Syntax: Lispy JS PY Scala 3

In this tutorial, we will learn about **mutable values**, illustrated with **lists**.

The following program illustrates how to create lists

```
print([[1, 2], [3], []])
```

Run ▶

This program produces `[[1, 2], [3], []]`. It creates four lists:

- a two-**element** list that refers `1` and `2`
- a one-element list that refers `3`
- an empty list
- a three-element list that refers all three aforementioned lists

---

Syntax: Lispy JS PY Scala 3

What is the result of running this program?

```
n = 44
def f(x):
    return x + 1
print([n, f(n)])
```

Run ▶

Syntax: Lispy JS PY Scala 3

`[44, 45]`

Syntax: Lispy JS PY Scala 3

You got it right! 🎉🎉🎉

This program binds `n` to `44` and `f` to a function that adds 1 to its parameter. After that, the program creates a list that refers the value of `n`, which is `44`, and the value of `f(n)`, which is `45`. The list is printed as `[44, 45]`.

Click here to run this program in the Stacker.

Syntax: Lispy JS PY Scala 3

The following program illustrates how to refer to list elements.

```
v = [84, 75]
print(v[0])
print(v[1])
print(v[2])
```

Run ▶

This program produces `84 75 error`. It refers to the `0`-th (i.e., first) element, the `1`-th element, and then tries to refer to the `2`-th element.

Syntax: Lispy JS PY Scala 3

What is the result of running this program?

```
v = [[58, 43], [43, 66]]
```

Run ▶

```
v = [[58, 43], [43, 66]]
vr = v[1]
print(vr[0])
```

Run

Syntax: Lispy JS PY Scala 3

```
43
```

Syntax: Lispy JS PY Scala 3

You got it right!

Recall that the left-most element is the 0-th (rather than 1-th!) element. `v[1]` produces the value of `[43, 66]`. So, `vr[0]` produces `43`.

Click here to run this program in the Stacker.

Syntax: Lispy JS PY Scala 3

The following program illustrates how to mutate lists.

```
m = [62, 77]
m[0] = 83
print(m[0])
```

Run

This program produces `83`. It **mutates** the list by **replacing** the `0`-th element with `83` and then refers to the `0`-th element.

Syntax: Lispy JS PY Scala 3

What is the result of running this program?

```
x = [92, 73]
x[0] = 67
print(x)
```

Run

Syntax: Lispy JS PY Scala 3

```
[67, 73]
```

Syntax: Lispy JS PY Scala 3

You got it right!

`x` is bound to a list. The initial content of the list is `92` and `73`. The mutation replaces the first element with `67`. A list referring `67` and `73` is printed as `[67, 73]`.

Click here to run this program in the Stacker.

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: 1711097196536