Julia For Beginners 🤫

#DataRockie #Bootcamp11

Curriculum

- Variable
- Type
- Structure
- Control Flow
- Function
- DataFrame

Intro to Julia

Why Learn Third Language

- **Free**
- **Fast**
- **Easy to Learn**
  - Slightly more difficult than R, Python
- **Community**
- **Active Development Since 2012**

The Basics

Variable

```
# create new variables
x = 100
y = 200

println(100 + 200)
```

```
# simple example
income = 50000
expense = 27000

saving = income - expense
println(saving)
```

```
# get input from a user
println(“what’s your name?”)

name = readline()

println(“hi $name”)
```

The Basics

Type

- Int
- Float
- String
- Bool

```
# check data type
x = 10
println(typeof(x))

y = "Hello"
println(typeof(z))
```

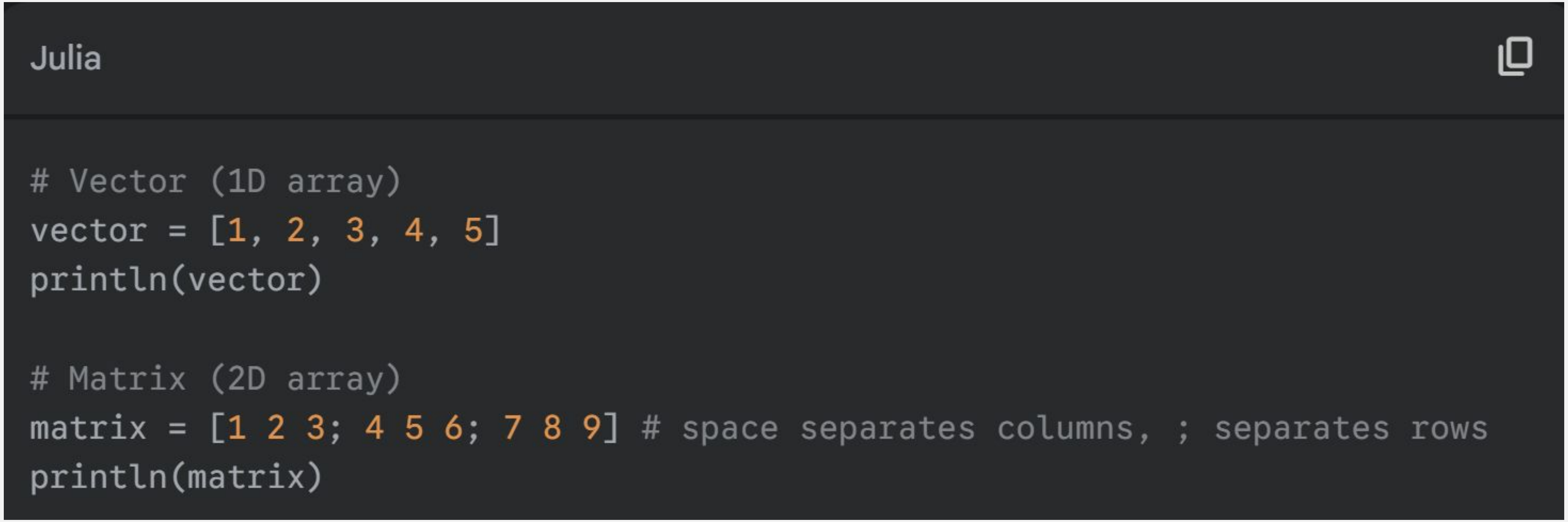
```julia
# use parse to check data type
println("Hello, Julia!")

user_input = readline()

number = parse(Int, user_input)

println("You entered: ", number)
```

The Basics

Structure

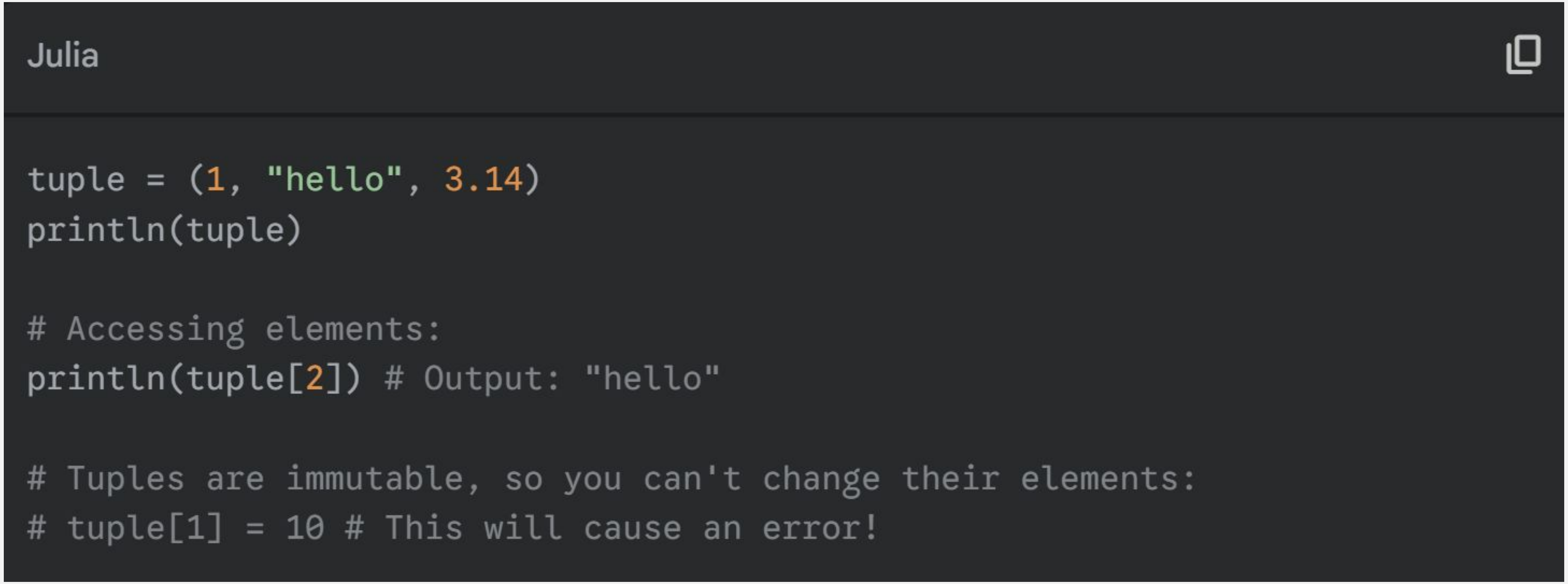
- Array
- Tuple
- Dictionary
- Set

Array



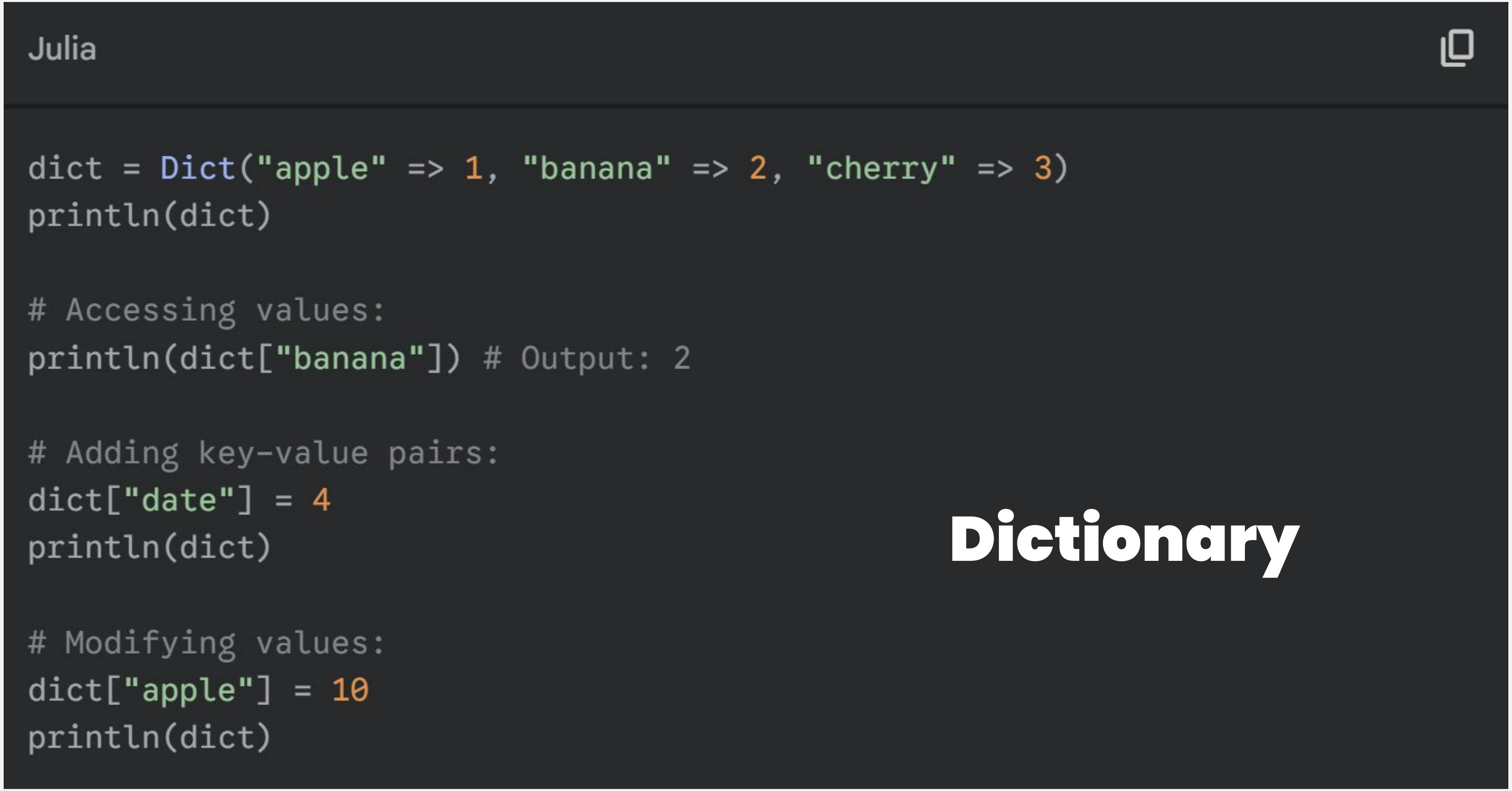
```julia
# Vector (1D array)
vector = [1, 2, 3, 4, 5]
println(vector)

# Matrix (2D array)
matrix = [1 2 3; 4 5 6; 7 8 9] # space separates columns, ; separates rows
println(matrix)
```

Tuple



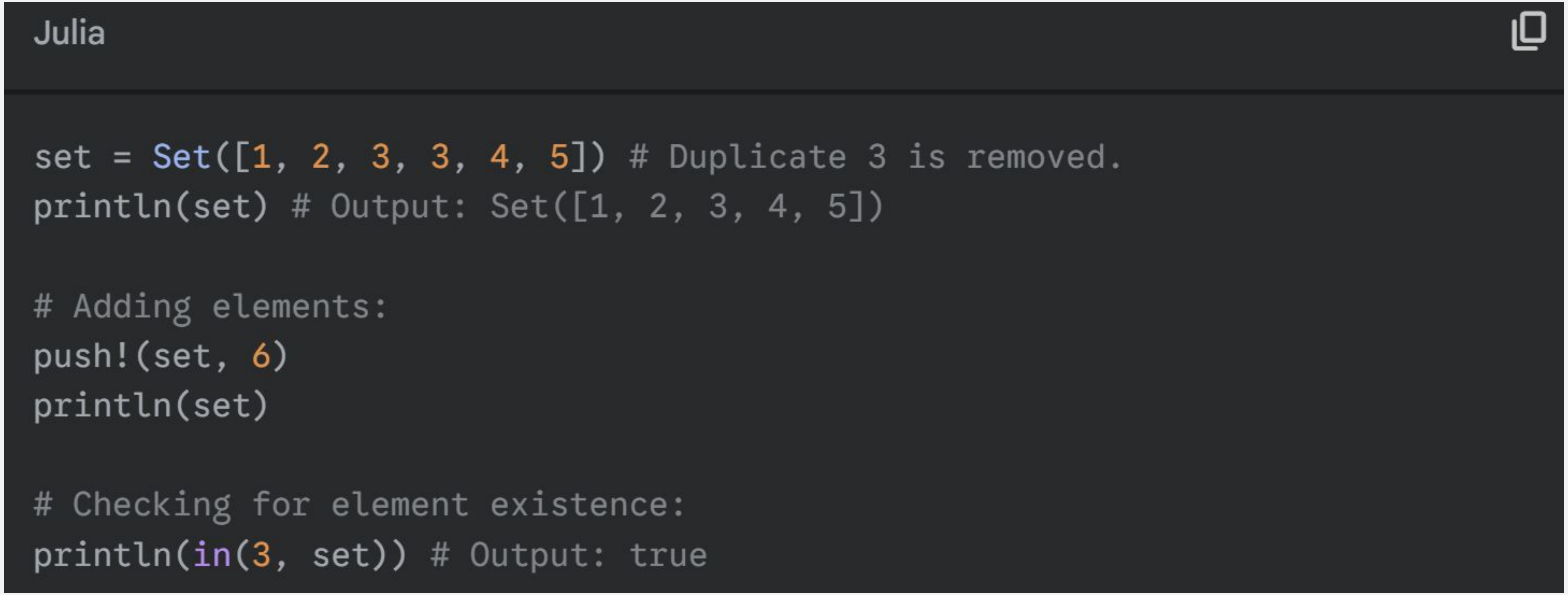
```julia
tuple = (1, "hello", 3.14)
println(tuple)

# Accessing elements:
println(tuple[2]) # Output: "hello"

# Tuples are immutable, so you can't change their elements:
# tuple[1] = 10 # This will cause an error!
```

Dictionary



```julia
dict = Dict("apple" => 1, "banana" => 2, "cherry" => 3)
println(dict)

# Accessing values:
println(dict["banana"]) # Output: 2

# Adding key-value pairs:
dict["date"] = 4
println(dict)

# Modifying values:
dict["apple"] = 10
println(dict)
```

Set

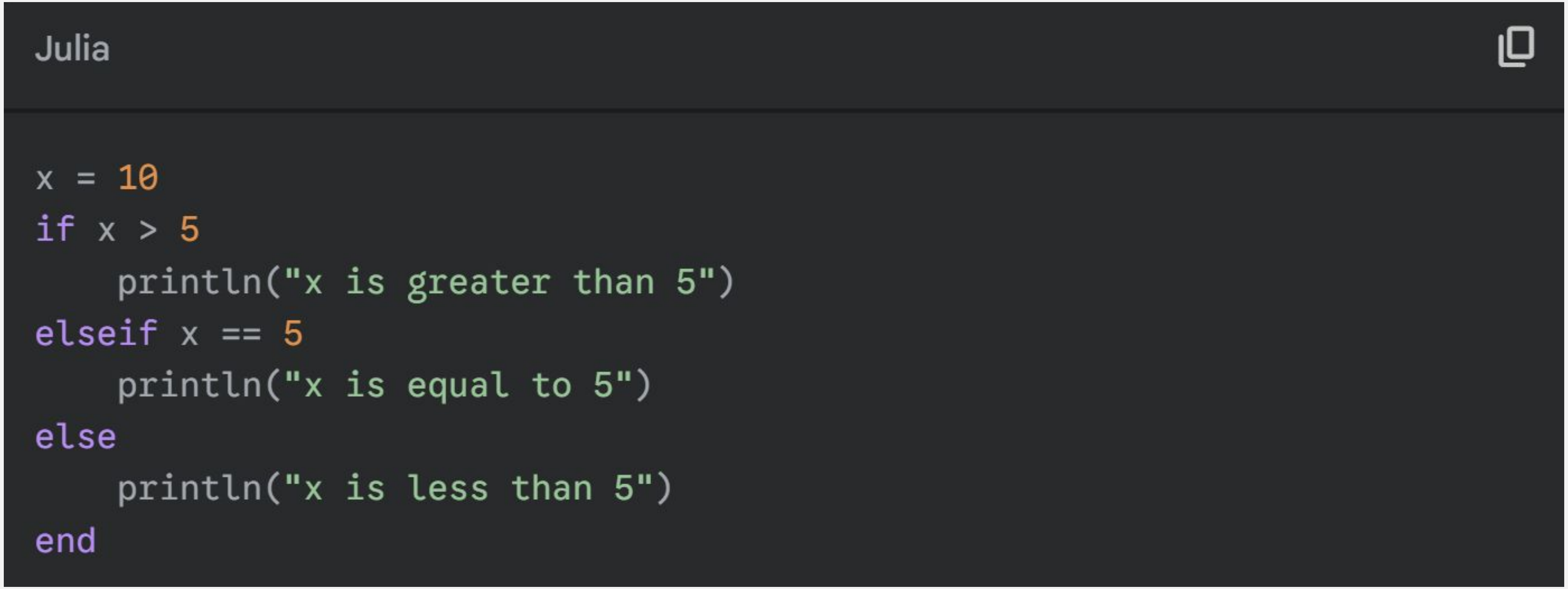

Julia

```julia
set = Set([1, 2, 3, 3, 4, 5]) # Duplicate 3 is removed.
println(set) # Output: Set([1, 2, 3, 4, 5])

# Adding elements:
push!(set, 6)
println(set)

# Checking for element existence:
println(in(3, set)) # Output: true
```

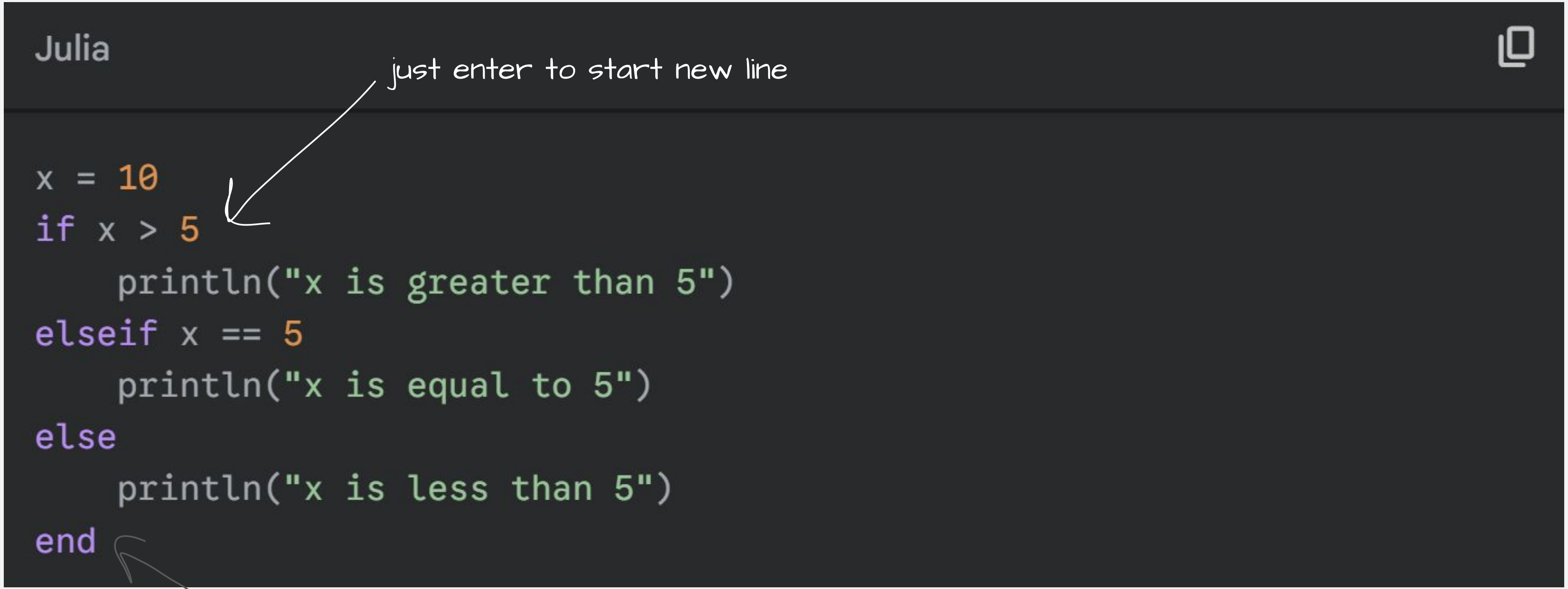
The Basics

Control Flow

- if elseif else
- for
- while


Julia

```julia
x = 10
if x > 5
    println("x is greater than 5")
elseif x == 5
    println("x is equal to 5")
else
    println("x is less than 5")
end
```



Julia

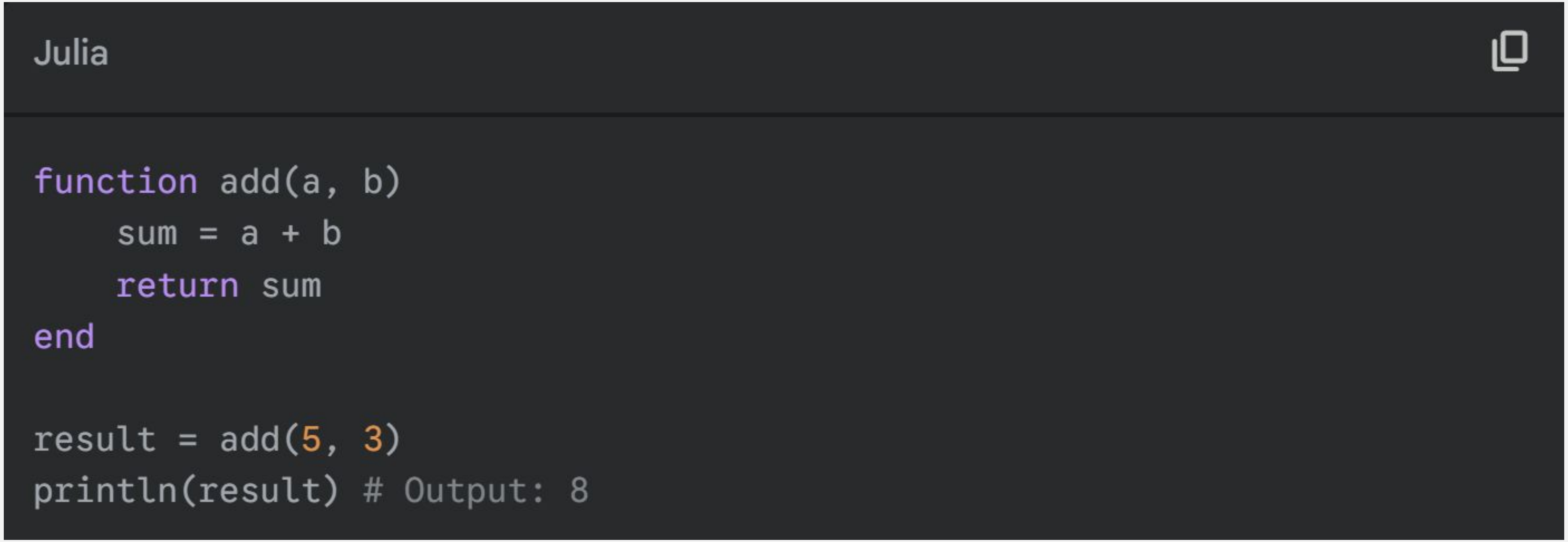
just enter to start new line

```julia
x = 10
if x > 5
    println("x is greater than 5")
elseif x == 5
    println("x is equal to 5")
else
    println("x is less than 5")
end
```

use **end** keyword to finish
your code block

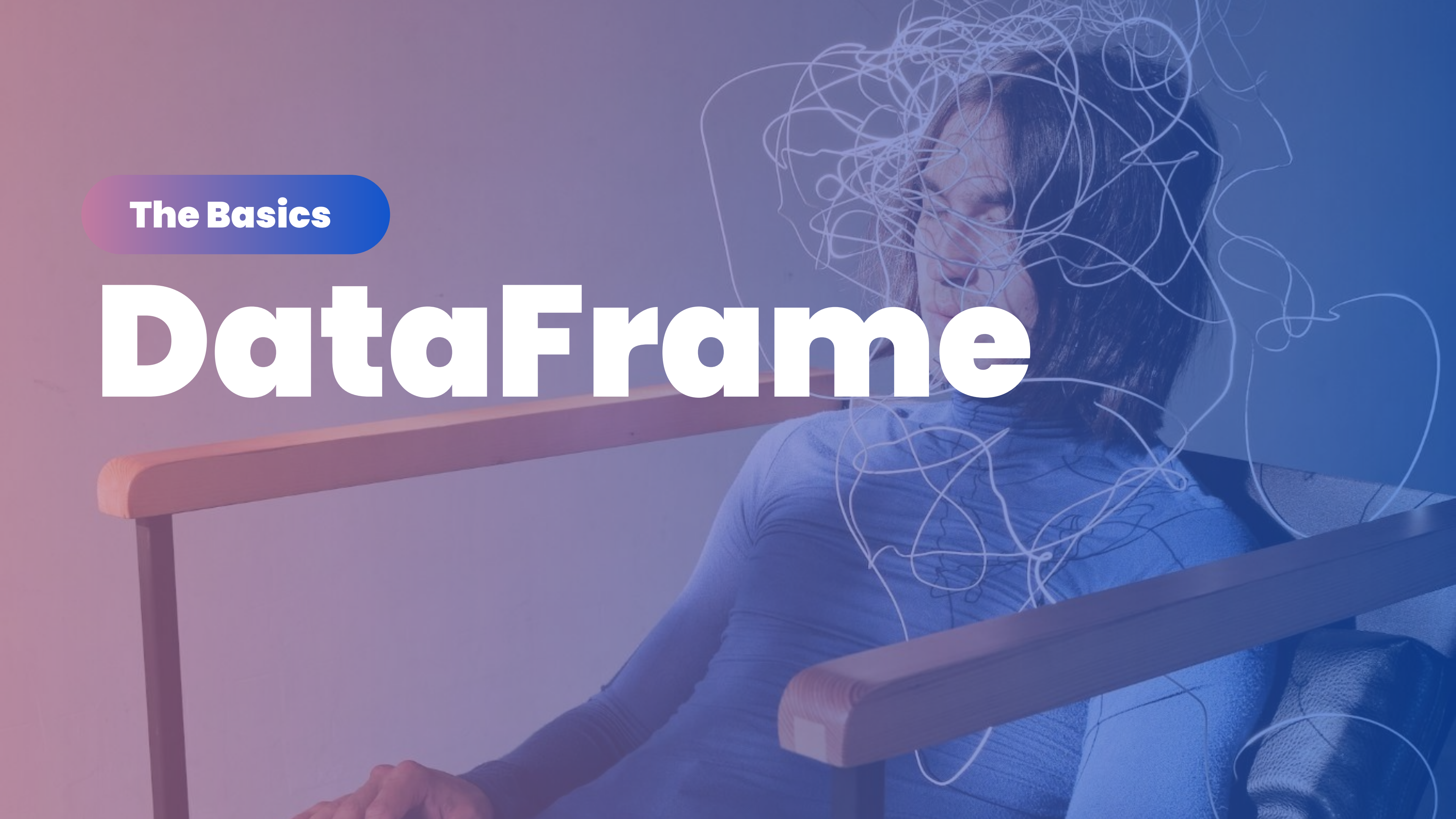
The Basics

Function



```julia
function add(a, b)
    sum = a + b
    return sum
end

result = add(5, 3)
println(result) # Output: 8
```

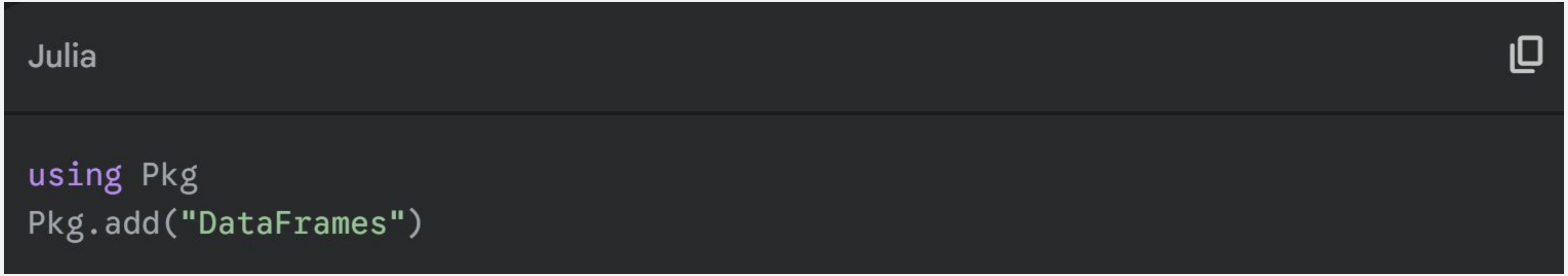
```julia
# create a new function
function add(a, b)
    return a + b
end

# create a new function one-line
multiply = (x, y) -> x * y
multiply(3, 5)
```



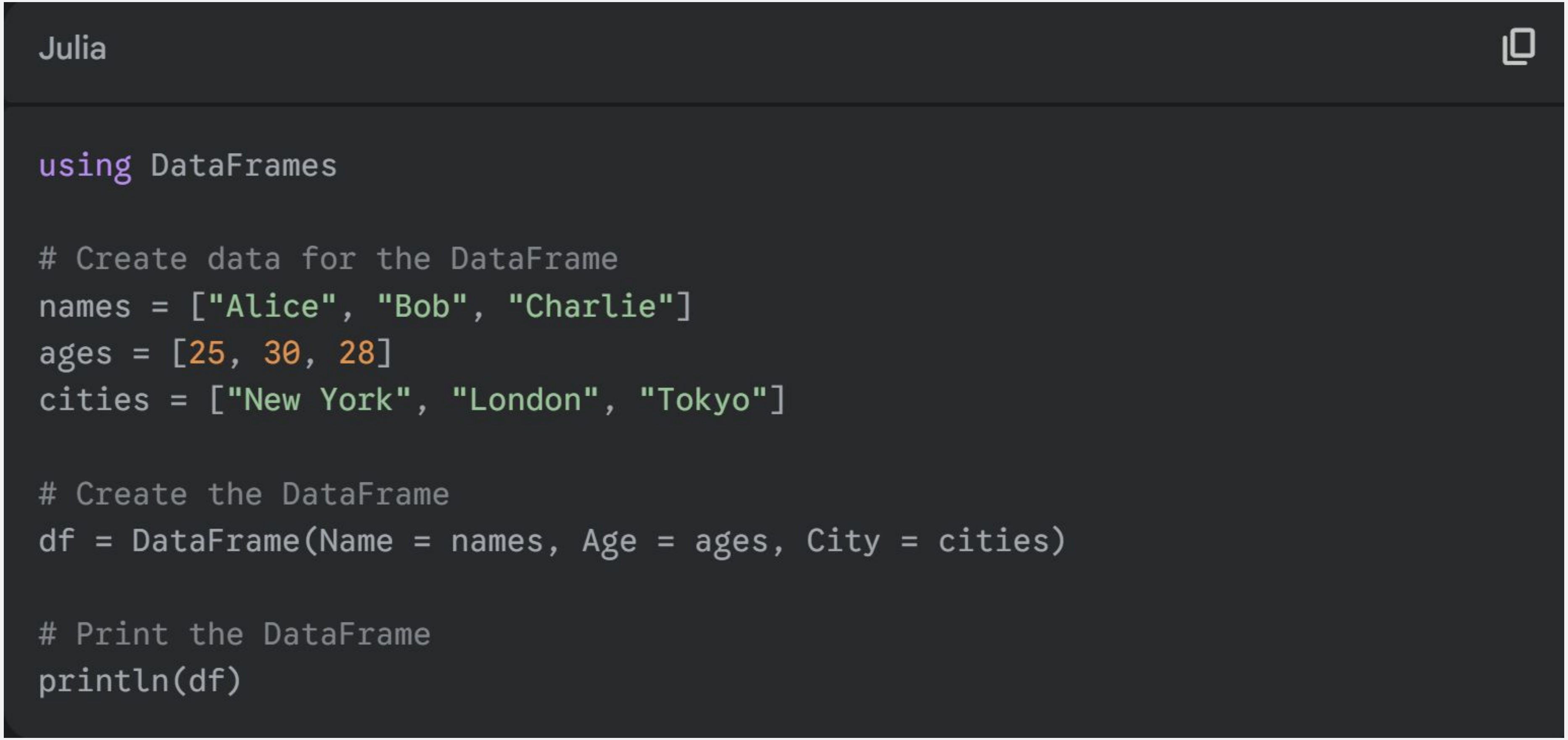
The Basics

DataFrame

Install package you need for your program


Julia

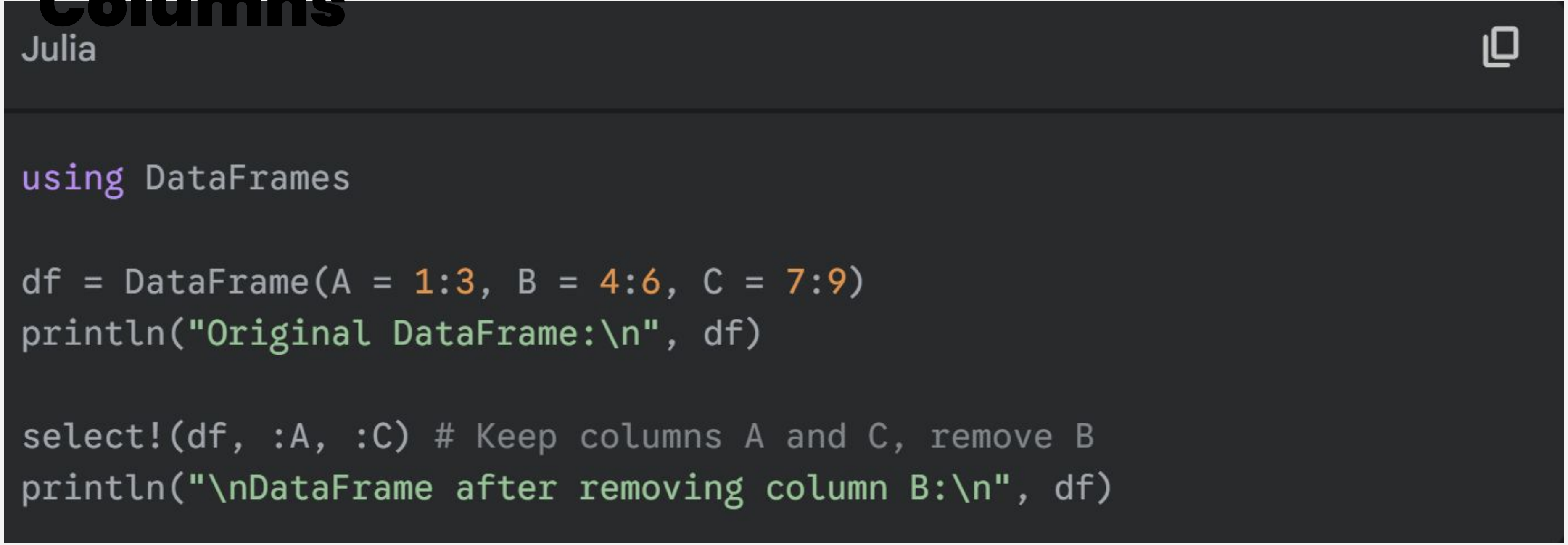
```julia
using Pkg
Pkg.add("DataFrames")
```


Julia

```julia
using DataFrames

# Create data for the DataFrame
names = ["Alice", "Bob", "Charlie"]
ages = [25, 30, 28]
cities = ["New York", "London", "Tokyo"]

# Create the DataFrame
df = DataFrame(Name = names, Age = ages, City = cities)

# Print the DataFrame
println(df)
```

- select!
- filter!

Select Columns



Julia

```julia
using DataFrames

df = DataFrame(A = 1:3, B = 4:6, C = 7:9)
println("Original DataFrame:\n", df)

select!(df, :A, :C) # Keep columns A and C, remove B
println("\nDataFrame after removing column B:\n", df)
```

Filter Records

Julia

```julia
using DataFrames

df = DataFrame(ID = 1:5, Value = [10, 20, 30, 40, 50])
println("Original DataFrame:\n", df)

filter!(row -> row.ID != 3, df) # Remove row where ID is 3
println("\nDataFrame after removing row with ID 3:\n", df)
```

Key Takeaway

- Tradeoff 🤫
  - Speed vs. Syntax (Easy to Learn)
- Growing community
- Julia for popular for numerical and scientific computation
- Explore the Julia **docs**: https://docs.julialang.org/en/v1/

Julia For Beginners 🤫

#DataRockie #Bootcamp11