

STEP 1. 상자글서의 변수

```
class Variable:
```

```
    def __init__(self, data):
```

```
        self.data = data
```

← 주어진 인수를 인스턴스 변수 data에 대입.

```
import numpy as np
```

```
data = np.array(1.0)
```

```
X = Variable(data)
```

```
print(X.data)
```

⇒ 1.0

Variable 인스턴스

실제 데이터는 X에 담겨있음.

STEP 2. 변수를 넣는 함수

- Variable 인스턴스를 변수를 다룰 수 있는 함수 구현

Class Function:

```
def *!__call__(self, input):  
    x = input.data      # 데이터를 꺼냄.  
    y = x ** 2          # 실제 계산  
    output = Variable(y)  # Variable 형태로 되돌림.  
    return output
```

```
x = Variable(np.array(10))
```

```
f = Function()
```

```
y = f(x)
```

```
print(type(y))    # type() 함수: 객체의 클래스를 알려줌.
```

```
print(y.data)
```

↙
y의 클래스는 Variable 이며,
데이터는 y.data 에 저장되어 있음.

→ Function 클래스 수정

RULE

- Function 클래스는 기반클래스, 모든 함수에 공통되는 기능 구현
- 구체적인 함수는 Function 클래스를 상속한 클래스에서 구현

Class Function:

```
def __call__(self, input):
```

```
    x = input.data
```

```
    y = self.forward(x)    # 구체적인 계산을 forward 메서드에서
```

```
    output = Variable(y)    # Variable 객체를 되돌림
```

```
    return output
```

```
def forward(self, x):
```

```
    raise NotImplementedError()    → forward 메서드의 구체적인 동작은  
                                   하위 클래스에서 구현
```

↓ Function 클래스를 상속하여
임의값을 제공하는 클래스 구현

```
Class Square (Function):
```

```
    def forward(self, x):
```

```
        return x ** 2
```

```
x = Variable (np.array (10))
```

```
f = Square ()
```

```
y = f(x)
```

```
print (type(y))
```

```
print (y.data)
```

STEP 3. 함수 연결

```
class Exp (Function):  
    def forward (self, x):  
        return np.exp(x)
```

Function 클래스의 `-- call --` 메서드는 입력과 출력이 모두 Variable 인스턴스이면 DeZero 함수들을 연이어 사용 가능.

Example :

$$y = (e^{x^2})^2$$



```
A = Square()
```

```
B = Exp()
```

```
C = Square()
```

```
x = Variable (np.array (0.5))
```

```
a = A(x)
```

```
b = B(a)
```

```
y = C(b)
```

```
print (y.data)
```

STEP 4. 수치미분

$f(x)$ 라는 함수에 대한 미분

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

극한 \rightarrow $y=f(x)$ 그래프의 두 점을 지나는 직선의 기울기

수치미분이란?

- 미세한 차이 ($0.0001 = 1e-4$)를 이용해 함수의 변화량을 구하는 방법.
- 작은값을 이용하여 '잔여한 미분'을 근사
- 근사오차를 줄이는 방법? '중앙차분 (centered difference)'

• 직선의 기울기: $\frac{f(x+h) - f(x-h)}{2h}$

numerical_diff (f, x, eps = $1e-4$)

- f: 미분의 대상이 되는 함수 (Function 의 인스턴스)
- x: 미분을 계산하는 변수 (Variable 인스턴스)
- eps (= epsilon) : 작은값

```
def numerical_diff (f, x, eps = 1e-4):
```

```
    x0 = Variable (x.data - eps)
```

```
    x1 = Variable (x.data + eps)
```

```
    y0 = f(x0)
```

```
    y1 = f(x1)
```

```
    return (y1.data - y0.data) / (2 * eps)
```

\downarrow Square 클래스를 대상으로 미분

```
f = Square()
```

```
x = Variable (np.array (2.0))
```

```
dy = numerical_diff (f, x)
```

```
print (dy)
```

* 1 : `--call--`

- 파이썬 특수 메소드
- `f = Function()` 형태로 함수의 인스턴스를 변수 `f`에 대입해두고,
나중에 `f(...)` 형태로 `--call--` 메소드 호출 가능

