

파이썬 프로그래밍

---

# 파이썬 언어의 기본 문형



한국기술교육대학교  
온라인평생교육원

---

## ■ 파이썬 예약어 및 내장 함수

### 1. 예약어

- 예약어 (또는 키워드)

- 파이썬에서 이미 문법적인 용도로 사용되고 있기 때문에 변수명 등의 식별자로 사용하면 안 되는 단어들

- 예약어는 Reserved(예약된) Words 또는 키워드라고 함
- 파이썬에서 이미 사용되고 있는(용도가 예약된) 단어들
- 이미 문법적인 용도로 사용 → 신택스(Syntax)
- 코딩할 때 활용할 예약어를 변수에 활용하면 안됨
- 예약어를 변수에 활용 → 에러는 없으나 고유 기능은 사라짐

## ■ 파이썬 예약어 및 내장 함수

### 2. 예약어의 종류 알아보기

```
import keyword
print keyword.kwlist
print
print len(keyword.kwlist)
```

```
['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'exec',
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print',
'raise', 'return', 'try', 'while', 'with', 'yield']
```

- 파이썬의 예약어 종류를 알아보는 방법은?
- keyword 모듈을 불러옴(import)
- keyword 모듈이 지원하는 kwlist를 출력(print)
- 이 때 print도 예약어이므로 다른 식별자로 사용하면 안됨
- print만 실행 → 한 줄을 띄워줌
- len → 특별한 모듈 추가 없이 사용할 수 있는 내장 함수
- keyword 모듈에 있는 kwlist에 몇 개의 단어가 있는지 알아봄
- 프로그램을 수행하면 콘솔에 수행결과가 나타남
- 수행결과는 문자열이 나열된 리스트 형태
- 리스트 안에 나열된 문자들 → 예약어
- 예약어의 개수는 31개(len 함수의 결과)
- 파이썬의 버전에 따라 예약어 종류 달라짐
- 파이썬 버전2.7에서는 31개 예약어 지원함
- 앞서 알아본 len이 내장 함수(Built-in Function)
- import keyword에서 keyword가 모듈
- 모듈을 불러올 때는 import(내장 함수) 사용
- 현재 프로그램에 사용된 예약어 → import, print
- keyword 모듈은 기본 인터프리터 환경에 추가 안됨
- 파이썬을 설치하면 수많은 모듈이 설치됨
- import keyword → 그 중 keyword 모듈을 사용하겠다
- 모듈 안에 있는 함수는 모듈을 import 해야 사용 가능

## ▣ 파이썬 예약어 및 내장 함수

### 3. 내장 함수

- 별도의 모듈(Module)의 추가 없이 기본적으로 제공되는 함수들
- 참고사이트
  - 내장(Built-in) 함수: <http://docs.python.org/2/library/functions.html>
- 대표적인 내장 함수
  - abs, max, min, pow, chr, str, range, type, ...
- abs(x) : 수치형 자료 x에 대해 x의 절대값을 반환하는 함수

- 내장 함수는 모듈 추가 없이 활용 가능
- 파이썬 공식 홈페이지의 내장 함수 안내 페이지
- abs → 수치형 자료를 절대값으로 반환하는 내장 함수

```
print abs(3)
print abs(-3)
```

```
3
3
```

- max(s)
  - 시퀀스 자료형(문자열, 리스트, 튜플)을 입력받아 그 자료가 지닌 원소 중 최대값을 반환하는 함수

- max → 주어진 자료 중 최대값을 반환하는 내장 함수

## ■ 파이썬 예약어 및 내장 함수

### 3. 내장 함수

```
print max(1,2)
print max([1, 2, 3])
print max("python")
```

```
2
3
y
```

- `print max(1, 2)` → 1과 2 중 더 큰 2를 반환
- `print max([1, 2, 3])` → 리스트의 원소 중 제일 큰 3을 반환
- `print max("python")` → 문자열 중 아스키 코드값이 가장 큰 문자 반환
- `max` → 주어진 자료 중 최소값을 반환하는 내장 함수

- `min(s)`
  - 시퀀스 자료형(문자열, 리스트, 튜플)을 입력받아 그 자료가 지닌 원소 중 최소값을 반환하는 함수

```
print min(1,2)
print min([1, 2, 3])
print min("python")
```

```
1
1
h
```

- `print min(1, 2)` → 1과 2 중 더 작은 1을 반환
- `print min("python")` → 문자열 중 아스키 코드값이 가장 작은 문자 반환

## ■ 파이썬 예약어 및 내장 함수

### 3. 내장 함수

- `pow(x,y)`
  - 수치형 자료형 `x`, `y`에 대해 `x`의 `y`승을 반환하는 함수

- `pow(a, b) → a의 b승 값을 반환`

```
print pow(2,4)
print pow(3, 3)
print pow(2, -1)
```

```
16
27
0.5
```

- `print pow(2, 4) → 2의 4승, 즉, 16을 출력`
- `print pow(3, 3) → 3의 3승, 즉, 27을 출력`
- `print pow(2, -1) → 2의 -1승, 즉, 0.5를 출력`

- `chr(i)`
  - 정수 형태의 ASCII코드 값을 입력으로 받아 그에 해당하는 문자를 반환하는 함수
  - 인수 `i`의 범위: 0부터 255까지

- `chr → 아스키 코드값을 문자로 변환해 주는 함수`

## ■ 파이썬 예약어 및 내장 함수

### 3. 내장 함수

```
print chr(97)
print chr(65)
print chr(48)
```

```
a
A
0
```

- `print chr(97)` → 아스키 코드값이 97인 문자 → a
- `print chr(65)` → 아스키 코드값이 65인 문자 → A
- `print chr(48)` → 아스키 코드값이 48인 문자 → 0
- 이 때 0은 숫자 0이 아니라 문자로써의 0

- `str(object)`
  - 임의의 객체 `object`에 대해 해당 객체를 표현하는 문자열을 반환하는 함수

- `str(a)` → a를 출력(`print`)하면 어떻게 나타나는지를 반환

```
print str(3)
print str([1, 2])
```

```
3
[1, 2]
```

- `str(3)` → 3이라는 객체를 출력했을 때 나타나는 결과 → 3
- `str([1,2])` → [1, 2]를 출력했을 때 나타나는 결과 → [1, 2]
- `str` → 해당 객체를 표현하는 문자열로 반환해주는 함수

## ■ 파이썬 예약어 및 내장 함수

### 3. 내장 함수

- `range([start,]stop[,step])`
  - 수치형 자료형으로 `start`, `stop`, `step` 등을 입력받아 해당 범위에 해당하는 정수를 리스트로 반환하는 함수
- 인수가 하나(`stop`)인 경우
  - 0부터 `stop-1`까지의 정수 리스트를 반환한다.
- 인수가 두 개(`start`, `stop`)인 경우
  - `start`부터 `stop-1`까지의 정수 리스트를 반환한다
- 인수가 세 개(`start`, `stop`, `step`)인 경우
  - `start`부터 `stop-1`까지의 정수를 반환하되 각 정수 사이의 거리가 `step`인 것들만 반환한다.

```
print range(10)
print range(3, 10)
print range(3, 10, 2)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[3, 4, 5, 6, 7, 8, 9]
[3, 5, 7, 9]
```

- `range(10)` → 0부터 10-1까지의 정수 리스트를 반환
- `range(3, 10)` → 3부터 10-1까지의 정수 리스트를 반환
- `range(3, 10, 2)` → 3부터 10-1까지 2개씩 건너 뛴 정수 리스트를 반환
- `range(3, 10, 2)` → [3, 5, 7, 9]
- $5 - 3 = 2(\text{step})$  → 앞, 뒤의 차이가 2씩 나게 함
- $7 - 5 = 2(\text{step})$  → 앞, 뒤의 차이가 2씩 나게 함
- $9 - 7 = 2(\text{step})$  → 앞, 뒤의 차이가 2씩 나게 함
- `range` 함수는 각각 원소의 차이가 `step`값만큼 나도록 할 수 있음
- `range([start], stop, [step])` → `start`와 `step`은 생략 가능
- `range(10)` → `stop`만 10으로 기술한 것
- `range(10)` → 10-1까지의 정수 리스트를 반환
- `start`가 생략되었을 경우 0부터 시작
- `range(a, b, c)` → `a`부터 `b-1`까지 `c`씩 차이 나게 정수 리스트를 반환



---

## ■ 파이썬 예약어 및 내장 함수

### 3. 내장 함수

```
print type(-1)
print type('abc')
print type([1, 2, 3])
```

```
<type 'int'>
<type 'str'>
<type 'list'>
```

- `type(a)` → `a`의 자료형을 반환
- `print type(-1)` → `-1`의 자료형을 반환 → `int`(정수형)
- `print type('abc')` → `'abc'`의 자료형을 반환 → `str`(문자형)
- `print type([1, 2, 3])` → `[1, 2, 3]`의 자료형을 반환 → `list`(리스트)
- `type` 함수를 사용하여 객체의 자료형을 알 수 있음