In this tutorial, we will learn about `begin`.

`begin` lets you evaluate a sequence of expressions and returns the value of the last expression.

What is the result of running this program?

```
(defvar x (begin 2 3 4))
x
```

COPY as program

4

You got it right! 🎉🎉🎉 The definition binds `x` to the value of the `begin` expression. A `begin` expression evaluates its sub-expressions from left to right and produces the value of the right-most expression. So `x` is bound to `4`.

What is the result of running this program?

```
(defvar z
  (begin
    (begin
      123
      45)
    (begin
      67)))
z
```

COPY as program

67

You got it right! 🎉🎉🎉 The value of `(begin (begin 123 45) (begin 67))` is the value of `(begin 67)`, which is `4`.

What is the result of running this program?

```
(defvar x 0)
(defvar y
  (begin
    (set! x (+ x 1))
    x))
x
```

COPY as program

Please wait a couple of seconds to let us record your response. There will be a pop-up window when we are done.