

# Understanding the Poly Language

Poly is a simple, educational programming language designed to look similar to Python but with a cleaner, smaller syntax. It is interpreted by the PolyLangInterpreter class written in Python. Poly supports variables, conditionals (if-else), functions, recursion, and closures. It's designed to teach programming logic and structure by mimicking how real languages work.

## 1. Variables

In Poly, variables are declared using the keyword **let**. You can store numbers, text, or even function results.

Poly Example:

```
let x = 10
let name = "Sarah"
print(x)
print(name)
```

Python Equivalent:

```
x = 10
name = "Sarah"
print(x)
print(name)
```

## 2. If-Else Statements

Poly uses a block structure with **then** and **else**: to control flow. It looks simpler than Python's colon-based syntax.

Poly Example:

```
let x = 7
if x > 5 then
    print("x is greater than 5")
else:
    print("x is 5 or less")
```

Python Equivalent:

```
x = 7
if x > 5:
    print("x is greater than 5")
else:
    print("x is 5 or less")
```

## 3. Functions and Recursion

Functions in Poly are declared with the keyword **func**. They can call themselves (recursion) just like in Python.

Poly Example (Recursive Factorial):

```
func fact(n) = if n == 1 then 1 else n * fact(n - 1)
print(fact(5))
```

Python Equivalent:

```
def fact(n):
if n == 1:
return 1
else:
return n * fact(n - 1)
print(fact(5))
```

## 4. Closures (Functions inside Functions)

Poly supports closures — meaning functions can return another function that remembers the outer variable context. This allows for functional programming concepts.

Poly Example:

```
func make_adder(x) = func(y) = x + y
let add5 = make_adder(5)
print(add5(10))
```

Python Equivalent:

```
def make_adder(x):
def inner(y):
return x + y
return inner
add5 = make_adder(5)
print(add5(10))
```

## Summary

Poly is designed to simplify programming by using minimal syntax and readable commands. You can learn about logic, recursion, and closures just like in Python but with a more compact structure. It's excellent for learning programming fundamentals or building a custom scripting tool.