

파이썬과 코딩테스트

01 파이썬 주요 문법 Review

02 좋은 습관/나쁜 습관

신 제 용



01 파이썬 주요 문법 Review

주요 파이썬 문법을 Review하는 시간을 가져봅시다.

학습 키워드 – 파이썬, 프로그래밍, 프로그래밍 언어

Chapter 01

파이썬 주요 문법
Review

Python 주요 문법

- 코딩테스트에서 자주 활용되는 주요 문법 Review
- Pythonic한 코드를 작성하는 데에 필수적인 지식
- 높은 가독성의 코드를 빠르게 작성할 수 있다.
- 혹시 기억나지 않거나 배우지 않았다면 알아둡시다!

Chapter 01
파이썬 주요 문법
Review

다양한 방식의 for 문 (1)

- range를 이용한 for 문과 인덱싱

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for i in range(len(a)):
    print(a[i])
```

- 반복자를 이용한 for 문

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for elem in a:
    print(elem)
```

Chapter 01

파이썬 주요 문법
Review

다양한 방식의 for 문 (2)

- enumerate를 이용한 인덱싱 + 반복자

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for i, elem in enumerate(a):
    print(f'a[{i}] = {elem}')
```

- zip을 이용한 여러 반복자 동시 반복

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
b = [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
for e1, e2 in zip(a, b):
    print(f'{e1}, {e2}')
```

Chapter 01

파이썬 주요 문법
Review



함수의 구현

- 기본적인 함수 정의

```
def solution(x, y):  
    return x + y
```

Chapter 01

파이썬 주요 문법
Review

지능형 리스트 (List comprehension)

- for문에 비해 빠르게 동작하는 장점이 있다.
 - for문과 append (가장 나쁜 방법)

```
a = []  
for i in range(10):  
    a.append(i)
```

- Pre-allocation (조금 나은 방법)

```
a = [0]*10  
for i in range(10):  
    a[i] = i
```

- List comprehension (가장 좋은 방법)

```
a = [x for x in range(10)]
```

Chapter 01

파이썬 주요 문법
Review

지능형 리스트의 친구들

- Generator – 미리 값을 만들지 않으므로 메모리를 적게 쓴다.

```
a = (x for x in range(100000))
```

- Dictionary comprehension – 딕셔너리도 지능형 리스트처럼!

```
a = {key:key**2 for key in range(10)}
```

Chapter 01

파이썬 주요 문법
Review

람다 표현식 (Lambda expression)

- 함수 프로그래밍을 하기 위한 핵심 문법

```
print((lambda x: x**2)(10))
```

조건 표현식 (Conditional expression)

- if 문 사용을 줄이는 핵심 문법

```
b = 20  
a = 1 if b > 0 else -1
```

Chapter 01

파이썬 주요 문법
Review

많이 쓰이는 함수 프로그래밍

- map 함수

```
a = [x for x in range(10)]  
a = list(map(lambda x: x**10, a))
```

- filter 함수

```
a = [x for x in range(100)]  
a = list(filter(lambda x: x % 2, a))
```

- reduce 함수

```
from functools import reduce  
a = [x for x in range(100)]  
a = reduce(lambda x, y: x + y, a, 0)
```

Chapter 01

파이썬 주요 문법
Review

클래스 구현

```
class Foo:
    cls_var = 10  # 클래스 멤버 변수

    def __init__(self, x):
        self.x = x  # 인스턴스 멤버 변수

    def method(self, y):  # 인스턴스 메소드
        return self.x + y

    @classmethod
    def cls_method(cls, z):  # 클래스 메소드
        return cls.cls_var + z

    @staticmethod
    def static_method(k):  # 스태틱 메소드
        return k
```

Chapter 01

파이썬 주요 문법 Review

02 좋은 습관/나쁜 습관

다음 챕터에는 파이썬 코딩에서 좋은 습관과 나쁜 습관에 대해 알아봅니다.

Chapter 01

파이썬 주요 문법
Review

02 좋은 습관/나쁜 습관

코드를 작성하는 것은 어렵지 않지만, 좋은 코드를 작성하는 습관을 들이기는 어렵습니다. 좋은 습관을 쌓아가기 위한 팁을 학습합니다.

학습 키워드 – 코드, 코드 퀄리티, 코드 컨벤션

Chapter 02

좋은 습관/나쁜 습관

좋은 습관

- 변수 명을 정할 때에는 의미가 있는 이름으로 결정
- 변수, 클래스 명 등은 컨벤션을 지켜서 일정하게 구현
(ClassName, method_name, var_name, CONST_NAME, ...)
- PEP-8을 지켜서 가독성 좋은 코드 작성
- map, filter, reduce 등을 이용하여 함수형 프로그래밍을 적절하게 채용

Chapter 02

좋은 습관/나쁜 습관

나쁜 습관

- C언어처럼 구현하는 습관 (항상 for문으로 모든 것을 처리)
Python은 for문으로 처리할 경우 속도가 느리고 코드의 가독성이 떨어짐
- 신기한 코드
나만 알아볼 수 있는 신기한 코드는 적절하지 않음
- 무조건 짧은 코드
Short code 자체도 의미는 있으나, 일반적으로는 사용하지 않음
- 무조건 내가 구현
기본 파이썬 라이브러리에 구현된 것은 import해서 쓰는 편이 좋다.
- 무조건 최적화
최적화 vs. 가독성 사이에 trade-off 관계가 있을 경우, 상황에 맞게 결정한다.

Chapter 02

좋은 습관/나쁜 습관