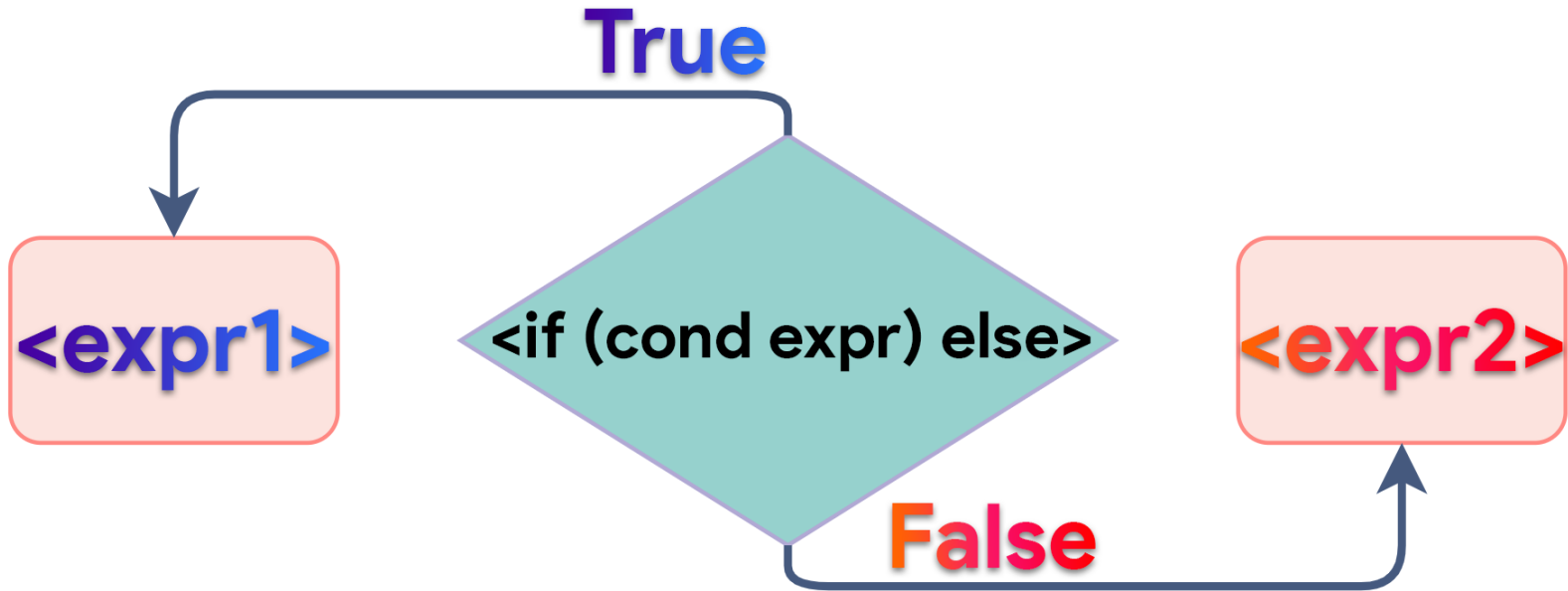


# Python Bootcamp & Masterclass

## ternary operator



As its name implies, ternary operator (also called a conditional operator or a conditional expression) has three operands. Its syntax is of the form : <expr1> if <cond\_expr> else <expr2> First, the middle operand ( if <cond\_expr> else ) is evaluated. If it is True, <expr1> is evaluated and if it is False. <expr2> is evaluated.



```
score = 98
print("Congrats! You've won a scholarship") if score >= 90 else print("Sorry, You did not win a scholarship")
```

Congrats! You've won a scholarship

```
score = 84
print("Congrats! You've won a scholarship") if score >= 90 else print("Sorry, You did not win a scholarship")
```

Sorry, You did not win a scholarship

```
x, y = 40, 80
11 + x if x > y else y + 12
```

92

```
x, y = 40, 40
2/0 if x > y else y + 12 # divide by zero if ignored because of short-circuiting
```

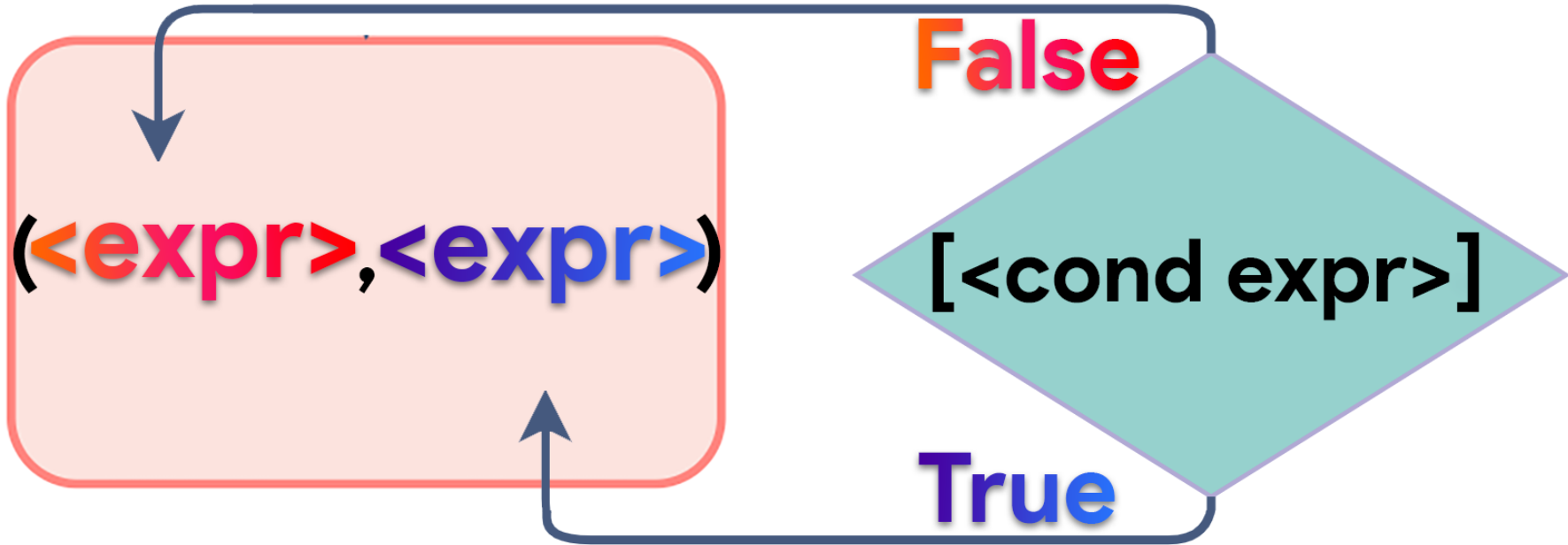
52

```
y = 20
print("no") if y > 42 else print("yes") if y == 42 else print("maybe") # nested ternary operator
```

maybe

A tuple can also be used in ternary operations.

Its syntax is of the form : (<expr1>, <expr2>)[ <cond\_expr> ] This is same as indexing as True has a value of 1 and False has a value of 0



```
a = 8
b = 5
larger = (b, a)[a > b]
smaller = (a, b)[a > b]
print("larger of", a, "and", b, "is:", larger)
print("smaller of", a, "and", b, "is:", smaller)
```

```
larger of 8 and 5 is: 8
smaller of 8 and 5 is: 5
```

```
overcast = 40
{True: 'Watch Golf', False: 'Play Golf'}[overcast > 50]
{False: 'Play Golf', True: 'Watch Golf'}[overcast > 50]
```

```
'Play Golf'
```

```
'Play Golf'
```



# Online Resources

**For best python resources, please visit:**



[gknxt.com/python/](https://gknxt.com/python/)

# Python Bootcamp & Masterclass

**Thank You**  
for your Rating & Review

