

『바로 쓰는 파이썬: 기초 편』 1쇄 정오표

76p	수정 전	<p>문자 출력하기</p> <p>4장에서 자세히 다루겠지만 일반적으로 파이썬에서 객체를 작은따옴표(') 또는 큰따옴표(")로 묶으면 문자를 표현하는 문자열이 된다.</p>
	수정 후	<p>문자 출력하기</p> <p>5장에서 자세히 다루겠지만 일반적으로 파이썬에서 객체를 작은따옴표(') 또는 큰따옴표(")로 묶으면 문자를 표현하는 문자열이 된다.</p>
102p	수정 전	<p>식별자로 사용할 수 있는 예</p> <p>다음 city, s, stax4, first_name, s_n_u 변수는 파이썬 식별자로 사용할 수 있다.</p>
	수정 후	<p>식별자로 사용할 수 있는 예</p> <p>다음 city, s, tax4, first_name, s_n_u 변수는 파이썬 식별자로 사용할 수 있다.</p>
159p	수정 전	<ul style="list-style-type: none"> • lcenter() 메소드는 문자열을 중앙 정렬해서 • lrjust() 메소드는 문자열을 오른쪽 정렬해서 ...
	수정 후	<ul style="list-style-type: none"> • center() 메소드는 문자열을 중앙 정렬해서 • rjust() 메소드는 문자열을 오른쪽 정렬해서 ...
161p	수정 전	<pre>>>> s = '12\t1234\t\t1234567\t12345678\t123' >>> s.expandtabs() '12 1234 S4567 12345678 123'</pre>
	수정 후	<pre>>>> s = '12\t1234\t\t1234567\t12345678\t123' >>> s.expandtabs() '12 1234 1234567 12345678 123'</pre>
165p	수정 전	<p>문자열 메소드 사용하기</p> <p>• '\t 나의 살-던 고향은 꽃피는- 산골-\n\t복숭아-꽃 살-구꽃 아-기 진-달래!!!'를 화이트스페이스 문자 기준으로 분리한 후 str1 변수에 할당한다.</p>
	수정 후	<p>문자열 메소드 사용하기</p> <p>• '\t 나의 살-던 고향은 꽃피는- 산골-\n\t복숭아-꽃 살-구꽃 아-기 진-달래!!!'를 변수 str1에 할당한다.</p>
198p	수정 전	<p>(x, y, z = i, j, k)</p> <p>또는 괄호 없이</p> <p>x, y, z = i, j, k</p>
	수정 후	<p>(x, y, z) = (i, j, k)</p> <p>또는 괄호 없이</p> <p>x, y, z = i, j, k</p>

200p	수정 전	<pre>(t = x, y, z) 또는 괄호 없이 t = x, y, z</pre>						
	수정 후	<pre>t = (x, y, z) 또는 괄호 없이 t = x, y, z</pre>						
215p	수정 전	<p>매핑값이 복합자료형인 경우에는 매핑값을 수정하는 방법도 매핑값 안의 이 객체를 수정하는 방법을 사용해야 한다.</p>						
	수정 후	<p>매핑값이 복합자료형인 경우에는 매핑값을 수정하는 방법도 매핑값의 자료형이 객체를 수정하는 방법을 사용해야 한다.</p>						
218p	수정 전	<ul style="list-style-type: none"> 다음 객체 쌍을 포함하는 딕셔너리를 생성한 후 변수 d에 할당한다. 키(key)의 자료형은 문자열, 매핑값(value)의 자료형은 리스트로 한다. <table border="1" data-bbox="446 779 1220 875"> <thead> <tr> <th>키(key)</th> <th>매핑값(value)</th> </tr> </thead> <tbody> <tr> <td>'짝수'</td> <td>2, 4, 6, 8, 10</td> </tr> <tr> <td>'홀수'</td> <td>1, 3, 5, 7, 9</td> </tr> </tbody> </table>	키(key)	매핑값(value)	'짝수'	2, 4, 6, 8, 10	'홀수'	1, 3, 5, 7, 9
키(key)	매핑값(value)							
'짝수'	2, 4, 6, 8, 10							
'홀수'	1, 3, 5, 7, 9							
	수정 후	<ul style="list-style-type: none"> 다음 객체 쌍을 포함하는 딕셔너리를 생성한 후 변수 d에 할당한다. 키(key)의 자료형은 문자열, 매핑값(value)의 자료형은 리스트로 한다. <table border="1" data-bbox="446 976 1220 1072"> <thead> <tr> <th>키(key)</th> <th>매핑값(value)</th> </tr> </thead> <tbody> <tr> <td>'짝수'</td> <td>2, 4, 6, 8</td> </tr> <tr> <td>'홀수'</td> <td>1, 3, 5, 7, 9</td> </tr> </tbody> </table>	키(key)	매핑값(value)	'짝수'	2, 4, 6, 8	'홀수'	1, 3, 5, 7, 9
키(key)	매핑값(value)							
'짝수'	2, 4, 6, 8							
'홀수'	1, 3, 5, 7, 9							
219p	수정 전	<p>fromkeys() 메소드를 사용해서 딕셔너리를 만들 때는 전달인자로 키 값을 지정해줘야 한다. 전달인자로는 시퀀스형만 올 수 있다. 이때 시퀀스형의 개별 항목을 키로 하는 딕셔너리의 매핑값은 기본값인 None이다. 예를 들어 문자열인 'abcde'를 fromkeys() 메소드의 전달인자로 사용하면 다음과 같은 결과를 보여준다.</p> <pre>>>> d1 = dict.fromkeys('abcde') # 문자열의 개별 문자로 키만 생성해서 딕셔너리를 만든다. >>> d1 {'a': None, 'b': None, 'c': None, 'd': None, 'e': None}</pre> <p>문자열의 각 문자가 키로 할당되고, 기본값인 None이 모든 키의 매핑값으로 할당되었다. 이번에는 다른 시퀀스형인 리스트를 사용해서 딕셔너리를 생성해보자.</p> <p>fromkeys() 메소드를 사용해서 딕셔너리를 만들 때는 전달인자로 키 값을 지정해줘야 한다. 전달인자로는 순회형만 올 수 있다. 이때 순회형의 개별 항목을 키로 하는 딕셔너리의 매핑값은 기본값인 None이다. 예를 들어 문자열인 'abcde'를 fromkeys() 메소드의 전달인자로 사용하면 다음과 같은 결과를 보여준다.</p> <pre>>>> d1 = dict.fromkeys('abcde') # 문자열의 개별 문자로 키만 생성해서 딕셔너리를 만든다. >>> d1 {'a': None, 'b': None, 'c': None, 'd': None, 'e': None}</pre> <p>문자열의 각 문자가 키로 할당되고, 기본값인 None이 모든 키의 매핑값으로 할당되었다. 이번에는 다른 순회형인 리스트를 사용해서 딕셔너리를 생성해보자.</p>						

231p	수정 전	<p>[표 6-2] 딕셔너리 관련 메소드</p> <table border="1"> <thead> <tr> <th>문법</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>dict.fromkeys(s[, v])</td> <td>시퀀스형 s의 모든 객체가 키가 되고, v가 주어지지 않으면 매핑값이 None, v가 주어지면 매핑값이 v인 새로운 딕셔너리를 반환하는 클래스 메소드다.</td> </tr> <tr> <td>d.popitem()</td> <td>딕셔너리 d에서 임의 객체를 (키, 매핑값) 튜플로 반환하고 해당 객체를 딕셔너리 d에서 삭제한다. 딕셔너리 d가 비어 있으면 KeyError가 발생한다.</td> </tr> </tbody> </table>	문법	설명	dict.fromkeys(s[, v])	시퀀스형 s의 모든 객체가 키가 되고, v가 주어지지 않으면 매핑값이 None, v가 주어지면 매핑값이 v인 새로운 딕셔너리를 반환하는 클래스 메소드다.	d.popitem()	딕셔너리 d에서 임의 객체를 (키, 매핑값) 튜플로 반환하고 해당 객체를 딕셔너리 d에서 삭제한다. 딕셔너리 d가 비어 있으면 KeyError가 발생한다.
	문법	설명						
dict.fromkeys(s[, v])	시퀀스형 s의 모든 객체가 키가 되고, v가 주어지지 않으면 매핑값이 None, v가 주어지면 매핑값이 v인 새로운 딕셔너리를 반환하는 클래스 메소드다.							
d.popitem()	딕셔너리 d에서 임의 객체를 (키, 매핑값) 튜플로 반환하고 해당 객체를 딕셔너리 d에서 삭제한다. 딕셔너리 d가 비어 있으면 KeyError가 발생한다.							
수정 후	<p>[표 6-2] 딕셔너리 관련 메소드</p> <table border="1"> <thead> <tr> <th>문법</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>dict.fromkeys(i[, v])</td> <td>순회형 i의 모든 객체가 키가 되고, v가 주어지지 않으면 매핑값이 None, v가 주어지면 매핑값이 v인 새로운 딕셔너리를 반환하는 클래스 메소드다.</td> </tr> <tr> <td>d.popitem()</td> <td>딕셔너리 d에서 마지막 객체를 (키, 매핑값) 튜플로 반환하고 해당 객체를 딕셔너리 d에서 삭제한다. 딕셔너리 d가 비어 있으면 KeyError가 발생한다.</td> </tr> </tbody> </table>	문법	설명	dict.fromkeys(i[, v])	순회형 i의 모든 객체가 키가 되고, v가 주어지지 않으면 매핑값이 None, v가 주어지면 매핑값이 v인 새로운 딕셔너리를 반환하는 클래스 메소드다.	d.popitem()	딕셔너리 d에서 마지막 객체를 (키, 매핑값) 튜플로 반환하고 해당 객체를 딕셔너리 d에서 삭제한다. 딕셔너리 d가 비어 있으면 KeyError가 발생한다.	
문법	설명							
dict.fromkeys(i[, v])	순회형 i의 모든 객체가 키가 되고, v가 주어지지 않으면 매핑값이 None, v가 주어지면 매핑값이 v인 새로운 딕셔너리를 반환하는 클래스 메소드다.							
d.popitem()	딕셔너리 d에서 마지막 객체를 (키, 매핑값) 튜플로 반환하고 해당 객체를 딕셔너리 d에서 삭제한다. 딕셔너리 d가 비어 있으면 KeyError가 발생한다.							
237p	수정 전	따라서 문자열을 set() 생성자의 전달인자로 사용할 때는 주의가 필요하다. 더우기 세트는 중복 값을 허용하지 않기 때문에 문자열 안에 같은 단어가 들어 있다면 하나로 처리하니 주의해야 한다.						
	수정 후	따라서 문자열을 set() 생성자의 전달인자로 사용할 때는 주의가 필요하다. 더욱이 세트는 중복 값을 허용하지 않기 때문에 문자열 안에 같은 단어가 들어 있다면 하나로 처리하니 주의해야 한다.						
249p	수정 전	<ul style="list-style-type: none"> 이전 물품: 수박: 15000, 시금치: 2000, 초콜릿: 1500, 생선: 9000, 파리채: 1000 현재 물품: 시금치: 2000, 초콜릿: 1500, 생선: 9000, 감: 800, 전기장판: 20000 						
	수정 후	<ul style="list-style-type: none"> 이전 물품: 수박: 15000, 콩나물: 850, 초콜릿: 1500, 생선: 9000, 파리채: 1000 바뀐 물품: 콩나물: 850, 초콜릿: 1500, 생선: 9000, 감: 800, 전기장판: 20000 						
252p	수정 전	<pre>>>> G = networkx.Graph # 빈 그래프 인스턴스 G를 선언한다. >>> edges = [# 그래프의 연결 정보를 튜플로 저장한 리스트 ... ('A', 'B'), ('B', 'C'), ('C', 'A'), ... ('C', 'D'), ('D', 'E'), ('E', 'A'), ... ('B', 'F'), ('G', 'C'), ('H', 'C') ...]</pre>						
	수정 후	<pre>>>> G = networkx.Graph # 빈 그래프 인스턴스 G를 선언한다. >>> edges = [# 그래프의 연결 정보를 튜플로 저장한 리스트 ... ('A', 'B'), ('B', 'C'), ('C', 'A'), ... ('C', 'D'), ('D', 'E'), ('E', 'A'), ... ('B', 'F'), ('G', 'C'), ('H', 'C') ...]</pre>						
259p	수정 전	그림 7-1은 x, y, z에 x를 할당한 결과 z는 x의 객체 참조를 복사해서 갖고 있음을 보여준다.						
	수정 후	그림 7-1은 z에 x를 할당한 결과 z는 x의 객체 참조를 복사해서 갖고 있음을 보여준다.						
294p	수정 전	그리고 코드 세 번째 줄에서 센터널 값인 빈 문자열('')을 변수 fruits에 할당해서 센터널 변수를 초기화한다.						
	수정 후	그리고 코드 세 번째 줄에서 센터널 값인 빈 문자열('')을 변수 fruit에 할당해서 센터널 변수를 초기화한다.						

352p	수정 전	<pre>>>> alist = ('a', 'b', 'c', 'd', 'e')</pre>
	수정 후	<pre>>>> alist = ['a', 'b', 'c', 'd', 'e']</pre>
367p	수정 전	<pre>>>> def value_returning_product3(i, j): ... # 전달받은 두 정수의 사칙연산 결과를 튜플을 사용해서 반환하는 함수다. ... return i + i, i - j, i * j, i / j</pre>
	수정 후	<pre>>>> def value_returning_product3(i, j): ... # 전달받은 두 정수의 사칙연산 결과를 튜플을 사용해서 반환하는 함수다. ... return i + j, i - j, i * j, i / j</pre>
378p	수정 전	<pre>>>> string_format(s, 5, '*') # length를 10, fills를 '*'로 지정한다(s가 더 길다). '파이썬과 빅데이터 분석'</pre>
	수정 후	<pre>>>> string_format(s, 5, '*') # length를 5, fills를 '*'로 지정한다(s가 더 길다). '파이썬과 빅데이터 분석'</pre>
444p	수정 전	뒤집은 후 맨 마지막에 오는 객체부터 파일로 저장하려면 어떻게 할까? 이때는 for문을 사용하지 않고 다음 코드처럼 reversed() 함수로 리스트의 객체들을 역순으로 정렬한 후 문자열 메소드인 joint()를 사용한다.
	수정 후	뒤집은 후 맨 마지막에 오는 객체부터 파일로 저장하려면 어떻게 할까? 이때는 for문을 사용하지 않고 다음 코드처럼 reversed() 함수로 리스트의 객체들을 역순으로 정렬한 후 문자열 메소드인 join()을 사용한다.
470p	수정 전	11-2 노래 가사집 완성하기
	수정 후	11-2 The Zen of Python 완성하기
471p	수정 전	참고 먼저 이 문서를 텍스트 파일로 'ex11_2.py' 파일과 같은 폴더에 저장한 후 'ex11_2.py' 파일을 실행해야 한다. 문서는 https://www.python.org/dev/peps/pep-0020/ 에서 열람할 수 있다. 이 페이지를 방문해서 20문장으로 된 'The Zen of Python'을 전체 복사하여 메모장에 붙여넣은 후, 파일 확장명이 '.txt'인 파일로 저장한다. 여기서는 파일 이름을 'zen.txt'로 했다.
	수정 후	참고 먼저 이 문서를 텍스트 파일로 'ex11_2.py' 파일과 같은 폴더에 저장한 후 'ex11_2.py' 파일을 실행해야 한다. 문서는 https://www.python.org/dev/peps/pep-0020/ 에서 열람할 수 있다. 이 페이지를 방문해서 19개 문구로 된 'The Zen of Python'을 전체 복사하여 메모장에 붙여넣은 후, 파일 확장명이 '.txt'인 파일로 저장한다. 파일 저장 옵션으로 인코딩을 'utf-8'로 한다. 여기서는 파일 이름을 'zen.txt'로 했다.

500p	수정 전	<pre>>>> PartTimer.twhours(7) Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: twhours() missing 1 required positional argument: 'hours_worked' >>> PartTimer.total_wage() Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: total_wage() missing 1 required positional argument: 'self'</pre>
	수정 후	<pre>>>> PartTimer.whours(7) Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: whours() missing 1 required positional argument: 'hours_worked' >>> PartTimer.total_wage() Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: total_wage() missing 1 required positional argument: 'self'</pre>
507p	수정 전	<p>여기 나오는 super() 메소드는 하위 클래스에서 상위 클래스의 메소드를 호출할 때 사용한다. super() 메소드를 통해 상위 클래스인 PartTimer의 초기화 메소드인 __init__()를 호출하면서 수석 아르바이트생의 닉네임을 전달해서 인스턴스를 생성한다.</p>
	수정 후	<p>여기 나오는 super() 함수는 하위 클래스에서 상위 클래스의 메소드를 호출할 때 사용한다. super() 함수를 통해 상위 클래스인 PartTimer의 초기화 메소드인 __init__()를 호출하면서 수석 아르바이트생의 닉네임을 전달해서 인스턴스를 생성한다.</p>