

Section 5.7: Engineering Practices and Python Conventions



(A few) Software Engineering Good Practices

- Use version control
- Write tests! Unit, Integration, Acceptance tests + more (see Section 9)
- Trunk-based development and peer reviews
- Understand your system dependencies
- Use CI/CD (Section 8 is dedicated to this)

Python Conventions 1: Style (PEP 8)

- The style guide for Python
- Makes working with your code much easier for other developers



Python Conventions 2: Type Hints (PEP 484)

```
1  import typing as t
2
3
4  def add_two_integers(first: int, second: t.Optional[int] = None) -> int:
5      """Sum two numbers."""
6
7      result = first
8      if second is not None:
9          result = first + second
10
11     return result
```

Python Conventions 3: Literal String Interpolation (PEP 498)



```
>>> import datetime
>>> name = 'Fred'
>>> age = 50
>>> anniversary = datetime.date(1991, 10, 12)
>>> f'My name is {name}, my age next year is {age+1}, my anniversary is {anniversary:%A, %B
%d, %Y}.'
```

'My name is Fred, my age next year is 51, my anniversary is Saturday, October 12, 1991.'

```
>>> f'He said his name is {name!r}.'
```

"He said his name is 'Fred'."

Python Conventions 4: Forcing Key Word Arguments (kwargs)

```
def my_function(*, foo):  
    pass
```

- When we call this function, we have to name foo:

```
my_function(foo="bar")
```



See you in the next section!

