

Tutorial II.1 - Variables and Types

Applied Optimization with Julia

Introduction to Variables and Types in Julia

Welcome to this interactive Julia tutorial which introduces the basics of variables and types. Understanding variables and their types is crucial as they are the building blocks of any program. They determine how data is stored, manipulated, and how efficiently your code runs.

This script is designed to be interactive. Follow the instructions, write your code in the designated code blocks, and then execute the corresponding code. Each exercise is followed by an `@assert` statement that checks your solution.

Note

Always replace 'YOUR CODE BELOW' with your actual code.

Section 1: Variables

Introduction to Variables: A variable in Julia can be used to store data like numbers, text, or more complex types. To declare a variable in Julia, simply assign a value to a name: `variable_name = value`.

Exercise 1.1

```
# Declare a variable named 'x' and assign it the value 1
# YOUR CODE BELOW

# Test your answer
@assert x == 1 "Check again, the value of x should be 1. Remember to assign the value
↳ directly to x."
println("Great, you have correctly assigned the value $x to the variable 'x'.")
```

Exercise 1.2

```
# Declare a variable named 'hi' and assign it the string "Hello, Optimization!"
# YOUR CODE BELOW

# Test your answer
@assert hi == "Hello, Optimization!" "Make sure the variable 'hi' contains the exact
↳ string \"Hello, Optimization\\\"!"
println("Good, the variable 'hi' now states \"\\$hi\\\".")
```

Section 2: Basic Types

Julia is a dynamically typed language, yet it allows you to indicate types explicitly. Understanding types is crucial for writing efficient Julia code. Common basic types include Integers, Floats, Booleans, and Strings. To check the type of a variable, use the `typeof()` function.

Julia is a dynamically typed language, yet it allows you to indicate types explicitly. Understanding types is crucial for writing efficient Julia code. Common basic types include Integers, Floats, Booleans, and Strings. To check the type of a variable, use the `typeof()` function.

Exercise 2.1

```
# Create an Integer variable 'answerUniverse' and set it to 42
# YOUR CODE BELOW

# Test your answer
@assert answerUniverse == 42 "The variable 'answerUniverse' should hold 42."
println("Great, the answer to all questions on the universe is $answerUniverse now.")
```

Exercise 2.2

```
# Create a Float variable 'money' and set it to 1.35
# YOUR CODE BELOW

# Test your answer
@assert money == 1.35 "The variable 'money' should hold the Float64 1.35."
println("Perfect, the you have stored $money in the variable 'money'.")
```

Exercise 2.3

```
# Create a Boolean variable 'isStudent' and set it to true
# YOUR CODE BELOW

# Test your answer
@assert isStudent == true "The variable 'isStudent' should be set to true."
println("Correct, you are a student now.")
```

Conclusion

Congratulations! You have completed the first tutorial on Variables and Types. You've learned about the basics of variables, integers, floats, booleans, and strings. Continue to the next file to learn more.