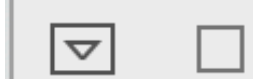


Functions guided tour

```
5 func greet( [REDACTED], day: String) -> String {  
6     return "Hello \ (person), today is \ (day)."  
7 }  
8 print(greet(person: "Bob", day: "Tuesday"))  
9
```

⌵ □
Hello Bob, [REDACTED].

```
15 func greet(_ person: String, on day: String) ->  
    String {  
16     return "Hello \ (person), today is \ (■■■■) ."  
17 }  
18 greet( "John", ■■■■ : "Wednesday" )  
19
```



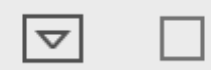
Hello Bob, today is Tuesday.

```
1 var y = 10
2 func addFive(){
3     y += █
4 }
5 █()
6 print(y)
7
```

10

15

"15\n"

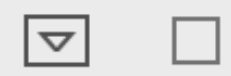


15

```
9 func returnFifteen() -> Int {
10     var y = 10
11     func [REDACTED]() {
12         y += 5
13     }
14     addFive()
15     [REDACTED] y
16 }
17 returnFifteen()
```

```
1 func addOne(number: Int) {  
2     return 1 + number  
3 }
```

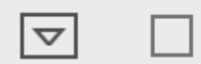
```
4 print(addOne(4))
```



5

The output argument of `makeIncrementer` is a function.

```
1 func makeIncrementer() -> ((Int)         ) {  
2     func addOne(number: Int) -> Int {  
3         return 1 + number  
4     }  
5     return addOne  
6 }  
7 var increment = makeIncrementer()  
8 print(increment(7))
```



A function can take another function as one of its arguments.

```
67 func hasAnyMatches(list: [Int], condition: (Int) ->
    Bool) -> Bool {
68     for item in list {
69         if condition(item) {
70             return true
71         }
72     }
73     return false
74 }
75 func lessThanTen(number: Int) -> Bool {
76     return number < 10
77 }
78 var numbers = [20, 19, 7, 12]
79 hasAnyMatches(list: numbers, condition: lessThanTen)
```

A function can take another function as one of its arguments.

```
67 func hasAnyMatches(list: [Int], condition: (Int) ->
    Bool) -> Bool {
68     for item in list {
69         if condition(item) {
70             return true
71         }
72     }
73     return false
74 }
75 func lessThanTen(number: Int) -> Bool {
76     return number < 10
77 }
78 var numbers = [20, 19, 7, 12]
79 hasAnyMatches(list: numbers, condition: lessThanTen)
```

```
54 func calculateStatistics(scores: [Int]) -> (min:
    Int, max: Int, sum: Int) {
55     var min = scores[0]
56     var max = scores[0]
57     var sum = 
58
59     for scores {
60         if score > max {
61             max = score
62         } else if score < min {
63             min = score
64         }
65         sum += score
66     }
67
68     return ( , sum)
69 }
```

```
54 func calculateStatistics(scores: [Int]) -> (min:
    Int, max: Int, sum: Int) {
55     var min = scores[0]
56     var max = scores[0]
57     var sum = 0
58
59     for score in scores {
60         if score > max {
61             max = score
62         } else if score < min {
63             min = score
64         }
65         sum += score
66     }
67
68     return (min, max, sum)
69 }
```