



파이썬기초

6주차 - 1교시
클래스





학습내용

- 객체 지향의 개념
- 파이썬 객체지향프로그래밍



학습목표

- 객체 지향의 개념에 대해 이해하고 설명할 수 있다.
- 파이썬에서 제공하는 객체지향 프로그래밍 방법을 이용하여 프로그래밍 할 수 있다.



모듈

- 파이썬 코드 파일 (확장자 .py)로 파이썬 코드들을 저장하는 기본단위
- 서로 관련 있는 작은 프로그램 조각들을 모아놓은 함수들의 집합임



모듈

- 모듈의 구분

- > 표준 모듈

- ✓ 파이썬을 설치하면 파이썬 패키지안에 기본으로 포함되어 제공되는 모듈

- > 사용자 정의 모듈

- ✓ 사용자가 직접 생성하여 사용하는 모듈



모듈

- 모듈의 구분

> 외부 모듈

- ✓ 파이썬 이외의 다른 기관이나 SW회사 등에서 배포한 모듈로 특정 기능을 제공하는 모듈
- ✓ 외부(서드파티) 모듈 설치에 pip(the package installer for Python) 명령을 이용하여 설치함



표준 모듈

- 파이썬은 프로그래밍을 개발하기 위해 기본적으로 사용해야 하는 문자 처리, 웹, 수학과 관련된 다양한 내장 표준 모듈을 제공함
- 추가 설치 없이 import문 한 줄로 사용함
- os 모듈, math 모듈, random 모듈, datetime 모듈, urllib 모듈 등이 있음



사용자 정의 모듈

- 사용자 정의 모듈은 “모듈이름.py” 파일로 코드 작성
- `import` 모듈이름(파일이름) 형식으로 모듈을 읽어온 후, 모듈에서 정의한 멤버들을 사용



외부 모듈

- 파이썬 이외의 다른 기관이나 SW회사 등에서 배포한 모듈로 특정 기능을 제공하는 모듈
- 외부 모듈을 다양하게 제공

데이터 분석, 통계

Numpy, Pandas, matplotlib ...

인공지능

Tensorflow, PyTorch, Keras...

웹 크롤링

BeautifulSoup, Selenium...

- 외부 모듈을 사용하기 위해서는 외부 모듈을 설치 후 사용



생각해 봅시다

객체의 의미를 생각해보고
객체의 의미를 이용해서 프로그래밍을 한다면
어떻게 할 수 있을지 생각해 봅시다.



01



객체 지향의 개념



1 | 객체 지향

01 정의

객체(object)

실생활에 존재하는 실제적인 물건 또는 개념임

→ 객체는 속성과 기능을 가짐

1 | 객체 지향

02 속성과 기능

개념	설명	예시
객체	실생활에 존재하는 사물 또는 개념	축구 선수
속성	객체가 가지고 있는 변수	선수 이름, 팀 이름, 등번호
기능	객체가 실행하는 함수 (메소드)	공을 찬다, 달린다, 패스한다

1 | 객체 지향

03 객체 지향 프로그래밍(Object Oriented Programming, OOP)

- 객체의 개념을 활용하여 **프로그램으로 표현**하는 기법을 말함
- **재활용의 관점**으로 코드를 만들어 놓고 쓰는 방법임
 - 변수와 함수를 묶어서 **하나의 객체로 만들어 사용함**

➤ 함수처럼 어떤 기능을 함수 코드로 묶어 두는 것이 아니라, **특정 기능을 수행하는 하나의 단일 프로그램을 객체라고 하는 코드로 만들어** 다른 프로그래머가 재사용할 수 있도록 함

1 | 객체 지향

04 클래스(class)

- 객체는 하나의 프로그램에서 여러 개가 사용될 수도 있으므로 객체들을 위한 설계도를 만들어야 함

클래스(class)

객체가 가져야 할 기본 정보를 담은 코드, 일종의 설계도 코드

1 | 객체 지향

05 인스턴스(객체)

인스턴스(객체)

클래스를 기반으로 생성한 객체

1 | 객체 지향

05 인스턴스(객체)

- DOG 클래스



속성(변수)

품종, 크기, 나이, 색상



행동(기능)

먹는다, 잠을 잔다, 똥다

1 | 객체 지향

05 인스턴스(객체)

속성 (변수)	품종	breed	말티즈
	크기	size	small
	나이	age	2 year
	색상	color	white
행동 (기능)	먹는다	eat()	
	잠을 잔다	sleep()	
	뛰다	run()	

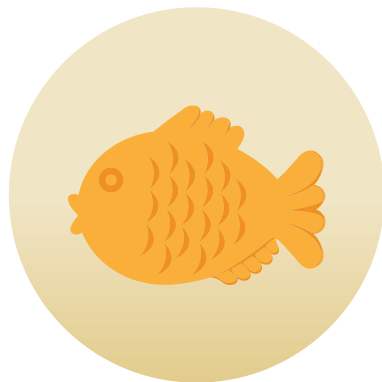
1 | 객체 지향

05 인스턴스(객체)

- 붕어빵 틀과 붕어빵



붕어빵 틀(클래스)



붕어빵(인스턴스)

02



파이썬 객체지향 프로그래밍



1 | 객체지향 프로그래밍

01 정의



객체지향 프로그래밍

객체를 생성할 수 있도록 클래스를 정의하는 것임

→ 클래스를 이용하여 객체를 생성하고 객체가 가지는 속성과 메소드를 이용하여 프로그래밍함

1 | 객체지향 프로그래밍

02 클래스를 정의하는 구조



- 예약어인 class를 코드의 맨 앞에 입력함
- 만들고자 하는 클래스 이름을 작성함
→ 이름을 작성할 때는 대문자로 시작함
- 상속받아야 하는 다른 클래스의 이름을 괄호 안에 넣음

1 | 객체지향 프로그래밍

03 축구선수 클래스 정의

- 속성 정의

 속성(변수) 선언 방법

> __init__() 예약 함수 사용

클래스에서 사용할 변수를
정의하는 함수

1 | 객체지향 프로그래밍

03 축구선수 클래스 정의

- 속성 정의

```
class FootballPlayer():  
    def __init__(self, name, team, back_number) :  
        self.name = name  
        self.team = team  
        self.back_number = back_number
```

1 | 객체지향 프로그래밍

04 self

- 객체의 그 자체를 의미함
 - 객체 자기 자신을 참조하는 매개변수임
- 다른 객체지향 언어는 self 안보이게 전달하지만, 파이썬은 클래스의 메소드를 정의할 때 self를 명시해서 정의함
- 메소드를 호출할 때 self는 자동으로 전달함
 - self를 사용함으로 클래스 내에 정의한 멤버에 접근할 수 있음

1 | 객체지향 프로그래밍

05 클래스 속성정의 예제

In [1]: **class** FootballPlayer():

속성정의

```
def __init__(self, name, team, back_number):  
    self.name = name  
    self.team = team  
    self.back_number = back_number
```

1 | 객체지향 프로그래밍

05 클래스 속성정의 예제

- 클래스를 이용한 객체 생성 후 사용

```
In [2]: p = FootballPlayer("King", "Sejong", 7)
```

```
In [3]: print(p.name)
         print(p.team)
         print(p.back_number)
```

King

Sejong

7

1 | 객체지향 프로그래밍

06 메소드 정의

메소드

클래스에서 정의하는 함수

→ 클래스 내부에 다양한 기능을 함수로 정의할 수 있음

클래스 내부에서 함수를 정의하는 방법은 동일함

차이점은 매개변수 앞에 self를 반드시 넣어야 함

1 | 객체지향 프로그래밍

06 메소드 정의

```
class FootballPlayer():  
    # 속성정의 생략  
  
    # 기능정의  
    def show_info(self) :  
        print('선수정보 =====')  
        print('이름', self.name)  
        print('팀', self.team)  
        print('등번호', self.back_number)
```

1 | 객체지향 프로그래밍

06 메소드 정의

- 클래스 메소드 정의 예제

In [5]: **class** FootballPlayer():

속성정의

```
def __init__(self, name, team, back_number):  
    self.name = name  
    self.team = team  
    self.back_number = back_number
```

1 | 객체지향 프로그래밍

06 메소드 정의

- 클래스 메소드 정의 예제

In [5]: # 기능정의

```
def show_info(self):  
    print('선수정보 =====')  
    print('이름', self.name)  
    print('팀', self.team)  
    print('등번호', self.back_number)
```

1 | 객체지향 프로그래밍

06 메소드 정의

- 클래스를 이용한 객체 생성 후 사용

```
In [6]: p = FootballPlayer("SON", "토틀넘", 7)
```

```
In [7]: p.show_info()
```

선수정보 =====

이름 SON

팀 토틀넘

등번호 7

1 | 객체지향 프로그래밍

06 메소드 정의

- `__str__(self)` 메소드

→ `print(객체이름)`을 사용해서 클래스 내부에서 정의한 정보를 출력할 때 정의해서 사용

```
class FootballPlayer():  
    # 속성정의 코드 생략  
    # 기능정의 코드 생략  
    def __str__(self):  
        return '{0} ( 팀:{1}, 등번호:{2} )'.format(self.name, self.team,  
self.back_number)
```


1 | 객체지향 프로그래밍

06 메소드 정의

- `__str__(self)` 메소드 정의 예제

In [12]: **class** FootballPlayer():

속성정의

```
def __init__(self, name, team, back_number):  
    self.name = name  
    self.team = team  
    self.back_number = back_number
```

1 | 객체지향 프로그래밍

06 메소드 정의

- `__str__(self)` 메소드 정의 예제

In [12]: # 기능정의

```
def show_info(self) :  
    print('선수정보 =====')  
    print('이름', self.name)  
    print('팀', self.team)  
    print('등번호', self.back_number)
```

1 | 객체지향 프로그래밍

06 메소드 정의

- `__str__(self)` 메소드 정의 예제

```
In [12]: def __str__(self):  
          return '{0} ( 팀:{1}, 등번호:{2} )'.format(self.name, self.team,  
          self.back_number)
```

1 | 객체지향 프로그래밍

06 메소드 정의

- 클래스를 이용한 객체 생성 후 사용

```
In [6]: p = FootballPlayer("SON", "토틀넘", 7)
```

```
In [7]: print(p)
```

SON (팀:토틀넘, 등번호:7)



파이썬기초

NEXT

객체지향의 필요성

