스스로 하는
프로그래밍 공부

코딩 자율학습

구독자 30만
조회수 1,500만

나도코딩의 파이썬 입문

초보자 눈높이에 맞춘 친절한 프로그래밍 자습서

나도코딩 지음

빈틈 없는 설명과 단계별 예제로 파이썬 문법과 개념 이해

언제 어디서나 웹북과 동영상으로

코딩 자율학습단과 함께 공부하기

길벗

독자의 1초를
아껴주는 정성을
만나보세요!

세상이 아무리 바쁘게 돌아가더라도 책까지 아무렇게나 빨리 만들 수는 없습니다.
인스턴트 식품 같은 책보다 오래 익힌 술이나 장맛이 밴 책을 만들고 싶습니다.
땀 흘리며 일하는 당신을 위해 한 권 한 권 마음을 다해 만들겠습니다.
마지막 페이지에서 만날 새로운 당신을 위해 더 나은 길을 준비하겠습니다.

코딩 자율학습 **나도코딩의 파이썬 입문**

Python for Beginners with Nadocoding

초판 발행 • 2023년 2월 20일

지은이 • 나도코딩
발행인 • 이중원
발행처 • (주)도서출판 길벗
출판사 등록일 • 1990년 12월 24일
주소 • 서울시 마포구 월드컵로 10길 56(서교동)
대표 전화 • 02)332-0931 | **팩스** • 02)323-0586
홈페이지 • www.gilbut.co.kr | **이메일** • gilbut@gilbut.co.kr

기획 및 책임편집 • 정지연(stopy@gilbut.co.kr) | **디자인** • 책돼지 | **제작** • 이준호, 손일순, 이진혁, 김우식
마케팅 • 임태호, 전선하, 차명환, 박민영, 지운집, 박성용 | **영업관리** • 김명자 | **독자지원** • 윤정아, 최희창

교정교열 • 이미연 | **전산편집** • 책돼지 | **출력 및 인쇄** • 천일문화사 | **제본** • 신정제본

ISBN 979-11-407-0330-2 93000
(길벗 도서번호 080357)

정가 24,000원

코딩 자율학습

구독자 30만
조회수 1,500만

나도코딩의 파이썬 입문

초보자 눈높이에 맞춘
친절한 프로그래밍 자습서

나도코딩 지음

길벗

베타 학습단의 한마디

프로그래밍을 처음 접하는 사람들을 위해 어려운 개념이나 복잡한 설명은 절제하고, 간결하고 쉬운 설명과 비유로 이해를 도우려는 나도코딩 님의 세심한 배려가 느껴졌습니다. **_박소영**

파이썬의 기본 개념을 확실하게 잡아주는 책입니다. 누구나 이해할 수 있도록 철저하게 입문자의 눈높이에 맞춰 진행하지만, 결코 대충 넘어가는 내용이 없고 흐름이 매끄러워 자연스럽게 탄탄한 기본기를 갖출 수 있습니다. 특히나 입문자에게 난관인 클래스를 비중 있게 다루고 실제 게임과 비슷한 실습 예제로 객체지향 프로그래밍까지 경험할 수 있습니다. **_이용택**

책과 강의를 함께 보니 정말 더할 나위 없이 좋았습니다. 이제 막 파이썬에 입문하는 분께 하나의 초석이 될 수 있는 책입니다. 재미있는 예제들로 코드를 직접 실행해 보니 기초 공부에 정말 많은 도움이 됐습니다. **_김예원**

다른 파이썬 기초 책을 봤지만 실습이 부족했는지 자신감이 없었습니다. 그런데 실생활을 반영한 예제와 재미있는 강의 덕분에 실력 향상에 도움이 많이 됐습니다. 특히 9장에서 게임을 이용한 코드로 흥미를 높이고, 기본 코드를 입력하고 기능을 확장하면서 완성해 가는 방법으로 학습하는 부분이 인상 깊었습니다. **_박준희**

핵심만 짚은 간결한 설명, 직접적으로 와닿는 비유, 챕터 간 개념이 점진적으로 누적되어 가는 구성으로 비전공자도 부담 없이 코딩에 입문할 수 있도록 돕는 책입니다. 저는 나도코딩 님 강의로 코딩에 입문한 뒤, 이 책을 읽으며 전체 내용과 세부 개념을 다시 정리했습니다. 책은 강의에서 간단하게 설명한 개념들을 시각적으로 보여 주고, 명확한 정의 과정을 통해 살을 붙여 주기 때문에 강의를 들었더라도 얻어갈 부분이 많았습니다. 코딩을 처음 접하는 분도 부담 없이 시작할 수 있는 책입니다. **_전영수**

이미 유튜브 강의를 본 상태지만, 뒤로 갈수록 모르고 지나쳤던 내용과, 작동 원리나 기능을 잘 알지 못하면서 넘긴 부분을 제대로 학습할 수 있었습니다. 이 책은 프로그래밍을 입문하기에 아주 최적화되어 있어서 아무것도 모르고 프로그래밍을 입문하는 분께 강력 추천하고 싶습니다. 익숙하고 이해하기 쉬운 예제와 좋은 팁까지 있어서 이해할 때까지 반복 학습한다면 한 단계 더 성장할 수 있을 겁니다. **_민성철**

다른 파이썬 책은 상세한 설명을 생략해서 헤맬 때가 종종 있는데, 이 책은 조금이라도 막힐 만한 부분은 자세히 설명해 놓아서 좋았습니다. 가장 도움이 됐던 부분은 클래스와 메서드를 이용해 게임을 제작하는 부분이었습니다. 게임을 제작하며 클래스를 직접 만들고 상속하고 유닛을 생성하는 등 풍부한 예시로 클래스를 생생하게 체험할 수 있었습니다. 초보자에게 가장 친절하고 완벽한 교재라고 생각합니다. **_차민경**

꼼꼼하게 공부할 수 있는 기회가 되어 좋았습니다. 업무나 전공과 관련 없고 주변에 파이썬을 공부하는 사람도 없어서 좀 고독했는데 학습단과 함께해서 끝까지 포기하지 않고 공부할 수 있었습니다. **_김민지**

40대 아줌마도 도전할 정도로 쉽게 만들어졌습니다. 동영상 강의로만 보다가 책을 보니 코드에 대한 이해도가 상승했습니다. 또한, 실습 문제를 풀며 앞에서 배운 내용을 이용하다 보니 앞으로 공부할 게 많이 남았다는 사실을 깨닫게 됐습니다. 앞으로 더 배워야겠지만 첫걸음을 잘 디딘 것 같아 기쁘고, 다른 언어에 대한 호기심도 생겨 전체적으로 만족합니다. **_이진향**

책 출간에 직접 참여하는 활동이 처음이라 설레기도 하고 학생이기에 지식이 부족해 남들에 미치지 못할까봐 걱정이기도 했습니다. 그러나 학습단을 하다 보니 공부에 열중하게 됐습니다. 학습단이 끝나니 아쉬우면서도 이런 활동에 참여했다는 것에 자부심을 느꼈습니다. 이 책을 계기로 제 진로가 더 확실해졌고 한 차원 높은 프로그래밍 실력을 가질 수 있을 것 같습니다. 이런 책을 편찬해 주셔서 정말 감사드립니다. **_손지형**

파이썬 기초를 다시 다질 수 있어서 좋았습니다. 파이썬 공부 초기에 나도코딩 님 영상으로 도움을 많이 받았는데 작게나마 책에 참여할 기회를 얻어 기뻤습니다. 앞으로 나올 도서에서도 도움이 될 수 있으면 좋겠습니다. **_박민영**

파이썬을 잠깐 공부했지만, 중도 포기했었습니다. 이번 기회를 통해 하루 1장씩 공부하며 책을 읽고, 코드를 치며 잘 마무리할 수 있었습니다. 드디어 파이썬 기초를 뗄 수 있겠네요! **_김시유**

개발자를 꿈꾸는 분께 큰 힘이 되는 책입니다. 실습에 필요한 환경 설치 및 실행 방법, 개발에 필요한 이론적인 내용과 예제 그리고 간단한 게임 제작 등 알차게 구성되어 있습니다. 어떤 원리로 코드가 작동되는지 자세히 알 수 있어서 입문자에게 매우 도움이 됩니다. 주위에 파이썬을 배우려는 사람이 있다면 이 책을 추천하겠습니다. **_김기홈**

내용이 직관적이고 잘 읽혔습니다. 파이썬의 기초부터 핵심 내용을 쉬운 예제들로 설명해 입문자가 배우고 이해하기 쉬웠습니다. 책 구성이 좋아 부담되지 않고 재미있게 공부할 수 있었네요. **_조근행**

학습단에 참여해 파이썬이라는 언어를 깊게 이해하고 하나씩 배워 갈 수 있었습니다. 공부한 내용을 응용해서 간단한 게임까지 직접 만들어 볼 수 있어서 매우 도움이 됐습니다. 배운 내용을 업무 자동화, 웹 스크래핑 등에 활용하며 실력을 키워 보겠습니다. **_차하나**

중간중간에 어려운 부분도 있었지만 학습단에 참여하면서 여유 시간도 활용하고 가물가물하던 파이썬 내용도 다시 기억할 수 있었습니다. 커뮤니티를 운영하며 질문에 답변해 주신 점도 매우 만족합니다. **_나희수**

베타 학습단에 참여해 주신 모든 분께 감사드립니다.
여러분의 소중한 의견이 모여 더 좋은 책을 만들 수 있었습니다.

지은이의 말

"너무 너무 감사합니다. 전 현재 60세 여자인데 코딩을 선생님 강의로 시작해서 완강했어요. 너무 너무 재미있었습니다." _lee**** 님

나도코딩 유튜브 채널의 파이썬 강의 동영상에 달린 이 댓글을 보고 많은 분이 긍정적인 충격을 받았습니다. 초등학생을 비롯한 10대뿐 아니라 40~50대 구독자 분들도 이 댓글에 힘입어 코딩 공부를 다짐하게 됐고, 어느 분은 출퇴근 시간마저 아껴가며 공부하기 위해 일부러 대중교통을 이용한다고도 했습니다.

제가 어렸을 때는 모든 차량의 변속기가 수동이었습니다. 아버지가 운전하실 때 1단부터 직접 기어를 바꾸며 바쁘게 움직이는 손이 참 멋있게 느껴졌습니다. 하지만 오르막길에서는 앞차가 출발할 때 뒤로 조금씩 밀리는 일이 있었고 심지어는 시동이 꺼지는 차량도 종종 볼 수 있었습니다. 과연 제가 어른이 됐을 때 운전을 잘 할 수 있을까 하는 걱정도 들더군요.

그런데 그 걱정은 생각보다 오래 가지 않았습니다. 언제부터인가 자동 변속기가 나왔고 지금은 대부분 차량에 자동 변속기가 장착됐습니다. 기어를 D에 놓으면 전진하고, R에 놓으면 후진하는 간단한 조작법 덕에 이제는 정말 많은 사람이 쉽게 운전할 수 있게 됐습니다.

프로그래밍 세계에서는 파이썬이 이런 자동 변속기 역할을 하고 있는 것 같습니다. 정말 많은 부분을 신경 쓰며 코딩해야 했던 다른 언어들과 달리, 파이썬은 아주 간결하면서 이해하기 쉬운 문법으로 누구나 쉽게 코딩에 발을 들여 놓게 했고 프로그램을 만들 수 있게 해 주었습니다. 또한, 전 세계 수많은 파이썬 개발자가 이미 만들어 둔 수십만 개의 패키지는 초보자가 좀 더 쉽고 편리하게 개발할 수 있게 했습니다. 프로그램의 모든 기능을 직접 만들 필요 없이 필요한 기능이 있을 때는 패키지를 찾아 사용하기만 하면 되죠.

파이썬은 활용처도 정말 다양합니다. 친구들과 함께 즐길 수 있는 게임 개발부터 프로그램을 더 사용하기 쉽게 해주는 GUI 프로그래밍, 웹페이지에서 데이터를 수집할 수 있는 웹 스크래핑, 업무를 컴퓨터가 대신해 주는 업무 자동화(RPA), 대량의 자료를 분석하고 원하는 형태의 그래프로 나타낼 수 있는 데이터 분석 및 시각화, 이미지나 동영상을 분석해 얼굴 인식 등 다양하게 응용할 수 있는 이미지 처리, 데이터로부터 의미 있는 모델을 도출해 새로운 데이터에 대한 결

과를 예측할 수 있는 머신러닝, 하드웨어 기기를 프로그래밍하고 주변 장치와 연결해 정보를 주고받을 수 있는 사물 인터넷(IoT), 자신만의 전략을 적용해 자동으로 매매할 수 있는 주식 자동화, 온라인 웹 페이지 제작, 인공지능 챗봇 등 정말 많은 분야에서 파이썬을 사용합니다.

이처럼 많은 용도로 쓰이는 파이썬은 비전공자도 쉽게 배울 수 있고 특히 빠르게 공부해서 바로 활용하려는 사람에게는 더없이 좋은 프로그래밍 언어입니다. 실제로 파이썬 기본편 강의를 수강한 분들이 회사에서 코딩 역량을 발휘해 의미 있는 성과를 내고 승진하거나, 뛰어난 경쟁자들이 참여하는 대회에서 당당히 대상을 수상하고, IT 기업에 개발자로 취업 또는 개발과 관련 없는 팀에서 개발팀으로 인사 이동을 하는 등 많은 결실을 맺었습니다.

이제는 여러분 차례입니다. 파이썬을 공부하고 나면 세상에 없던 여러분만의 프로그램을 마음껏 만들 수 있게 될 겁니다. 학생은 배움의 재미를, 취준생은 취업의 기회를, 직장인은 빠른 퇴근을, 일반인은 훌륭한 취미를 갖게 될 것입니다. '이렇게 쉬워도 되나?'라는 생각이 들 정도로 누구나 쉽고 재미있게 공부하실 수 있도록 나도코딩이 도와 드리겠습니다.

나도코딩

지은이 소개 **나도코딩**(nadocoding@gmail.com)

누구나 쉽고 재미있게 코딩을 공부할 수 있도록 다양한 강의를 제공하는 유튜버이자 개발자다. 유튜브 강의를 통해 개발자로 취업에 성공하거나 업무 성과를 인정받아 승진했다는 등의 감사 인사를 받을 때 가장 큰 보람과 희열을 느낀다. '예제를 통한 쉬운 설명', '군더더기 없는 깔끔한 강의'라는 수강평을 받은 이후로 이 두 가지는 반드시 지키자는 다짐으로 새로운 강의 제작에 임하고 있다.

유튜브 https://www.youtube.com/@nadocoding

블로그 https://nadocoding.tistory.com

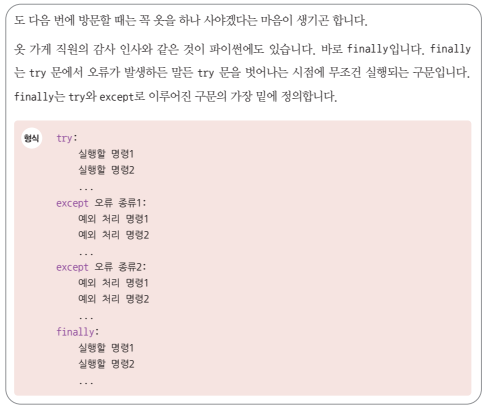
인프런 https://www.inflearn.com/users/@nadocoding

이 책의 구성

코딩을 처음 배우는 사람도 자세한 설명과 친절한 지시선으로 막힘없이 따라 할 수 있습니다. 기본 설명 외에도 팁, 노트 등을 적재적소에 배치해 완벽한 자율학습을 할 수 있게 이끕니다. 일상 속 재미있는 예제로 문법을 익히고 1분 퀴즈, 실습 문제, 셀프체크로 이어지는 단계별 학습으로 개념을 완벽하게 이해할 수 있습니다.

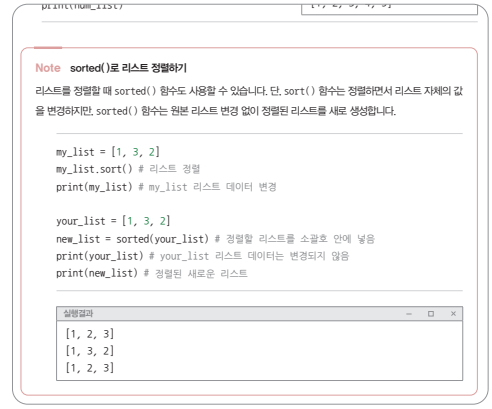
① 형식

형식을 이해하고 활용하기 쉽게 기본 구문 정리



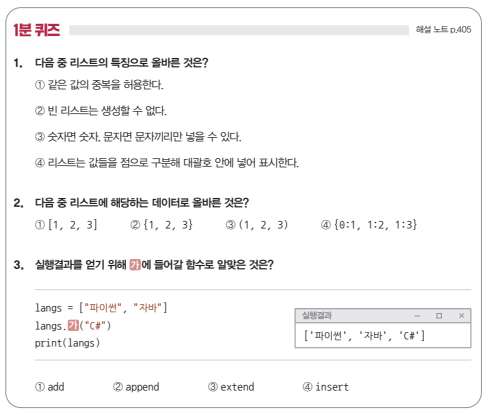
② 팁과 노트

팁과 노트로 실습 시 생길 수 있는 의문점을 해결



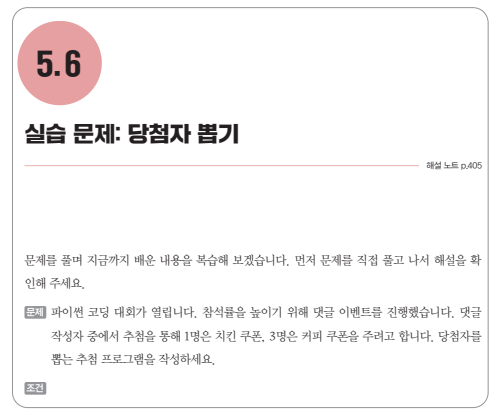
③ 1분 퀴즈

간단한 퀴즈 문제로 배운 내용 바로 확인



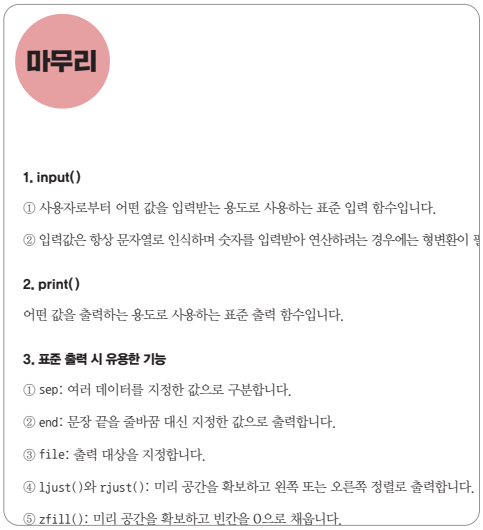
④ 실습 문제

간단한 프로그램 문제로 기본 개념 이해



⑤ 마무리

개념을 다질 수 있게 장마다 핵심 내용 정리



마무리

1. input()

① 사용자로부터 어떤 값을 입력받는 용도로 사용하는 표준 입력 함수입니다.

② 입력값은 항상 문자열로 인식하며 숫자를 입력받아 연산하려는 경우에는 형변환이 필

2. print()

어떤 값을 출력하는 용도로 사용하는 표준 출력 함수입니다.

3. 표준 출력 시 유용한 기능

① sep: 여러 데이터를 지정한 값으로 구분합니다.

② end: 문장 끝을 줄바꿈 대신 지정한 값으로 출력합니다.

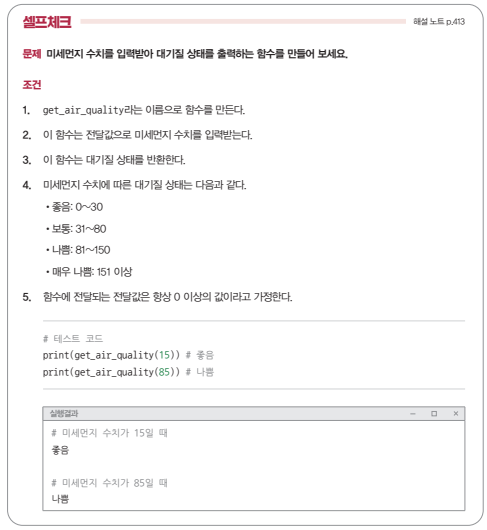
③ file: 출력 대상을 지정합니다.

④ ljust()와 rjust(): 미리 공간을 확보하고 왼쪽 또는 오른쪽 정렬로 출력합니다.

⑤ zfill(): 미리 공간을 확보하고 빈칸을 0으로 채웁니다.

⑥ 셀프체크

직접 코드를 짜 보면서 배운 내용을 이해했는지 확인



셀프체크 해설 노트 p.413

문제 미세먼지 수치를 입력받아 대기질 상태를 출력하는 함수를 만들어 보세요.

조건

1. get_air_quality라는 이름으로 함수를 만든다.
2. 이 함수는 전달값으로 미세먼지 수치를 입력받는다.
3. 이 함수는 대기질 상태를 반환한다.
4. 미세먼지 수치에 따른 대기질 상태는 다음과 같다.
   - 좋음: 0~30
   - 보통: 31~80
   - 나쁨: 81~150
   - 매우 나쁨: 151 이상
5. 함수에 전달되는 전달값은 항상 0 이상의 값이라고 가정한다.

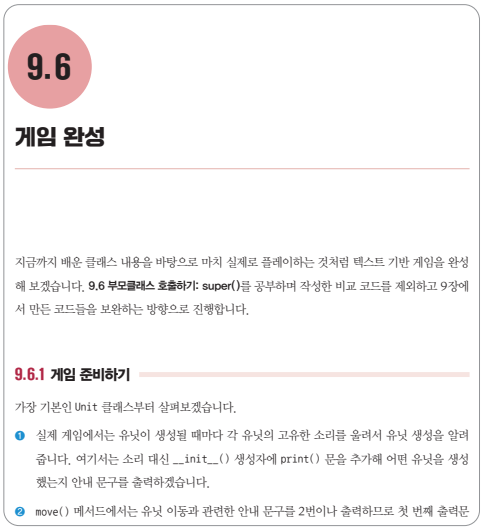
```
# 테스트 코드
print(get_air_quality(15)) # 좋음
print(get_air_quality(85)) # 나쁨
```

실행결과

```
# 미세먼지 수치가 15일 때
좋음

# 미세먼지 수치가 85일 때
나쁨
```

⑦ 게임 만들기 예제

게임을 만들어 보며 클래스 이해



9.6

게임 완성

지금까지 배운 클래스 내용을 바탕으로 마치 실제로 플레이하는 것처럼 텍스트 기반 게임을 완성해 보겠습니다. **9.6 부모클래스 호출하기: super()**를 공부하며 작성한 비교 코드를 제외하고 9장에서 만든 코드들을 보완하는 방향으로 진행합니다.

9.6.1 게임 준비하기

가장 기본인 Unit 클래스부터 살펴보겠습니다.

❶ 실제 게임에서는 유닛이 생성될 때마다 각 유닛의 고유한 소리를 울려서 유닛 생성을 알려줍니다. 여기서는 소리 대신 __init__() 생성자에 print() 문을 추가해 어떤 유닛을 생성했는지 안내 문구를 출력하겠습니다.

❷ move() 메서드에서는 유닛 이동과 관련한 안내 문구를 2번이나 출력하므로 첫 번째 출력문

⑧ 웹북과 동영상

웹북과 동영상으로 자유롭게 학습

웹북 https://thebook.io

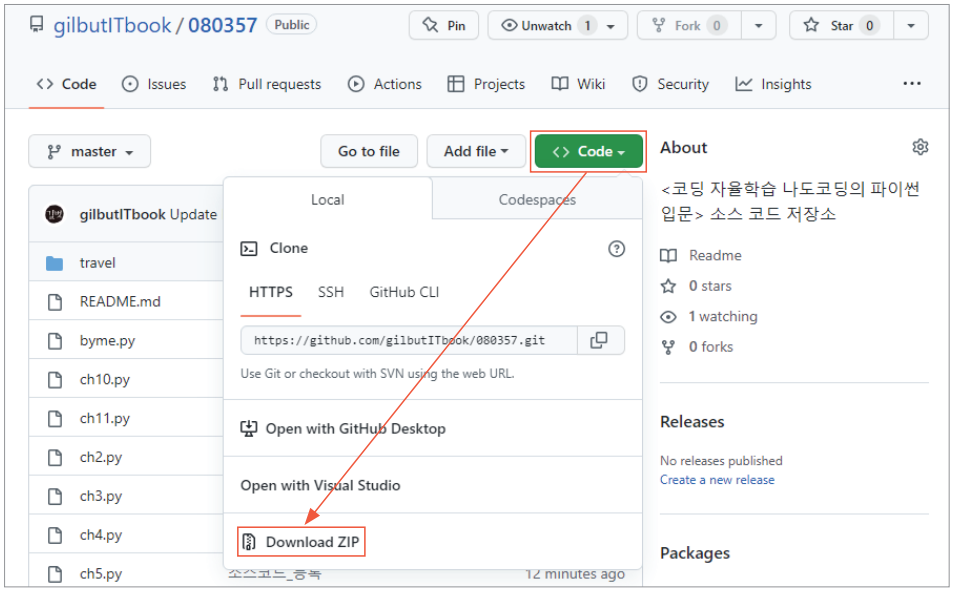
동영상 https://www.youtube.com/@nadocoding

이 책을 학습하는 방법

깃허브에서 소스 코드 내려받기

① 웹 브라우저로 **https://github.com/gilbutITbook/080357**에 접속합니다.

② 화면 오른쪽에 보이는 **Code** 버튼을 누르면 아래로 펼쳐지는 메뉴에서 **Download ZIP**을 눌러 압축 파일을 내려받습니다.



③ 내려받은 파일의 압축을 풀면 장별로 파일이 있고, 파일을 열면 소스 코드를 확인할 수 있습니다.

길벗 홈페이지 자료실에서 소스 코드 내려받기

① 길벗 홈페이지(https://www.gilbut.co.kr)에 접속해 **검색창에 도서명을 입력**하고 검색된 도서를 선택해 해당 페이지로 갑니다.

② 표지 아래쪽에 보이는 **자료실** 선택하고 **실습예제**란에 보이는 파일명을 클릭해 압축 파일을 내려받습니다.

학습 플랜

혼자 공부하기 어렵다면 학습단에 참여해 함께 공부해 보세요.

• **코딩 자율학습단 참여하기** https://cafe.naver.com/gilbutitbook

일차	학습 진도	학습할 내용	학습 날짜
1일	**1장 Hello, 파이썬!**	파이썬과 VSCode 설치, 첫 번째 프로그램 실행	/
2일	**2장 자료형과 변수**	자료형과 변수 정의, 주석 사용	/
3일	**2장 자료형과 변수**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
4일	**3장 연산자**	연산자 종류, 변수와 함수로 연산하기	/
5일	**3장 연산자**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
6일	**4장 문자열 다루기**	문자열 이해하고 슬라이싱하기, 함수로 문자열 처리 문자열 포매팅, 탈출 문자	/
7일	**4장 문자열 다루기**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
8일	**5장 자료구조**	리스트와 딕셔너리 생성, 함수 사용하기	/
9일	**5장 자료구조**	튜플과 세트 생성, 자료구조 변환 실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
10일	**6장 제어문**	조건문과 반복문 사용	/
11일	**6장 제어문**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
12일	**7장 함수**	함수 정의하고 호출하기, 변수 범위 파악	/
13일	**7장 함수**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
14일	**8장 입출력**	표준 입력과 표준 출력, format() 함수로 출력하는 방법	/
15일	**8장 입출력**	파일 입출력, 파일에 데이터 저장, with 문	/
16일	**8장 입출력**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
17일	**9장 클래스**	게임 구조 이해하고 클래스와 객체 생성	/
18일	**9장 클래스**	클래스 상속, pass와 super() 사용	/
20일	**9장 클래스**	게임 완성하고 전체 구조 파악	/
21일	**9장 클래스**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
22일	**10장 예외 처리**	예외 처리하고 오류 발생시키기	/
23일	**10장 예외 처리**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/
24일	**11장 모듈과 패키지**	모듈과 패키지 생성, 모듈 공개 설정과 실행 패키지 설치, 내장 함수와 외장 함수	/
25일	**11장 모듈과 패키지**	실습 문제와 셀프체크 풀고 마무리로 개념 다지기	/

목차

4장 문자열 다루기 093

8장 입출력 221

9장 클래스 257

11장 모듈과 패키지 351

5장

자료구조

앞에서 변수를 사용해 데이터 하나를 저장하는 방법을 알아봤습니다. 그런데 출석부와 같이 여러 학생의 이름을 관리하려면 어떻게 해야 할까요? 학생의 이름을 각각 name1, name2, name3, … 처럼 서로 다른 변수에 저장하려면 너무 많은 변수가 필요할 텐데요. 다행히 파이썬에서는 여러 데이터를 한 번에 관리하기 위한 방법을 제공합니다. 이 장에서 하나씩 살펴보겠습니다.

5.1 리스트

10명이 각각 차를 몰고 서울역에서 사당역까지 이동한다고 합시다. 1번부터 10번까지 차 10대가 나란히 갈 수 있을까요? 동시에 출발하면 아마 처음에는 가능할지 몰라도 이내 순서가 달라지고 다른 차들과 뒤섞이게 될 겁니다.

지하철을 타고 간다면 어떨까요? 지하철은 여러 칸이 하나로 연결돼 있어서 지하철 각 칸에 1명씩 타면 서울역에서 사당역까지 한꺼번에 이동할 수 있습니다. 출발도, 도착도 똑같겠지요. 다른 칸으로 자리를 옮기지만 않는다면 순서도 그대로 유지될 거고요.

앞에서 변수를 배울 때 나이, 이름, 취미 등 서로 다른 의미의 값들을 각각 변수에 저장해서 사용했습니다. 그렇다면 관련 있는 값을 여러 개 사용할 때는 어떻게 저장하는지 알아보겠습니다.

5.1.1 리스트 생성하기

지하철 칸마다 사람들이 몇 명씩 타고 있는지를 다음과 같이 변수로 나타낸다고 합시다.

```
# 지하철 칸별로 10명, 20명, 30명 승차
subway1 = 10
subway2 = 20
subway3 = 30
```

현재는 3칸밖에 없지만, 지하철이 수십 칸이라면 변수도 수십 개가 돼야 합니다. 이럴 때 **리스트**(list)를 사용하면 변수 하나로 관리할 수 있습니다.

리스트를 사용하는 형식은 다음과 같습니다. 여러 값을 쉼표(,)로 구분해 대괄호에 넣어 주면 됩니다.

형식 리스트명 = [값1, 값2, ...]

변수마다 값을 하나씩 넣었던 것과 달리 리스트는 값을 여러 개 넣을 수 있습니다. 즉, 하나의 변수가 하나의 값을 가질 수 있었다면 하나의 리스트는 여러 값을 가질 수 있습니다. 그리고 각 값의 자료형은 다를 수 있습니다.

앞의 지하철 예를 리스트로 바꿔 봅시다.

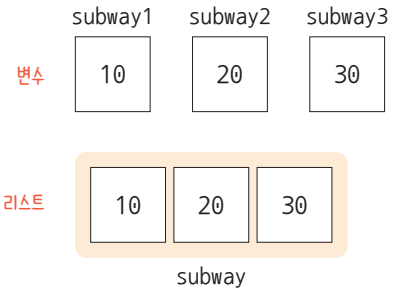
```
subway = [10, 20, 30]
print(subway)
```

실행결과

```
[10, 20, 30]
```

이렇게 하면 값 3개를 저장하는 데 subway1, subway2, subway3 변수를 사용할 필요 없이 subway 리스트 하나에 저장할 수 있습니다.

그림 5-1 변수와 리스트



Note **빈 리스트 생성하기**

아무 값이 없는 빈 리스트를 만들고 나중에 값을 추가하고 싶다면 리스트를 생성할 때 다음과 같이 대괄호만 열고 닫으면 됩니다.

```
empty_list = [] # 빈 리스트 생성하기
```

5.1.2 값 추가/삽입/삭제하기

문자열도 같은 방법으로 리스트에 저장할 수 있습니다. 곰돌이 푸와 친구들이 지하철 한 칸에 한 명씩 타고 있다고 해볼까요? 리스트로 다음과 같이 나타낼 수 있습니다.

```
subway = ["푸", "피글렛", "티거"]
print(subway)
```

실행결과

```
['푸', '피글렛', '티거']
```

문자열에서 인덱스는 데이터의 위치를 나타낸다고 했습니다. 리스트에서도 인덱스로 리스트에 저장한 데이터의 위치를 표시합니다.

그림 5-2 리스트의 인덱스

리스트도 인덱스가 있으므로 문자열처럼 리스트명에 대괄호를 붙이고 그 안에 인덱스를 넣으면 그 위치에 해당하는 값에 접근할 수 있습니다.

예를 들어, 피글렛이 몇 번째 칸에 있는지 확인해 보겠습니다. 리스트도 인덱스로 위치를 표시하므로 문자열 처리 함수에서 배운 index() 함수를 사용할 수 있습니다.

```
# 피글렛이 몇 번째 칸에 탔는가?
print(subway.index("피글렛"))
```

실행결과

```
1
```

1이 나옵니다. 왜 1이 나오는지 알죠? 인덱스는 0부터 시작한다는 점을 잊지 마세요.

다음 역에서 이요르가 티거 다음 칸에 탑니다. 리스트에서 값을 추가할 때는 append() 함수를 사용합니다.

형식 append(추가할 값)

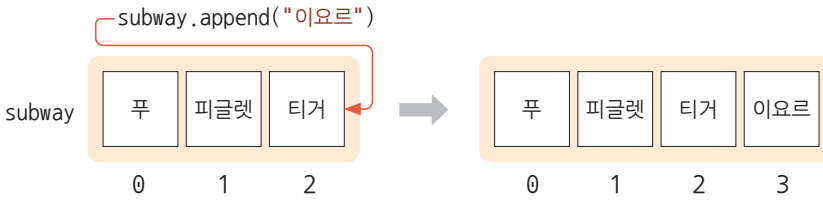
append() 함수는 리스트 끝에 값을 추가합니다. 따라서 다음과 같이 작성하면 subway 리스트에서 티거 뒤에 이요르가 들어갑니다.

```
# 이요르 탑승
subway.append("이요르")
print(subway)
```

실행결과

```
['푸', '피글렛', '티거', '이요르']
```

그림 5-3 리스트에 값 추가



이번에는 루가 지하철을 탑니다. 그런데 푸와 피글렛이 탄 칸 사이에 새로운 칸이 삽입되고 여기에 탔다고 합시다. 실제로는 일어날 수 없는 일이지만, 리스트에서는 가능합니다. 리스트에서는 insert() 함수로 중간에 값을 삽입할 수 있습니다. 이때 삽입할 위치는 인덱스로 지정합니다.

형식 insert(인덱스, 삽입할 값)

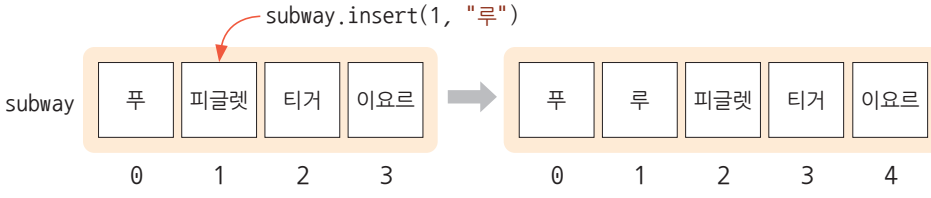
푸와 피글렛 사이에 삽입되므로 원래 피글렛 위치인 1에 삽입해야 합니다. 따라서 insert() 함수에 인덱스 1과 삽입할 값인 루를 넣고 실행합니다. 참고로 이때 인덱스를 0으로 하면 리스트의 시작, 즉 푸 앞에 삽입할 수도 있습니다.

```
# 루를 푸와 피글렛 사이(인덱스 1 위치)에 삽입
subway.insert(1, "루")
print(subway)
```

실행결과

```
['푸', '루', '피글렛', '티거', '이요르']
```

그림 5-4 리스트에 값 삽입



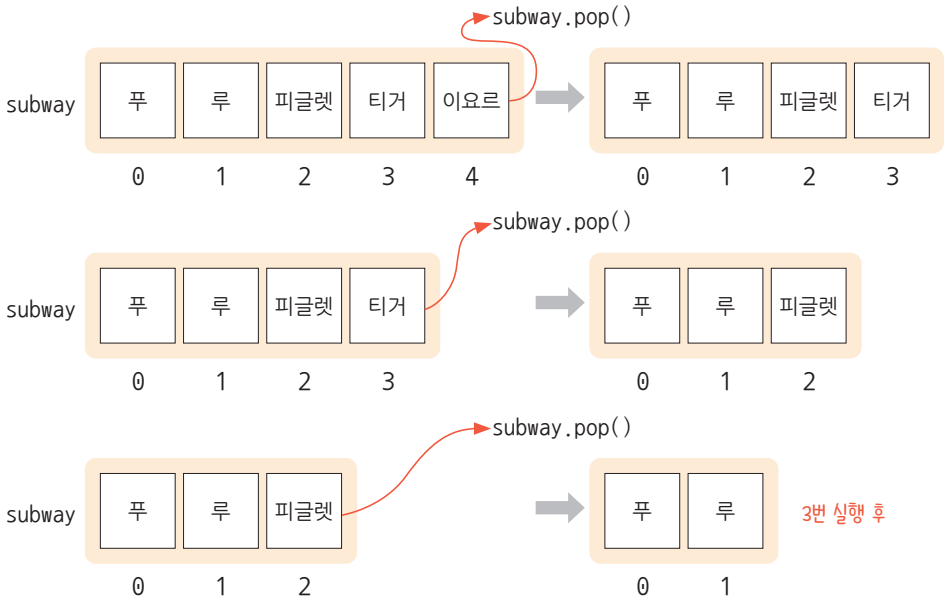
드디어 목적지에 도착해 역마다 한 명씩 지하철에서 내립니다. 이럴 때 리스트에서는 pop() 함수를 사용합니다. pop() 함수는 리스트 끝에서부터 값을 하나씩 꺼내어 반환한 뒤 삭제합니다. 값을 총 3번 삭제하는 동안 리스트가 어떻게 바뀌는지 확인해 보겠습니다.

```
# 지하철 역마다 한 명씩 내림
print(subway.pop()) # 이요르 내림
print(subway)

print(subway.pop()) # 티거 내림
print(subway)

print(subway.pop()) # 피글렛 내림
print(subway)
```

실행결과

```
이요르
['푸', '루', '피글렛', '티거']
티거
['푸', '루', '피글렛']
피글렛
['푸', '루']
```

그림 5-5 리스트에서 값 삭제



pop() 함수를 3번 실행하니 subway 리스트에는 푸와 루만 남았네요.

종착역에서는 남은 승객이 모두 내립니다. 이처럼 리스트에서는 값이 더 이상 필요 없을 때 또는 새 값을 저장하고 싶을 때 clear() 함수로 리스트의 모든 값을 지울 수 있습니다.

```
# 지하철에서 모두 내림
subway.clear()
print(subway)
```

실행결과

```
[]
```

이제 subway 리스트에는 아무 값도 없습니다.

5.1.3 중복 값 확인하기

지하철에 같은 이름의 승객이 몇 명 타고 있는지 알아보겠습니다. 문자열과 마찬가지로 count() 함수를 사용하면 리스트 안에 같은 값이 몇 개인지 확인할 수 있습니다. 설명을 위해 처음에 지하철에 있던 인원에서 푸를 뒤에 한 명 더 태우겠습니다. 그리고 count() 함수로 subway 리스트에 푸가 몇 명 있는지 확인합니다.

```
# 같은 이름이 몇 명 있는지 확인
subway = ["푸", "피글렛", "티거"]
subway.append("푸") # 푸 추가
print(subway)
print(subway.count("푸"))
```

실행결과

```
['푸', '피글렛', '티거', '푸']
2
```

푸가 지하철에 2명 있네요. 이처럼 중복 값을 확인하는 방법은 아주 간단합니다.

5.1.4 리스트 정렬하기

예제를 바꿔서 숫자로 구성된 리스트 하나를 만들어 보겠습니다.

```
num_list = [5, 2, 4, 3, 1]
```

리스트에는 1부터 5 사이 숫자들이 뒤섞여 있습니다. 여기에 sort() 함수를 사용하면 뒤섞인 숫자를 오름차순으로 정렬할 수 있습니다.

```
num_list.sort() # 오름차순 정렬
print(num_list)
```

실행결과

```
[1, 2, 3, 4, 5]
```

이때 sort() 함수 안에 다음과 같이 추가하면 리스트를 내림차순으로 정렬할 수도 있습니다.

```
num_list.sort(reverse=True) # 내림차순 정렬
print(num_list)
```

실행결과

```
[5, 4, 3, 2, 1]
```

마지막으로 리스트에 reverse() 함수를 사용하면 값의 순서를 거꾸로 뒤집을 수도 있습니다.

```
num_list.reverse() # 순서 뒤집기
print(num_list)
```

실행결과

```
[1, 2, 3, 4, 5]
```

Note **sorted()로 리스트 정렬하기**

리스트를 정렬할 때 sorted() 함수도 사용할 수 있습니다. 단, sort() 함수는 정렬하면서 리스트 자체의 값을 변경하지만, sorted() 함수는 원본 리스트 변경 없이 정렬된 리스트를 새로 생성합니다.

```
my_list = [1, 3, 2]
my_list.sort() # 리스트 정렬
print(my_list) # my_list 리스트 데이터 변경

your_list = [1, 3, 2]
new_list = sorted(your_list) # 정렬할 리스트를 소괄호 안에 넣음
print(your_list) # your_list 리스트 데이터는 변경되지 않음
print(new_list) # 정렬된 새로운 리스트
```

실행결과

```
[1, 2, 3]
[1, 3, 2]
[1, 2, 3]
```

5.1.5 리스트 확장하기

리스트에 반드시 같은 자료형의 값만 넣을 필요는 없습니다. 정수형, 실수형, 문자열, 불 형, 심지어 리스트도 집어넣을 수 있습니다.

```
mix_list = ["푸", 20, True, [5, 2, 4, 3, 1]]
print(mix_list)
```

실행결과

```
['푸', 20, True, [5, 2, 4, 3, 1]]
```

서로 다른 리스트를 합칠 수도 있습니다. extend() 함수를 다음과 같이 사용하면 됩니다.

형식 리스트1.extend(리스트2)

num_list와 mix_list를 합쳐 봅시다.

```
mix_list = ["푸", 20, True]
num_list = [5, 2, 4, 3, 1]
num_list.extend(mix_list) # 리스트 합치기
print(mix_list)
print(num_list)
```

실행결과

```
['푸', 20, True]
[5, 2, 4, 3, 1, '푸', 20, True]
```

실행하면 num_list와 mix_list의 값이 하나로 합쳐져서 num_list에 들어가게 됩니다.

지금까지 파이썬의 가장 기본 자료구조인 리스트를 공부해 봤습니다. 리스트를 사용하면 연관된 데이터를 묶어 한꺼번에 관리할 수 있습니다.

Note **리스트에서 함수 사용**

리스트에서 사용할 수 있는 함수 중에서 append(), insert(), clear(), sort(), reverse(), extend()는 별도 문장으로 실행한 뒤에 변경된 리스트의 내용을 print() 문으로 출력했습니다. 그리고 index(), pop(), count()는 print() 문 안에서 실행함과 동시에 값을 출력했습니다.

별도 문장으로 실행하는 함수들은 수행한 후에 반환하는 값 없이 리스트 자체가 변경됩니다. 그래서 print(subway.append("이요르"))라고 작성하고 실행하면 None이라는 값을 출력합니다. 반면에 print() 문 안에서 실행하는 함수들은 동작 수행 후에 어떤 값을 반환해서 이 값을 print() 문으로 출력한 것입니다. 다른 자료구조에서도 마찬가지인데, 두 실행 방식은 함수의 반환값 유무에 따른 차이입니다. 해당 내용은 **7.2 전달값과 반환값**에서 자세히 배우겠습니다.

1분 퀴즈

해설 노트 p.405

1. 다음 중 리스트의 특징으로 올바른 것은?

① 같은 값의 중복을 허용한다.

② 빈 리스트는 생성할 수 없다.

③ 숫자면 숫자, 문자면 문자끼리만 넣을 수 있다.

④ 리스트는 값들을 점으로 구분해 대괄호 안에 넣어 표시한다.

2. 다음 중 리스트에 해당하는 데이터로 올바른 것은?

① [1, 2, 3]　② {1, 2, 3}　③ (1, 2, 3)　④ {0:1, 1:2, 1:3}

3. 실행결과를 얻기 위해 가에 들어갈 함수로 알맞은 것은?

```
langs = ["파이썬", "자바"]
langs.가("C#")
print(langs)
```

실행결과

```
['파이썬', '자바', 'C#']
```

① add　② append　③ extend　④ insert

5.6

실습 문제: 당첨자 뽑기

해설 노트 p.405

문제를 풀며 지금까지 배운 내용을 복습해 보겠습니다. 먼저 문제를 직접 풀고 나서 해설을 확인해 주세요.

문제 파이썬 코딩 대회가 열립니다. 참석률을 높이기 위해 댓글 이벤트를 진행했습니다. 댓글 작성자 중에서 추첨을 통해 1명은 치킨 쿠폰, 3명은 커피 쿠폰을 주려고 합니다. 당첨자를 뽑는 추첨 프로그램을 작성하세요.

조건

1. 편의상 댓글은 20명이 작성했고, 아이디는 1~20이라고 가정한다.
2. 무작위로 추첨하되 중복은 허용하지 않는다.
3. `random` 모듈의 `shuffle()`과 `sample()` 함수를 활용한다.
4. 실행결과는 다음과 같이 표시하고 치킨 당첨자 1명, 커피 당첨자 3명을 뽑는다.

실행결과

```
-- 당첨자 발표 --
치킨 당첨자 : 6
커피 당첨자 : [9, 3, 10]
-- 축하합니다! --
```

힌트

퀴즈를 푸는 데 필요한 `random` 모듈의 `shuffle()` 함수와 `sample()` 함수의 사용법을 잠시 설명하겠습니다.

마무리

1. 리스트

① 리스트는 여러 값을 가질 수 있고 각 값의 자료형은 다를 수 있습니다. 값의 중복을 허용하며 순서를 보장합니다.

형식 `리스트명 = [값1, 값2, ...]`

② 리스트는 인덱스로 값에 접근할 수 있습니다.

형식 `리스트명[인덱스]`

③ 리스트에서 제공하는 주요 함수는 다음과 같습니다. 이 중에서 값을 반환하는 함수는 `print()` 문 안에서 바로 실행해 값을 확인할 수 있습니다. 값을 반환하지 않는 함수는 별도 문장으로 실행해야 하고, `print()` 문 안에서 실행해도 None이라고만 출력합니다.

함수	설명	값 반환 여부
index()	리스트 내 특정 데이터의 위치 반환	○
append()	리스트 맨 뒤에 데이터 추가	×
insert()	리스트의 정해진 위치에 데이터 삽입	×
pop()	리스트 뒤에서부터 데이터를 하나씩 꺼내어 반환한 뒤 삭제	○
clear()	리스트의 모든 데이터 삭제	×
count()	리스트에 포함된 데이터 개수 반환	○
sort()	리스트 내 데이터를 오름차순 또는 내림차순으로 정렬	×
reverse()	리스트 내 데이터 순서 뒤집기	×
extend()	서로 다른 리스트 합치기	×

셀프체크

해설 노트 p.409

문제 **나도대학교에서 수강신청 기간에 시스템 오류로 일부 과목이 중복 신청되는 문제가 발생했습니다. 중복 과목을 없애는 프로그램을 작성하세요.**

조건

1. 신청 과목은 리스트로 관리된다.
2. 리스트에 같은 과목이 2번 이상 포함된 경우 1개만 남기고 나머지는 삭제한다.
3. 출력 시 신청 과목의 순서는 변경해도 괜찮다.

힌트

1. 자료구조는 서로 변환할 수 있습니다.
2. 세트는 중복을 허용하지 않습니다.
3. 세트는 데이터의 순서를 보장하지 않으므로 결과를 실행할 때마다 달라집니다.

실행결과

```
# 리스트에 '자료구조, 알고리즘, 자료구조, 운영체제'를 넣었을 때
신청한 과목은 다음과 같습니다.
['자료구조', '알고리즘', '운영체제']
```

전공 불문! 나이 불문!
이 책 하나로 코딩을 시작할 수 있다!

유튜브와 인프런 최고의 인기 강의를 책으로 만나 보세요

나도코딩 30만 명의 유튜브 구독자와 2만 6천 명의 인프런 수강생이 증명한 최고의 파이썬 강의를 한 권에 담았습니다. 책을 읽으며 파이썬의 개념과 원리를 이해하고, 예제를 따라 하며 프로그램을 완성할 수 있습니다. 나도코딩의 프로그래밍 학습 노하우를 여러분께 그대로 전달해 드립니다.

코딩, 쉽고 재미있게 시작하세요

일상 속 재미있는 예제로 파이썬 기본 개념을 배우고 **1분 퀴즈**, **실습 문제**, **셀프체크**로 이어지는 단계별 학습으로 파이썬을 완공할 수 있습니다. 이제 코딩은 선택이 아닌 필수! 코딩은 전공자만 배울 수 있다는 생각으로 지레 포기하지 마세요. 파이썬은 초보자가 가장 쉽게 배울 수 있는 프로그래밍 언어입니다. 관심만 있다면 누구나 코딩을 배우는 재미를 누릴 수 있습니다. 나도코딩이 쉽고 재미있게 알려드립니다.

한 권으로 충분히 입문할 수 있습니다

코딩을 처음 배우는 사람도 단계적 용어 설명과 친절한 지시선으로 막힘없이 따라 할 수 있습니다. 기본 설명 외 팁, 노트 등을 적재적소에 배치해 혼자 고민할 필요 없는 완벽한 코딩 자율학습으로 이끕니다.

이 책을 먼저 본 베타 학습단의 한마디

- 기본 개념을 확실하게 잡아 주고 철저하게 초보자 눈높이에 맞춰 진행합니다. **_이용택**
- 핵심을 짚는 설명, 적절한 비유, 개념의 점진적 확장으로 비전공자도 부담 없이 입문할 수 있는 책입니다. **_전영수**
- 아무것도 모르는 초보자가 코딩에 입문하기에 아주 최적화된 책입니다. **_민성철**
- 이 책을 보고 진로가 더 확실해졌습니다. 베타 학습단에 참여한 것에 자부심을 느낍니다. **_손지형**
- 40대 아줌마인 제가 도전할 수 있을 정도로 쉽습니다. 첫걸음을 잘 디딘 것 같아 기쁩니다. **_이진향**
- 게임을 만들어 보며 클래스를 배울 수 있게 구성해서 이해하는 데 도움이 많이 됐습니다. **_차민경**

난이도

입문 초급 중급 고급

코딩 자율학습
나도코딩의 파이썬 입문
Python for Beginners
with Nadocoding

정가 24,000원

9 791140 703302 93000
ISBN 979-11-407-0330-2