


# 1주차 정리

 소유자	 jiwun kim
 태그	

## Print()

- 파이썬은 print() 함수를 사용해서 콘솔에 출력이 가능합니다.

**""" 혹은 " 사용해서 문자를 출력할 수 있습니다.**

```
print("Hello World")
print('Heelo World')
```

**숫자를 넣어서 숫자를 출력할 수 있습니다.**

```
print(3)
print(5.5)
print(-2)
```

**"," 를 사용해서 숫자 2개 이상의 값을 출력 가능합니다.**

```
print(3, 5, 6)
print(3, 5, "5")
```

**\n을 사용해서 한 줄 띄어쓰기가 가능합니다.**

```
print("Hello\nWorld")
```

**sep를 사용해서 "," 구분을 공백 대신 다른 값으로 대체 가능합니다.**

- sep 값을 설정 하지 않으면 기본값으로 공백입니다.

```
print(3, 5, 6, sep="")
print(3, 5, 6, sep=", ")
print(3, 5, 6, sep="- ")
print(3, 5, 6, sep="/")
print(3, 5, 6, sep=" ")
```

**end를 사용해서 print() 시 문자 마지막에 다른 값을 추가할 수 있습니다.**

- end를 사용하지 않으면 기본적으로 \n 가 기본적으로 적용 됩니다.

```
print("Hello World", end="@")
print("Hello World")

print("Hello World", end="- ")
print("Hello World")

print("Hello World", end="")
print("Hello World")
```

## 변수란?

- 특정한 데이터를 저장하는 저장 공간 입니다.

## 파이썬에서 변수 사용하는 방법

- 변수\_이름 = 저장할 값
- "=" 는 대입 연산자로 오른쪽에 있는 걸 왼쪽에 저장한다는 뜻입니다.
- 아래는 num1 과 msg에 각각의 데이터를 저장하고 print() 를 사용해서 값을 출력하는 코드 입니다.

```
num1 = 5
msg = "안녕하세요"
```

```
print(num1)
print(msg)
```

## 자료형이란?

- 프로그래밍을 할 때 쓰이는 숫자, 문자열 등과 같이 자료 형태로 사용하는 모든 것을 의미
- 프로그램의 기본이자 핵심 단위

## 숫자형 - link

- 숫자 형태로 이루어진 자료형
- 123 같은 정수와 12.34 같은 실수, 8진수 16진수등이 있다.

항목	파이썬 사용 예
정수	123, -345, 0
실수	123.45, -1234.5, 3.4e10
8진수	0o34, 0o25
16진수	0x2A, 0xFF

## 정수형

- 정수를 뜻하는 자료형

```
a = 123
a = -1751231
a = 0
c = 1231321321
```

## 실수형

- 소수를 뜻하는 자료형

```
a = 1.2
a = -3.14
```

## 사칙 연산 사용해보기

연산	뜻
+	더하기
-	빼기
*	곱하기
/	나눗셈 후
//	나눗셈 후 몫 (정수만 리)
%	나눗셈 후 나머지

```
print("3 + 5 = ", 3 + 5)
print("3 - 5 = ", 3 - 5)
print("3 * 5 = ", 3 * 5)
print("3 / 5 = ", 3 / 5)
print("5 / 3 = ", 5 / 3)
print("3 // 5 = ", 3 // 5)
print("3 % 5 = ", 3 % 5)
print("6 % 3 = ", 6 % 3)
print("3 % 6 = ", 3 % 6)
```

## 제곱 표현하기

- \*\* 를 사용해서 제곱을 표현할수 있습니다.
- 아래는 3의 4제곱을 표현한 예시 입니다.

```
print("3의 4제곱 =", 3 ** 4)
```

## 복합 연산자 표현

- +, - 같은 산술 연산자와 "=" 대입 연산자를 합친 것 입니다.

```
num1 = 10
num1 += 10
print(num1)
```

```
num1 = 10
num1 -= 10
print(num1)
```

```
num1 = 10
num1 += 10
print(num1)
```

```
num1 = 10
num1 *= 10
print(num1)
```

- 위 예시에 나오것 말고도 /=, //=, %=, \*\*= 같은 것들이 있습니다.

---

## 문자형 - Link

**" 혹은 '로 감싸서 문자를 표현할수 있습니다.**

- 위 print() 설명 참고

```
print("Hello World")
print('Heelo World')
```

**" 혹은 '를 연속으로 3개 사용해서 문자를 표현할 수도 있습니다.**

```
print("""Hello World!""")
print(''Hello World!'')
```

**“혹은 ’를 출력하고 싶을때는 서로 다른 따표로 감싸면 됩니다.**

```
print("'Hello World!'")
print("Hello's World!")
print('Hello "W"orld!')
print("""'Hello World!'""")
print(''Hello World!'')
```

**\를 사용해서도 “ 혹은 ’를 표시할 수 있습니다.**

```
print("Hello\' World!")
print("Hello\" World!")
```

**\\를 사용하면 \를 출력할 수 있습니다.**

```
print("Hello\\s World!")
```

**변수를 사용한 문자 출력**

```
msg1 = "안녕하세"
print(msg1)

msg2 = "안녕\n하세용"
print(msg2)

multiline = '''
Life is too short
You need python
'''
print(multiline)
```

**이스케이프 코드**

코드	설명
<code>\n</code>	문자열 안에서 줄을 바꿀 때 사용
<code>\t</code>	문자열 사이에 탭 간격을 줄 때 사용
<code>\\</code>	<code>\</code> 를 그대로 표현할 때 사용
<code>\'</code>	작은따옴표(')를 그대로 표현할 때 사용
<code>\"</code>	큰따옴표(")를 그대로 표현할 때 사용
<code>\r</code>	캐리지 리턴(줄 바꿈 문자, 커서를 현재 줄의 가장 앞으로 이동)
<code>\f</code>	폼 피드(줄 바꿈 문자, 커서를 현재 줄의 다음 줄로 이동)
<code>\a</code>	벨 소리(출력할 때 PC 스피커에서 '뽕' 소리가 난다)
<code>\b</code>	백 스페이스
<code>\000</code>	널 문자

- `\n` 은 응근 자주사용하니 기억하기!

## 문자열 곱하기

- `*`를 사용해서 문자를 곱해서 표시해 줄 수 있습니다.

```
msg = "으으 귀찮아!"
print(msg * 2)
```

```
msg = "으으 귀찮아!"
msg *= 2
print(msg)
```

## 문자열 길이 구하기

- `len()` 를 사용해서 문자열 길이를 구할 수 있습니다
- `len()`은 공백을 포함합니다.

```
msg = "길이알려줘"
print(len(msg))
```

```
msg = "길이 알려줘"
print(len(msg))
```

## 문자열 인덱싱

- 문자열중 개별 문자에 접근하기 위해 사용
- 인덱스는 0부터 시작해서 문자 최대길이 -1 까지 입니다.
- "Life is too short, You need Python" 문자열의 인덱스 번호 입니다.

L	i	f	e		i	s		t	o	o		s	h	o	r	t	,		Y	o	u		n	e	e	d		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33

```
text = "Hello, World!"  
print(text[0]) # 'H'  
print(text[7]) # 'W'
```

음수를 사용하면 뒤에서부터 문자를 가져옵니다.

```
text = "Hello, World!"  
print(text[-1]) # '!'  
print(text[-4]) # 'r'
```

## 문자열 슬라이싱

- 슬라이싱을 사용하면 문자열을 잘라서 사용할 수 있습니다.
- "start index:end index" 을 사용해서 문자열 슬라이싱이 가능합니다.
- 끝 인덱스 -1 까지 표시해 줍니다.

```
msg = "Hello World"  
print(msg[1:4])
```

":End index" 를 하면 0번 부터 end index -1 까지 표시합니다.



```
msg = "Hello World"
print(msg[:4])
```

**"Start Index:" 를 하면 Start Index 부터 문자 끝까지 표시합니다.**

```
msg = "Hello World"
print(msg[2:])
```

## 문자열 포매팅

코드	설
%s	문자열
%c	문자 1개
%d	정수
%f	부동 소수
%o	8진수
%x	16진수
%%	Literal % (문자 % 자체)

## 문자열에 숫자 대입하기

```
msg = "나는 %d살이야" % 3
print(msg)
```

## 2개 이상 대입하기

```
msg = "나 %s는 %d살이야" % ("아무개는", 3)
print(msg)
```

**%d 와 %를 같이 쓸 때는 %%를 써야 합니다.**

```
msg = "오늘 주식 수익률은 %d%%야" % 25
print(msg)
```

## 소수점 자리수 제한

- f앞에 **.제한할자리수** 입력

```
msg = "오늘 주식 수익률은 %.2f야" % 31.22131
print(msg)
```

## f문자열 포매팅

- f 문자열 포매팅은 파이썬 3.6 이상부터 사용가능합니다.
- 문자열 앞에 f를 추가하고 {} 안에 변수 및 각종 연산등을 할 수 있습니다.

```
age = 98
msg = f"나는 {age}살이야"
print(msg)

msg = f"나는 {3 * 2}살이야"
print(msg)
```