

PEP 8 스타일 가이드를 따르자

파이썬 개선 제안서(Python Enhancement Proposal) #8, 다른 말로 PEP 8은 파이썬 코드를 어떻게 구성할지 알려주는 스타일 가이드다. 문법만 잘 지킨다면 파이썬 코드를 맘대로 작성해도 괜찮다. 하지만 일관성 있는 스타일을 사용하면 유지보수가 더욱 쉬워지고 가독성도 높아진다. 더 큰 커뮤니티에 속한 다른 파이썬 프로그래머들과 공통된 스타일을 공유하면 다양한 프로젝트에서 협업도 가능하다. 하지만 작성한 코드를 읽는 사람이 자신뿐이라고 해도 스타일 가이드를 따르면 나중에 수정하기가 더 쉬울 것이다.

PEP 8에는 파이썬 코드를 명확하게 작성하는 방법이 자세히 나와 있다. 파이썬 언어가 진화하는 데 맞춰 PEP 8도 지속적으로 업데이트될 것이다. 전체 가이드는 온라인(<https://www.python.org/dev/peps/pep-0008/>)에서 볼 수 있다. 반드시 따라야 하는 몇 가지 규칙은 다음과 같다.

화이트스페이스(whitespace): 파이썬에서 화이트스페이스(공백)는 문법적으로 의미가 있다. 파이썬 프로그래머는 특히 코드의 명료성 때문에 화이트스페이스의 영향에 민감한 편이다.

- 탭이 아닌 스페이스로 들여쓴다.
- 문법적으로 의미 있는 들여쓰기는 각 수준마다 스페이스 네 개를 사용한다.
- 한 줄의 문자 길이가 79자 이하여야 한다.
- 표현식이 길어서 다음 줄로 이어지면 일반적인 들여쓰기 수준에 추가로 스페이스 네 개를 사용한다.
- 파일에서 함수와 클래스는 빈 줄 두 개로 구분해야 한다.
- 클래스에서 메서드는 빈 줄 하나로 구분해야 한다.
- 리스트 인덱스, 함수 호출, 키워드 인수 할당에는 스페이스를 사용하지 않는다.
- 변수 할당 앞뒤에 스페이스를 하나만 사용한다.

명명(naming): PEP 8은 언어의 부분별로 독자적인 명명 스타일을 제안한다. 이 스타일을 따르면 코드를 읽을 때 각 이름에 대응하는 타입을 구별하기 쉽다.

- 함수, 변수, 속성은 lowercase_underscore 형식을 따른다.
- 보호(protected) 인스턴스 속성은 _leading_underscore 형식을 따른다.
- 비공개(private) 인스턴스 속성은 _double_leading_underscore 형식을 따른다.
- 클래스와 예외는 CapitalizedWord 형식을 따른다.
- 모듈 수준 상수는 ALL_CAPS 형식을 따른다.
- 클래스의 인스턴스 메서드에서는 첫 번째 파라미터(해당 객체를 참조)의 이름을 self로 지정한다.
- 클래스 메서드에서는 첫 번째 파라미터(해당 클래스를 참조)의 이름을 cls로 지정한다.

표현식과 문장: 파이썬의 계명(The Zen of Python)에는 “어떤 일을 하는 확실한 방법이 (될 수 있으면 하나만) 있어야 한다.”는 표현이 있다. PEP 8은 표현식과 문장의 본보기로 이 스타일을 정리하고 있다.

- 긍정 표현식의 부정(if not a is b) 대신에 인라인 부정(if a is not b)을 사용한다.
- 길이를 확인(if len(somelist) == 0)하여 빈 값([] 또는 "")을 확인하지 않는다. if not somelist를 사용하고, 빈 값은 암시적으로 False가 된다고 가정한다.
- '비어 있지 않은 값([1] 또는 'hi')에도 위와 같은 방식이 적용된다. 값이 비어 있지 않으면 if somelist 문이 암시적으로 True가 된다.
- 한 줄로 된 if 문, for와 while 루프, except 복합문을 쓰지 않는다. 이런 문장은 여러 줄로 나눠서 명료하게 작성한다.
- 항상 파일의 맨 위에 import 문을 놓는다.

- 모듈을 임포트할 때는 항상 모듈의 절대 이름을 사용하며 현재 모듈의 경로를 기준으로 상대 경로로 된 이름을 사용하지 않는다. 예를 들어 bar 패키지의 foo 모듈을 임포트하려면 그냥 `import foo`가 아닌 `from bar import foo`라고 해야 한다.
- 상대적인 임포트를 해야 한다면 명시적인 구문을 써서 `from . import foo`라고 한다.
- 임포트는 ‘표준 라이브러리 모듈, 서드파티 모듈, 자신이 만든 모듈’ 섹션 순으로 구분해야 한다. 각각의 하위 섹션에서는 알파벳 순서로 임포트한다.

Note

Pylint 도구(<http://www.pylint.org/>)는 파이썬 소스 코드를 정적 분석하는 도구로 유명하다. Pylint는 자동으로 PEP 8 스타일 가이드를 강요하고 파이썬 프로그램에서 일어나는 많은 유형의 일반적인 오류를 검출한다.

핵심 정리

- 파이썬 코드를 작성할 때 항상 PEP 8 스타일 가이드를 따르자.
- 큰 파이썬 커뮤니티에서 다른 사람과 원활하게 협업하려면 공통된 스타일을 공유해야 한다.
- 일관성 있는 스타일로 작성하면 나중에 자신의 코드를 더 쉽게 수정할 수 있다.

발췌: ‘파이썬 코딩의 기술