

```
import pprint as pp
import showattr as sa

value = 3
def F1() :
    value = 10
    def F1_inner() :
        value = 20
        return locals()
    sa.show_function_local('F1_inner function', F1_inner, F1_inner())
    return locals()

sa.show_module('current module', globals())
sa.show_function_local('F1 function', F1, F1())
```

current module

```
__name__      - __main__
__builtins__  - <class 'dict'>
pp            - <class 'module'>
sa            - <class 'module'>
value         - 3
F1            - <class 'function'>
```

F1_inner function

```
__name__      - F1_inner
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'NoneType'>
value         - 20
```

F1 function

```
__name__      - F1
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'NoneType'>
value         - 10
F1_inner      - <class 'function'>
```

Three modules, two functions

- current module
- pp module
- sa module
- F1_inner function
- F1 function

```
import showattr as sa

value = 3
def F1() :
    def F1_inner() :
        print('value at F1_inner = ', value)
        return locals()
    sa.show_function_local('F1_inner function', F1_inner, F1_inner())
    return locals()

sa.show_module('current module', globals())
sa.show_function_local('F1 function', F1, F1())
```

current module

```
__name__      - __main__
__builtins__  - <class 'dict'>
sa            - <class 'module'>
value        - 3
F1           - <class 'function'>
```

Two modules, two functions

- current module
- sa module
- F1_inner function
- F1 function

value at F1_inner = 3

F1_inner function

```
__name__      - F1_inner
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'NoneType'>
```

F1 function

```
__name__      - F1
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'NoneType'>
F1_inner      - <class 'function'>
```

```
import showattr as sa

value = 3
def F1() :
    value = 10
    def F1_inner() :
        print('value at F1_inner = ', value)
        return locals()
    sa.show_function_local('F1_inner function', F1_inner, F1_inner())
    return locals()

sa.show_module('current module', globals())
sa.show_function_local('F1 function', F1, F1())
```

current module

```
__name__      - __main__
__builtins__  - <class 'dict'>
sa            - <class 'module'>
value        - 3
F1           - <class 'function'>
```

value at F1_inner = 10

F1_inner function

```
__name__      - F1_inner
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'tuple'>
value        - 10
=> co_freevars : value - 10
```

F1_inner의 locals()에서 나온 것

F1 function

```
__name__      - F1
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'NoneType'>
F1_inner      - <class 'function'>
value        - 10
```

Two modules, two functions

- current module
- sa module
- F1_inner function
- F1 function

```
import showattr as sa

def add_ret(a, b):
    c = a + b
    return c, locals()

r = add_ret(10, 20)
sa.show_module('module', globals())
sa.show_function_local('add_ret function', add_ret, add_ret(1, 2)[1])
```

```
module
__name__      - __main__
__builtins__  - <class 'dict'>
sa            - <class 'module'>
add_ret       - <class 'function'>
r            - <class 'tuple'>
```

Two modules, one function

- current module
- sa module
- add_ret function

```
add_ret function
__name__      - add_ret
__globals__   - <class 'dict'>
__code__      - <class 'code'>
__closure__   - <class 'NoneType'>
a            - 1
b            - 2
c            - 3
```

```
value = 3
def add_num1(num=1):
    value = value + num

def add_num2(num=1):
    print(value)
    value = value + num
    print(value)

print(value)
add_num1()
add_num2()
print(value)
```

UnboundLocalError:
local variable 'value' referenced before assignment

showattr.py 의 함수 안내

1. show_function(title, func)

- 함수 객체의 속성 정보를 보여주는 함수, title 은 제목, func 은 함수 객체 전달용
- 함수 객체의 속성 정보 중 아래의 5 가지를 보여줌
__class__, __name__, __annotations__, __defaults__, __kwdefaults__

2. show_function_local(title, func, local)

- 함수 객체의 속성 정보 + locals()를 보여주는 함수, title 은 제목, func 은 함수 객체 전달, local 은 locals() 전달용
- 함수 객체의 속성 정보 중 아래의 5 가지를 보여줌
__name__, __globals__, __code__, __closure__
- __closure__ 가 None 이 아닌 경우 co_freevars 와 __closure__의 cell_contents 를 보여줌

3. show_code(title, code)

- 코드 객체의 속성 정보를 보여주는 함수, title 은 제목, code 는 코드 객체 전달
- 코드 객체의 속성 정보 중 아래의 8 가지를 보여줌
__class__, co_name, co_nlocals, co_varnames, co_consts, co_kwonlyargcount, co_freevars, co_cellvars

4. show_module(title, mobj)

- 모듈 객체의 속성 정보를 보여주는 함수, title 은 제목, mobj 는 모듈 객체 전달
- 모듈 객체의 속성 정보 중 아래의 2 가지를 보여줌
__name__, __builtins__