0

```
(Task 1 of 8) In this tutorial, we will learn about using functions as values.
```

Note: some programming languages do NOT consider functions or methods a kind of value. But many programming languages, from Python to Rust, do.

```
Lispy | 🤴
(Task 2 of 8) What is the result of running this program?
                 Lispy [Run 🔼]
                                                           JavaScript
 (deffun (twice f x)
                            function twice(f, x) {
   (f(fx))
                              return f(f(x));
 (deffun (double x)
   (+ \times \times)
                            function double(x) {
 (twice double 1)
                              return x + x;
                            console.log(twice(double, 1));
                                                                                           2
4
                                                                                           3
Please briefly explain why you think the answer is 4.
                                                                                           4
We call the function passed into twice twice
You predicted the output correctly ***
The value of (twice double 1) is the value of (f(fx)), where f is bound to double
and x is bound to 1. So, the result is the value of (double (double 1)), which is 4.
Click <u>here</u> to run this program in the Stacker.
```

```
Lispy | 🤴
(Task 3 of 8) What is the result of running this program?
           Lispy [Run ]
 (deffun (inc x)
                      definc(x : Int) =
   (+ \times 1)
                        x + 1
 (deffun (g)
                      def g =
   inc)
                        inc
 (defvar f (g))
                      val f = g
                      println(f(10))
 (f 10)
                                                                                         7
error
                                                                                         8
The answer is 11
```

1 of 4 1/24/25, 20:33

THE ALISVVEL IS II.

▼ Textual explanation

You might think that (g) errors because it returns a function, and that it would not error if it returns a value of other kinds (e.g., numbers and vectors). However, it does not error. In SMoL, functions are (also) *first-class* citizens of the value world.

Click <u>here</u> to run this program in the Stacker.

```
Lispy | 🧣
What is the result of running this program?
                    Lispy [Run 🔄
                                                       JavaScript
 (deffun (fun1)
                              function fun1() {
   (deffun (average x y)
                                function average(x, y) {
     (/ (+ x y) 2))
                                   return (x + y) / 2;
   average)
 (defvar x (fun1))
                                return average;
 (x 20 40)
                              let x = fun1();
                              console.log(x(20, 40));
                                                                                    10
30
                                                                                    11
You predicted the output correctly **
```

You predicted the output correctly

14

v is bound to a vector that refers to the function inc. The value of (vec-ref v θ) is the function inc. So, the value of ((vec-ref v θ) 2) is the value of (inc 2), which is 3.

Click here to run this program in the Stacker.

Lispy | 🚱

(Task 5 of 8) What is the result of running this program?

```
Lispy [Run 🔼
                                Python
 (deffun (f) def f():
                        return 42
   42)
                   g = f
 (defvar g f)
 (defvar h g) h = g
 (h)
                   print(h())
                                                                                             16
42
                                                                                             17
You predicted the output correctly
This program binds f to a function that returns 42, and then binds g and h to that function.
Finally, calling that function produces 42.
Click <u>here</u> to run this program in the Stacker.
                                                                                             18
(Task 6 of 8) What did you learn about functions from these programs?
                                                                                             19
Functions exist in the same way variables do. They can be passed around and all normal
variable rules apply to functions
                                                                                            20
 (Task 7 of 8) Functions are (also) first-class citizens of the value world. Specifically,

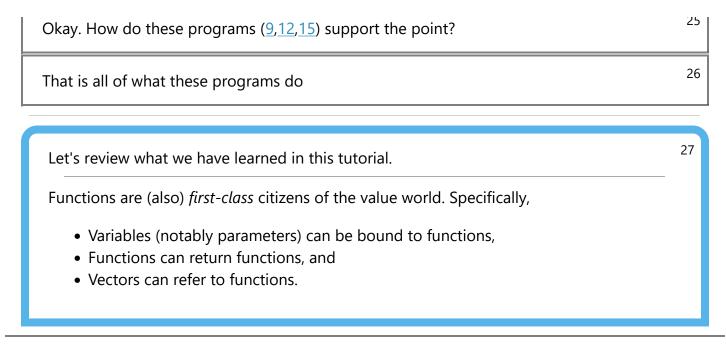
    Variables (notably parameters) can be bound to functions,

    · Functions can return functions, and

    Vectors can refer to functions.

                                                                                             21
Any feedback regarding these statements? Feel free to skip this question.
                                                                                             22
(You skipped the question.)
                                                                                             23
(Task 8 of 8) Please scroll back and select 1-3 programs that make the point above.
You don't need to select all such programs.
                                                                                             24
(You selected 3 programs)
```

3 of 4 1/24/25, 20:33



You have finished this tutorial

Please print the finished tutorial to a PDF file so you can review the content in the future. **Your** instructor (if any) might require you to submit the PDF.

Start time: 1737764497818

4 of 4 1/24/25, 20:33