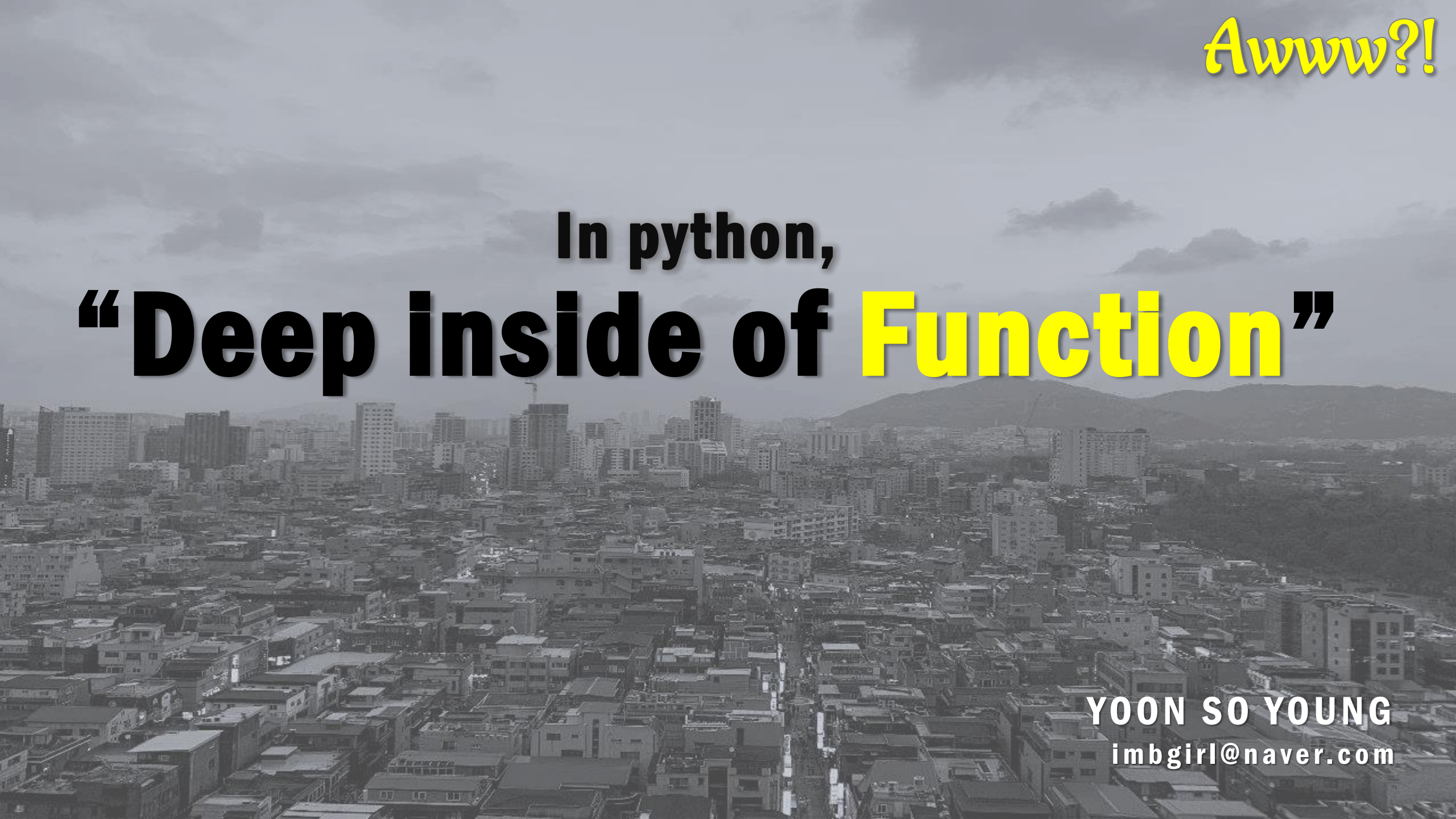

Awww?!
In python,
“Deep inside of Function”
YOON SO YOUNG
imbgirl@naver.com

‘Inside of Function 1/4’

Awww?!

Python → Everything is an object! — own information!

<class ‘function’> → a function object — own information!

- Let’s see the Function definition in Python

```
def function_name(parameter_list) :
    statement_list => ‘code’
```

function_name(argument_list)

- What kind of information a function should have?
  - ✓ __name__, parameters, __code__ (bytecode + local namespace)
  - ✓ __globals__ (global namespace)
  - ✓ __closure__ (to use outer function local namespace)

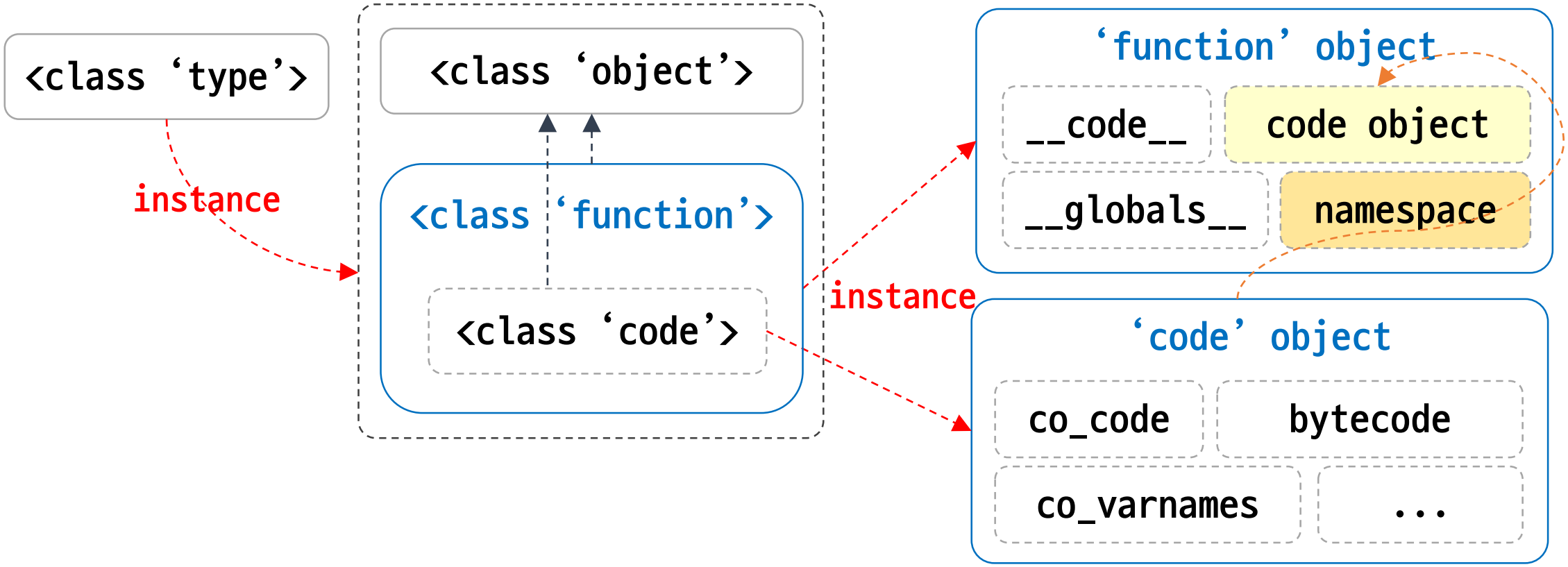
Awww!! All Right Reserved.

‘Inside of Function 2/4’

Awww?!

bytecode & local namespace ➡ ‘code object’

the reference of a ‘code object’ ➡ ‘function object’



Awww!! All Right Reserved.

‘Inside of Function 3/4’

Awww?!

- Attributes list of a function object

1. __doc__
2. __name__
3. __annotations__
4. __defaults__
5. __kwdefaults__
6. __globals__
7. __code__

```
name='soyoung'
def F1(a, b ='s', *, c, d=2) -> list:
    x = 10 + 20 + int('30')
    return [a, b, c, d]
```

```
print(F1.__name__) # print
```

https://docs.python.org/ko/3/library/inspect.html#types-and-members

Awww!! All Right Reserved.

‘Inside of Functions 4/4’

Awww?!

- Attributes list of code

1. co_code
2. co_name
3. co_nlocals
4. co_varnames
5. co_consts
6. co_kwonlyargcount

```
name='soyoung'
def F1(a, b ='s', *, c, d=2) -> list:
    x = 10 + 20 + int('30')
    return [a, b, c, d]
```

```
print(F1.__code__.co_name) # print
```

https://docs.python.org/ko/3/library/inspect.html#types-and-members

Awww!! All Right Reserved.