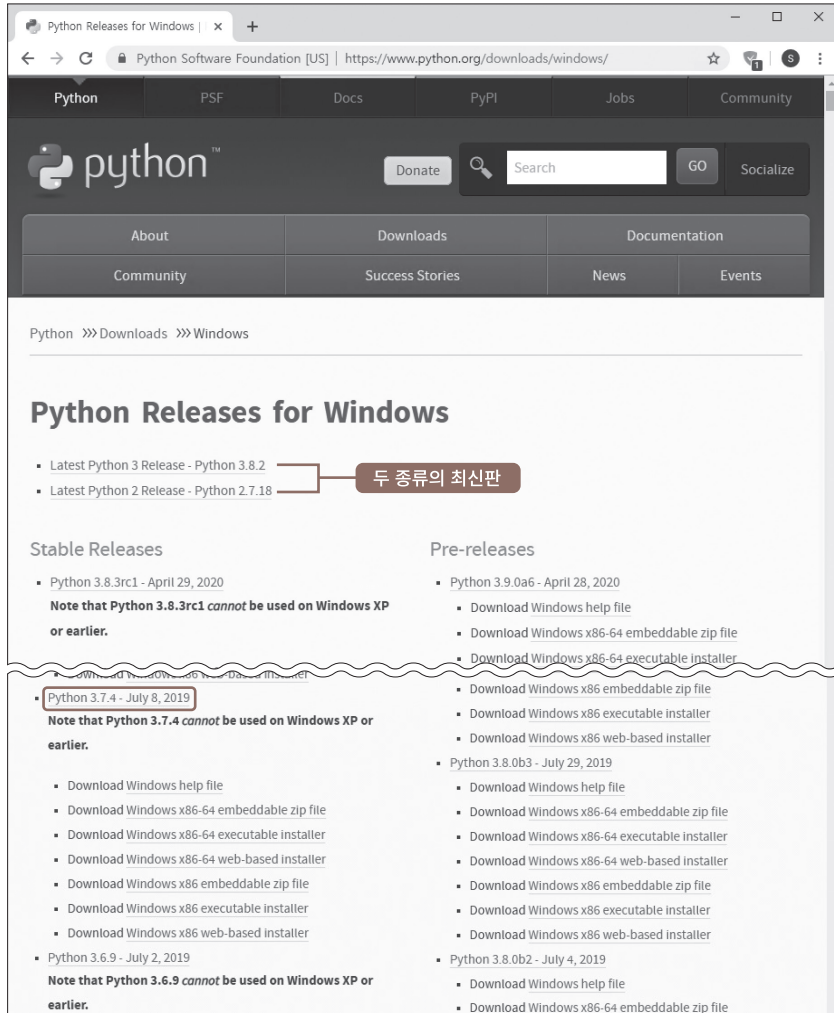


그림 1-5 Latest라고 표시된 파이썬 버전은 두 종류

- 4 이 책의 한국어판을 편집할 무렵의 최신 버전은 3.8입니다. 이 버전에서는 7장에서 소개할 라이브러리가 여전히 동작하지 않는 문제가 있었습니다. 독자가 책을 읽는 시점에는 수정되었을 수도 있지만, 최신판에서 여전히 문제가 생길 수 있어 이 책에서는 그림 1-5에 표시된 페이지를 스크롤하면 나오는 Stable Release에서 파이썬 3.7.4 버전을 클릭합니다.
- 5 사용하는 운영체제에 따라 3.7 또는 3.7.4가 아닌 다른 버전이 표시될 수도 있습니다.

Note 이 코드는 파이썬 2일까? 파이썬 3일까?

어떤 파이썬 버전을 선택해야 할까요? 앞으로는 파이썬 3을 선택해야 합니다.⁶ 하지만 인터넷에 올라온 파이썬 관련 정보를 검색하다 보면 여전히 파이썬 2로 작성한 정보가 많습니다. 어느 버전인지 나와 있으면 좋겠지만 표시되지 않은 경우도 많습니다.

그럴 때 간단히 구분하는 방법이 있는데 print 문을 보면 됩니다. 파이썬 2의 print는 독립적인 문⁷이므로 이렇게 작성합니다.

```
print "Hello, Python"
```

하지만 파이썬 3의 print()는 함수입니다. 따라서 이렇게 작성합니다.

```
print("Hello, Python")
```

소개된 코드를 잘 보고 확인하기 바랍니다.

파이썬 3 최신판(Python 3.7.4 권장) 페이지가 열리면 아래로 스크롤합니다. 하단의 Files 항목에 다운로드할 수 있는 파일 목록이 있습니다.

Windows로 시작하는 것에도 여러 종류가 있습니다. x86-64는 64bit 설치 패키지이고 x86만 적힌 것은 32bit입니다. 윈도우 10의 64bit OS라면 둘 다 사용할 수 있는데, 이 책에서는 64bit를 사용합니다. OS가 32bit인 경우에는 x86을 다운로드해야 합니다.⁸

32bit, 64bit에는 다시 세 종류의 다운로드 파일이 있는데 무엇을 선택해도 파이썬을 설치할 수 있지만 가장 간단히 설치할 수 있는 것은 executable in-

- 6 (옮긴이) 파이썬 2는 개발이 중지되어 더는 기능을 추가하거나 개선하지 않습니다. 따라서 특별한 이유가 없는 한 파이썬 3이 기본이라고 생각하면 됩니다.
- 7 명령이나 선언에서 사용하는 문(statement)은 값을 돌려주는 함수, 표현식과 더불어 프로그램을 구성하는 중요한 요소 가운데 하나입니다.
- 8 사용하는 윈도우 PC가 32bit인지 64bit인지 잘 모를 경우에는 [시작] 메뉴에서 [설정]을 열고 [시스템] → [정보]를 선택합니다. 시스템 정보를 확인해 보면 OS가 32bit인지 64bit인지 표시됩니다.

Version	Operating System	Description	MD5 Sum	File Size	GPB
Gzipped source tarball	Source release		f9f3768f75e34b342dbc06b41cbc844	24007411	SIG
XZ compressed source tarball	Source release		e9d6ebc92183a177b8e8a58cad5b8d67	17869888	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	f12203128b5c639dc08e5a43a2812cc7	30023420	SIG
Windows help file	Windows		7506675dccb9a1569b54e600ae66c9fb	8507261	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	1a98565285491c0ea65450e78afe6f8d	8017771	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	b5df1cbb2bc152cd70c3da9151cb510b	27586384	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	2586cdad1a363d1a8abb5fc102b2d418	1363760	SIG
Windows x86 embeddable zip file	Windows		121100c3cc00911100c4d5b560e3a7	7147713	SIG
Windows x86 executable installer	Windows		6f0ba59c7d8eba7bb0ee21682fe39748	26481424	SIG
Windows x86 web-based installer	Windows		04d97979534f4bd33752c183fc4ce680	1325416	SIG

그림 1-6 파이썬 3 최신판 페이지에서 다운로드 가능한 파일들

staller입니다. 사용하는 OS에 따라 32bit 또는 64bit의 executable installer를 자신의 컴퓨터에 다운로드합니다.

그리고 다운로드한 파일을 더블클릭하면 설치가 시작됩니다.⁹ 설치 화면에서 설정 가능한 항목을 살펴봅시다.

9 설치 도중에 “이 앱이 디바이스를 변경하도록 허용하겠습니까?”를 묻는 확인 화면이 표시된다면, <예(OK)>를 클릭해서 설치를 진행합니다.

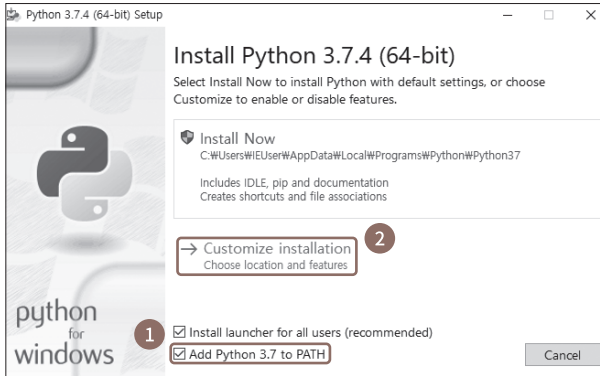


그림 1-7 설치 시작 화면

해당 화면에서 Add Python 3.7 to PATH를 체크합니다. 이렇게 하면 매번 파이썬이 설치된 폴더로 이동하지 않아도 파이썬을 실행할 수 있게 됩니다.

그리고 사용의 편의성을 위해 설정을 변경하도록 Customize installation을 클릭합니다. 그러면 [Optional Features] 화면이 표시됩니다.

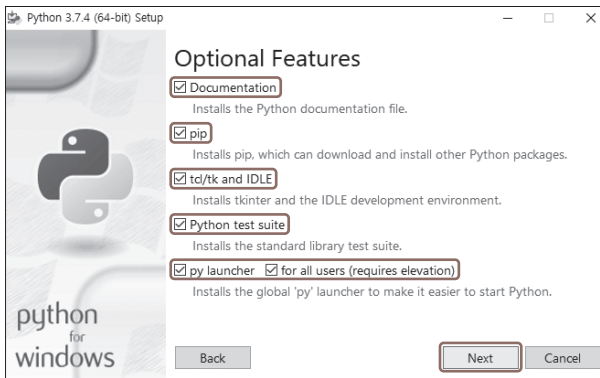


그림 1-8 Optional Features 설정 화면

표시된 선택 화면에서 기본값으로 모든 항목이 체크되어 있는지 확인하고 <Next> 버튼을 클릭합니다. 이제 [Advanced Options] 화면이 표시됩니다.

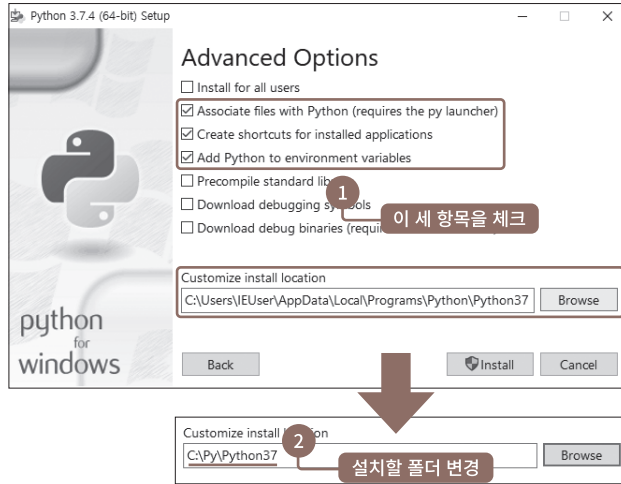


그림 1-9 Advanced Options에서 세 가지 항목을 체크하고 설치 폴더를 변경

[Advanced Options] 화면에서는 Associate files..., Create shortcuts..., Add Python to environment variables 세 항목을 체크하고, Customize install location에서 설치할 장소(폴더)를 변경합니다. 기본값 그대로라면 여러 단계의 폴더를 거쳐야 하므로 단순한 경로 폴더로 변경합니다.

예제에서는 c:\Py\Python37로 지정했습니다. 짧은 폴더 경로를 사용하는 것이 중요합니다. 경로를 변경했으면 <Install> 버튼을 클릭해서 설치를 진행합니다. Setup was successful이라고 표시되면 설치 완료입니다.

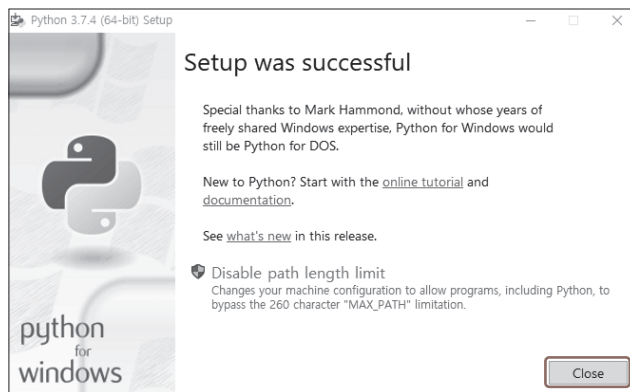


그림 1-10 설치 완료

설치 완료 화면 아랫부분에 Disable path length limit이라는 메시지가 있는데 이걸 클릭하면 OS에 설정된 경로 길이 제한(MAX_PATH)이 해제됩니다. 파이썬을 설치한 경로를 짧게 바꿨으므로 이 설정을 따로 변경하지 않아도 됩니다. <Close> 버튼을 클릭해서 설치를 종료합니다.

파이썬을 실행해서 동작 확인

설치가 끝났으면 [시작] 메뉴에 파이썬이 추가되어 있을 겁니다.

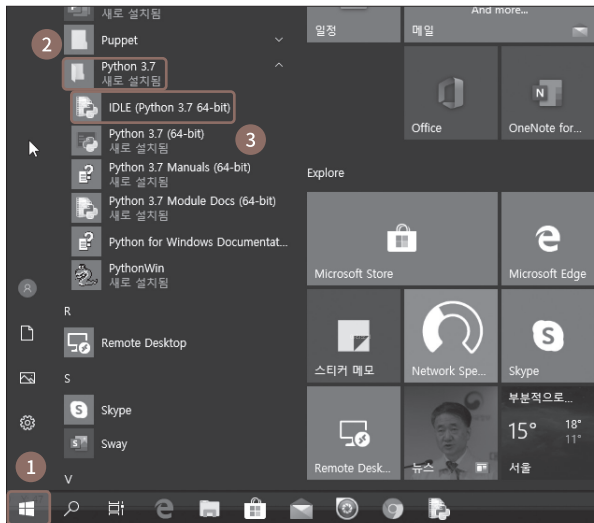


그림 1-11 [시작] 메뉴에서 Python 3.7 폴더를 확인

[시작] 메뉴에서 IDLE(Python 3.7 64-bit)를 클릭해서 실행합니다.

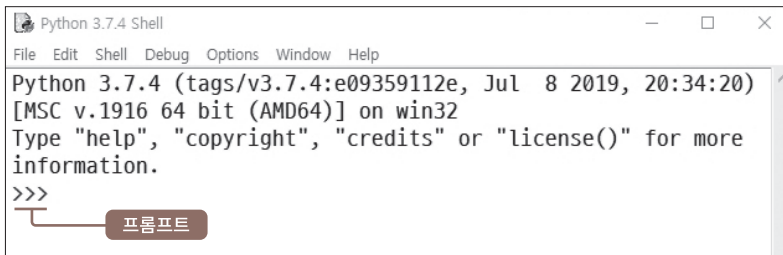


그림 1-12 파이썬 3.7.4 셸 화면