



(<https://learningspoons.com/course/detail/pythonforfinance/>)

검색어를 입력하세요.



■ 초보자를 위한 파이썬 300제

(/book/922)

00. 파이썬 문법 리뷰

01) 유튜브

01. 파이썬 시작하기

001 ~ 010

02. 파이썬 변수

011 ~ 020
03. 파이썬 문자열
021 ~ 030
031 ~ 040
041 ~ 050
04. 파이썬 리스트
051 ~ 060
061 ~ 070
05. 파이썬 튜플
071 ~ 080
06. 파이썬 딕셔너리
081 ~ 090
091 ~ 100
07. 파이썬 분기문
101 ~ 110
111 ~ 120
121 ~ 130
08. 파이썬 반복문
131 ~ 140
141 ~ 150
151 ~ 160
161 ~ 170

23. 6. 7. 오후 2:29281 ~ 290 - 초보자를 위한 파이썬 300제

171 ~ 180
181 ~ 190
191 ~ 200
09. 파이썬 함수
201 ~ 210
211 ~ 220
221 ~ 230
231 ~ 240
10. 파이썬 모듈
241 ~ 250
11. 파이썬 클래스
251 ~ 260
261 ~ 270
271 ~ 280
281 ~ 290
12. 파일 입출력과 예외처리
291 ~ 300



281 ~ 290

281 클래스 정의

다음 코드가 동작하도록 차 클래스를 정의하세요.

```
>> car = 차(2, 1000)
>> car.바퀴
2
>> car.가격
1000
```

▼ 정답확인

```
class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

car = 차(2, 1000)
print(car.바퀴)
print(car.가격)
```

282 클래스 상속

차 클래스를 상속받은 자전거 클래스를 정의하세요.

▼ 정답확인

```
class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

class 자전거(차):
    pass
```

283 클래스 상속

다음 코드가 동작하도록 자전거 클래스를 정의하세요. 단 자전거 클래스는 차 클래스를 상속받습니다.

```
>> bicycle = 자전거(2, 100)
>> bicycle.가격
100
```

▼ 정답확인

```
class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

class 자전거(차):
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

bicycle = 자전거(2, 100)
print(bicycle.가격)
```

284 클래스 상속

다음 코드가 동작하도록 자전거 클래스를 정의하세요. 단 자전거 클래스는 차 클래스를 상속받습니다.

```
>> bicycle = 자전거(2, 100, "시마노")
>> bicycle.구동계
시마노
```

▼ 정답확인

```
class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

class 자전거(차):
    def __init__(self, 바퀴, 가격, 구동계):
        super().__init__(바퀴, 가격)
        #차.__init__(self, 바퀴, 가격)
        self.구동계 = 구동계

bicycle = 자전거(2, 100, "시마노")
print(bicycle.구동계)
print(bicycle.바퀴)
```

285 클래스 상속

다음 코드가 동작하도록 차 클래스를 상속받는 자동차 클래스를 정의하세요.

```
>> car = 자동차(4, 1000)
>> car.정보()
바퀴수 4
가격 1000
```

▼ 정답확인

```
class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

class 자동차(차):
    def __init__(self, 바퀴, 가격):
        super().__init__(바퀴, 가격)

    def 정보(self):
        print("바퀴수 ", self.바퀴)
        print("가격 ", self.가격)

car = 자동차(4, 1000)
car.정보()
```

286 부모 클래스 생성자 호출

다음 코드가 동작하도록 차 클래스를 수정하세요.

```
>> bicycle = 자전거(2, 100, "시마노")
>> bicycle.정보()
바퀴수 2
가격 100
```

▼ 정답확인

```

class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

    def 정보(self):
        print("바퀴수 ", self.바퀴)
        print("가격 ", self.가격)

class 자동차(차):
    def __init__(self, 바퀴, 가격):
        super().__init__(바퀴, 가격)

class 자전거(차):
    def __init__(self, 바퀴, 가격, 구동계):
        super().__init__(바퀴, 가격)
        self.구동계 = 구동계

bicycle = 자전거(2, 100, "시마노")
bicycle.정보()

```

287 부모 클래스 메서드 호출

자전거의 정보() 메서드로 구동계 정보까지 출력하도록 수정해보세요.

```

>> bicycle = 자전거(2, 100, "시마노")
>> bicycle.정보()
바퀴수 2
가격 100
구동계 시마노

```

▼ 정답확인


```

class 차:
    def __init__(self, 바퀴, 가격):
        self.바퀴 = 바퀴
        self.가격 = 가격

    def 정보(self):
        print("바퀴수 ", self.바퀴)
        print("가격 ", self.가격)

class 자동차(차):
    def __init__(self, 바퀴, 가격):
        super().__init__(바퀴, 가격)

class 자전거(차):
    def __init__(self, 바퀴, 가격, 구동계):
        super().__init__(바퀴, 가격)
        self.구동계 = 구동계

    def 정보(self):
        super().정보()
        print("구동계 ", self.구동계)

bicycle = 자전거(2, 100, "시마노")
bicycle.정보()

```

288 메서드 오버라이딩

다음 코드의 실행 결과를 예상해보세요.

```

class 부모:
    def 호출(self):
        print("부모호출")

class 자식(부모):
    def 호출(self):
        print("자식호출")

```

```

나 = 자식()
나.호출()

```

▼ 정답확인

```
class 부모:
    def 호출(self):
        print("부모호출")

class 자식(부모):
    def 호출(self):
        print("자식호출")

나 = 자식()
나.호출()
```

289 생성자

다음 코드의 실행 결과를 예상해보세요.

```
class 부모:
    def __init__(self):
        print("부모생성")

class 자식(부모):
    def __init__(self):
        print("자식생성")
```

```
나 = 자식()
```

▼ 정답확인

```
class 부모:
    def __init__(self):
        print("부모생성")

class 자식(부모):
    def __init__(self):
        print("자식생성")

나 = 자식()
```

290 부모클래스 생성자 호출

다음 코드의 실행 결과를 예상해보세요.

```
class 부모:
    def __init__(self):
        print("부모생성")

class 자식(부모):
    def __init__(self):
        print("자식생성")
        super().__init__()

나 = 자식()
```

▼ 정답확인

```
class 부모:
    def __init__(self):
        print("부모생성")

class 자식(부모):
    def __init__(self):
        print("자식생성")
        super().__init__()

나 = 자식()
```

풀이 영상

초보자를 위한 파이썬 강의/기초 300 문제 같이 풀어보기 28...



마지막 편집일시 : 2022년 4월 28일 10:24 오후



([https://pyquant.co.kr/product/system-trading-](https://pyquant.co.kr/product/system-trading-with-tick-data/)

with-tick-data/)

댓글 4

피드백

- 이전글 : 271 ~ 280
- 다음글 : 12. 파일 입출력과 예외처리

↑ TOP