Quiz: CBV or CBN?

In this lesson, we will take quick quiz to better understand CBV and CBN.

While Scala’s default evaluation strategy is *call-by-value*, *call-by-name* can be enforced using `=>`. Hence, it is important to know when to use CBV and when to use CBN. Take a look at the function below.

```
def evaluate(x: Int, y: Int) ={
  x * x
}
```

This function is just like our square function with an added extra parameter `y`. Now, `y` isn’t required for the evaluation of the function’s body. Can you figure out which evaluation strategy is faster in the following function calls to the evaluate function?

> By *faster*, we simply mean taking a lesser number of steps to reach the final result.

1 evaluate(2,3)

2

evaluate(3+4, 8)

3

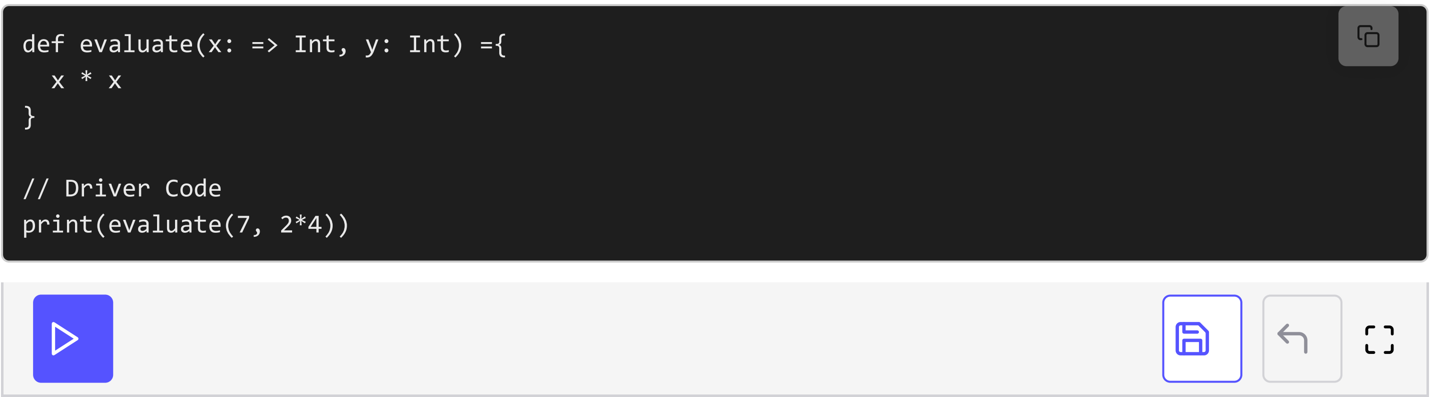
evaluate(7, 2*4)

Retake Quiz

If we wanted the `evaluate` function to evaluate its first parameter using CBN, the function would be written as below.

This code requires the following environment variables to execute:

LANG	C.UTF-8



```
def evaluate(x: => Int, y: Int) ={
  x * x
}

// Driver Code
print(evaluate(7, 2*4))
```

In the next lesson, we will go over which evaluation strategy should be used depending on the expression to be evaluated.