

파이썬 출력 이해

```
print("파이썬 웰컴") print("파이썬", "웰컴", "python") print("파이썬 \n",
"웰컴\n") print('파이썬 "매우" 웰컴') print(""" 굉장히 긴 문자열입니
다..... 파이썬 화이팅 파이썬자동화 """)
```

파이썬에서 사용되는 이스케이프 코드(Escape Codes)는 문자열 내에서 특정 문자를 표시하거나 특수 문자를 삽입할 때 사용

이스케이프 코드	설명	예시
<code>\\</code>	백슬래시 문자	<code>\\</code> → <code>\</code> c:\\windows\\
<code>\'</code>	작은따옴표	<code>\'</code> → <code>'</code>
<code>\"</code>	큰따옴표	<code>\"</code> → <code>"</code>
<code>\n</code>	줄바꿈 (개행)	<code>Hello\nWorld</code> → HelloWorld
<code>\t</code>	탭	<code>Hello\tWorld</code> → Hello World
<code>\r</code>	캐리지 리턴	<code>Hello\rWorld</code> → World
<code>\b</code>	백스페이스	<code>Hello\bWorld</code> → HellWorld
<code>\f</code>	폼 피드 (페이지 나누기)	텍스트 페이지 나누기
<code>\a</code>	벨 (경고음)	소리 발생 (일부 환경에서만)
<code>\v</code>	수직 탭	수직으로 탭
<code>\0</code>	널 문자	문자열 끝을 표시하는 데 사용
<code>\ooo</code>	8진수 값	<code>\101</code> → A
<code>\xhh</code>	16진수 값	<code>\x41</code> → A

```
name = input("이름을 입력하세요.") phone = input("번호를 입력하세요.") age = input("나이를 입력하세요.") print(name) print(phone) print(name, "의 전화번호는", phone, "입니다.", "나이는", age) #print(type(name)) #print(type(phone))
```

문자열 포매팅 정의

- Python에서 문자열 포매팅은 데이터를 문자열 형태로 변환하고, 이를 출력하거나 다른 문자열과 결합하기 위해 사용
- 변수나 계산 결과를 문자열에 삽입하고, 이를 원하는 형식으로 표현
- 파이썬의 세 가지 주요 문자열 포매팅 방식인 `%` 연산자, `.format()` 메소드, 그리고 `f-string`에 대해 자세히 배울 예정

실습

1. % 연산자 (지금 사용 안함)

- `%` 연산자를 이용한 포매팅
- 파이썬 초기 버전부터 사용되어온 방식 - 가장 오래된 방식 중 하나
- `%s`는 문자열, `%d`는 정수, `%f` 부동 소수점 숫자를 의미

```
name = "조정원" print("내 이름은 %s입니다." % name)
```

- `%` 연산자 뒤에 나오는 튜플 `(name, age)`의 각 요소로 대체

```
name = "조정원" age = 20 print("내 이름은 %s이고 나이는 %d살입니다." % (name, age))
```

2. format() 메서드

- `format()` 메서드를 이용한 포매팅
- 파이썬 2.6부터 도입된 방식
- 문자열 내의 중괄호 `{}`를 플레이스홀더로 사용하여 이 플레이스홀더들은 `.format()`에 전달된 인자로 대체

```
name = "조정원" age = 20 print("내 이름은 {}이고 나이는 {}살입니다.".format(name, age))
```

- 인덱스 사용하여 특정 순서로 인자 삽입

```
name = "조정원" age = 20 print("내 이름은 {}이고 나이는 {}살입니다.".format(age, name))
```

```
print("내 이름은 {1}이고 나이는 {0}살입니다.".format(age, name))
```

- 키워드 인자 사용

```
name = "조정원" age = 20 print("내 이름은 {n}이고 나이는 {a}살입니다.".format(a=age, n=name))
```

3. 포맷 문자열 리터럴(f-string) - 가장 최신 기능

- 파이썬 3.6에서 도입된 f-string 방식
- f-string은 문자열 포매팅을 위한 가장 직관적이고 현대적인 문자열 포매팅 방식
- f-strings를 사용하는 이 코드에서는 문자열 앞에 `f`를 붙여 포맷 문자열 리터럴을 나타냄
- `{}` 안에 직접 변수 `name`을 삽입하여, Python이 이를 `name` 변수의 값인 "김태영"으로 대체

```
name = "조정원" age = 20 print(f"내 이름은 {name}이고 나이는 {age}살입니다.")
```

결론적으로 3가지 방식으로 출력이 가능하나, f-string을 사용한다.

```
name = input("이름을 입력") phone = input("번호를 입력") age = int(input("나이를 입력"))
print(name, "의 전화번호는", phone, "입니다.", "나이는", age)
print("내 이름은 {}이고 나이는 {}살입니다.".format(name,age))
print(f"내 이름은 {name}이고 나이는 {age}살입니다.")
print(type(name))
print(type(age))
```

pip install deep_translator

설치가 안된다면

python.exe -m pip install deep_translator

```
from deep_translator import GoogleTranslator
input_text = input("한글을 입력하세요. ")
translated = GoogleTranslator(source='ko', target='en').translate(input_text)
print(f"입력한 한글 : {input_text}")
print(f"번역된 영어: {translated}")
```

```
languages = {
    'en': 'English',
    'es': 'Spanish',
    'fr': 'French',
    'ja': 'Japanese',
    'zh-CN': 'Chinese (Simplified)'
}
```

