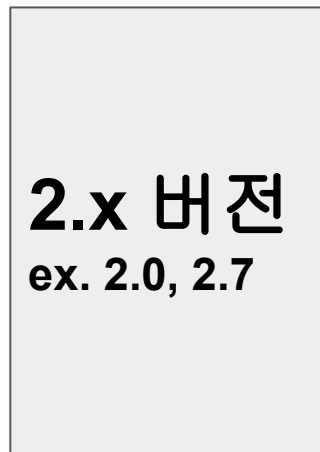




# 파이썬 기초 배우기

12/03 수업 정리

# 파이썬은 버전이 다양하다!



지원 종료 예정...



저희는 3.7.4 버전을  
사용하여 공부합니다!

# 파이썬 시작하기

## 소스의 형식

- 대문자 vs. 소문자 구분
- 주석은 #으로 표시(ctrl+/)
- 들여쓰기(4칸) 블록구분

## 출력

- `print(출력내용, sep=구분자, end=끝 문자)`

## 입력

- `변수=input('입력 받을 내용')`
- 입력 받을 내용은 문자열로 인식됨!

# 변수 네이밍 규칙

- 변수의 이름은 숫자로 시작할 수 없다
- **대문자와 소문자 구분**
- 스페이스를 포함하지 않고 특수문자를 제외한 알파벳 문자와 숫자, 언더스코어(\_)만 포함 가능

# 타입

## 1. 수치형 타입

- a. 정수형 타입(int), 실수형 타입(float)
- b. 진법: 16진법, 8진법, 2진법으로 변경할 수 있음
  - i. ex. `print(hex(26))` `print(oct(16))` `print(bin(13))`

## 2. 문자열 타입

- a. 일련의 문자를 따옴표(“”)로 감싸 나열한 것
- b. 큰따옴표로 시작했으면 큰따옴표로 마무리, 작은따옴표로 시작했으면 작은따옴표로 마무리!!
- c. 문자열 안에 따옴표 넣고 싶으면 밖에 쓴 것과 다른 종류로 쓴다
- d. 확장열
  - i. `\ + ?(ex. n, t, ‘,’, \)`
  - ii. 확장열 사용하고 싶지 않으면 문자열 앞에 `r` 접두 붙이는 것도 가능

## 3. Bool 타입

## 4. 컬렉션 타입

# 타입

1. 수치형 타입

2. 문자열 타입

3. **Bool** 타입

a. True / False

4. **컬렉션** 타입

a. 여러 개의 값을 모아서 저장하는 데이터 타입

b. 리스트 ([]), 튜플 (())

c. 튜플은 값을 변경할 수 없다

```
member=["손오공","저팔계","삼장법사","사오정"]  
print(member)  
member[0]="우마왕"  
print(member)  
member[-1]="돈오공"  
print(member)
```

```
C:\python_workspaces\hello_world2\venv\Scripts\python  
.exe C:/python_workspaces/hello_world2/input_.py
```

```
['손오공', '저팔계', '삼장법사', '사오정']
```

```
['우마왕', '저팔계', '삼장법사', '사오정']
```

```
['우마왕', '저팔계', '삼장법사', '돈오공']
```

Process finished with exit code 0

# 연산자

## 1. 산술연산자

a. +, -, \*, /

## 2. 고급 산술 연산자

a. \*(거듭제곱), //(정수나누기), %(나머지)

b. 복합 대입 연산자

`a = 5`

`a = a + 1`

`print(a)`

`a = 5`

`a += 1`

`print(a)`

## 3. 문자열 연산자

## 4. 연산자 우선순위와 결합순서

## 5. 연산자의 종류

# 연산자

1. 산술연산자

2. 고급 산술 연산자

3. 문자열 연산자

a. `print("만세"*5)` ⇒ 만세만세만세만세만세

b. 정수와 문자열은 섞어서 사용할 수 없다

i. 정수는 **str 함수** 사용하여 문자열로 변경

ii. 문자열은 **int 함수** 사용하여 정수로 변경

iii. 문자열은 **float 함수** 사용하여 실수로 변경

iv. 실수는 **int 함수** 사용하여 정수로 변경

v. 반올림 하기 위해서는 **round 함수** 사용

4. 연산자 우선순위와 결합순서

5. 연산자의 종류



# 연산자

1. 산술연산자
2. 고급 산술 연산자
3. 문자열 연산자

4. 연산자 우선순위와 결합순서
  - a. 곱하기, 나누기 먼저, 괄호 먼저

## 5. 연산자의 종류

연산자	설명
() [] {}	컬렉션 표현식
A[i] func() obj.attr	첨자 연산, 함수 호출, 멤버 읽기
**	거듭제곱
+ - ~	부호 연산자, 논리 부정 연산자
<< >>	쉬프트 연산
&	비트 논리곱
^	비트 배타적 논리합
	비트 논리합

연산자	설명
In, not in, is, is not, < <=> >= != ==	비교 연산자
not	논리 부정
and	논리곱
or	논리합
If else	조건 점검
lambda	람다 표현식

# 조건문

## 1. if 문

a. if 조건:

명령

## 2. 비교 연산자

- a. 비교연산자는 두 값의 상등 여부나 대소관계를 비교하여 참(True), 거짓(False)을 리턴한다
- b. ==(같다), !=(다르다), <=(좌변이 우변보다 작거나 같다)
- c. 문자열 비교: 대소문자 구별, 사전의 뒤쪽에 나오는 문자열을 큰 것으로 인식

## 3. 논리 연산자

## 4. 블록 구조

## 5. else 문

## 6. elif 문

## 7. if 문 중첩

# 조건문

1. if 문

2. 비교 연산자

**3. 논리 연산자**

- a. and: 두 조건이 모두 참
- b. or: 두 조건 중 하나라도 참
- c. not: 조건을 반대로 뒤집음

**4. 블록 구조**

- a. 조건문이 참일 때 실행할 명령이 2개 이상이면 아래쪽에 명령을 계속 나열
- b. 들여쓰기 수준은 같아야 함

5. else 문

6. elif 문

7. if 문 중첩

# 조건문

1. if 문
2. 비교 연산자
3. 논리 연산자
4. 블록 구조

## 5. else 문

## 6. elif 문

- a. elif는 여러 개가 들어갈 수 있지만,

if 문과 else 문은 한개만 들어갈 수 있다.

## 7. if 문 중첩

```
age = 22
if age < 19:
    print("애들은 가라")
    print("공부 열심히 해야지")
else:
    print("어서 옹쇼")
    print("즐거운 시간 되세요")
```

```
money = 7000
if money >= 20000:
    print("탕수육을 먹는다.")
else:
    if money >= 10000:
        print("쟁반 짜장을 먹는다.")
    else:
        if money >= 6000:
            print("짬뽕을 먹는다.")
        else:
            if money >= 4000:
                print("짜장면을 먹는다.")
            else:
                print("단무지를 먹는다.")
```

감사합니다