1.

```python

class Thing:

pass

print(Thing())

example = Thing()

print(example)

```

Output:

```

<\_\_main\_\_.Thing object at 0x7f83c6cb09d0>

<\_\_main\_\_.Thing object at 0x7f83c6cb0a00>

```

The printed values are different because they represent different objects, but they have the same type which is the class `Thing`.

2.

```python

class Thing2:

letters = 'abc'

print(Thing2.letters)

```

Output:

```

abc

```

3.

```python

class Thing3:

def \_\_init\_\_(self):

self.letters = 'xyz'

example = Thing3()

print(example.letters)

```

Output:

```

xyz

```

Yes, we need to create an object from the class `Thing3` to access the instance attribute `letters`.

4.

```python

class Element:

def \_\_init\_\_(self, name, symbol, number):

self.name = name

self.symbol = symbol

self.number = number

hydrogen = Element('Hydrogen', 'H', 1)

```

5.

```python

element\_dict = {'name': 'Hydrogen', 'symbol': 'H', 'number': 1}

hydrogen = Element(\*\*element\_dict)

```

6.

```python

class Element:

def \_\_init\_\_(self, name, symbol, number):

self.name = name

self.symbol = symbol

self.number = number

def dump(self):

print(f'name={self.name}, symbol={self.symbol}, number={self.number}')

hydrogen = Element('Hydrogen', 'H', 1)

hydrogen.dump()

```

Output:

```

name=Hydrogen, symbol=H, number=1

```

7.

```python

class Element:

def \_\_init\_\_(self, name, symbol, number):

self.name = name

self.symbol = symbol

self.number = number

def \_\_str\_\_(self):

return f'name={self.name}, symbol={self.symbol}, number={self.number}'

hydrogen = Element('Hydrogen', 'H', 1)

print(hydrogen)

```

Output:

```

name=Hydrogen, symbol=H, number=1

```

8.

```python

class Element:

def \_\_init\_\_(self, name, symbol, number):

self.\_\_name = name

self.\_\_symbol = symbol

self.\_\_number = number

@property

def name(self):

return self.\_\_name

@property

def symbol(self):

return self.\_\_symbol

@property

def number(self):

return self.\_\_number

hydrogen = Element('Hydrogen', 'H', 1)

print(hydrogen.name)

print(hydrogen.symbol)

print(hydrogen.number)

```

Output:

```

Hydrogen

H

1

```

9.

```python

class Bear:

def eats(self):

return 'berries'

class Rabbit:

def eats(self):

return 'clover'

class Octothorpe:

def eats(self):

return 'campers'

bear = Bear()

rabbit = Rabbit()

octothorpe = Octothorpe()

print(bear.eats())

print(rabbit.eats())

print(octothorpe.eats())

```

Output:

```

berries

clover

campers

```

10.

```python

class Laser:

def does(self):

return 'disintegrate'

class Claw:

def does(self):