TEAM MEMBERS:

- ANANTHA NARAYANAN S V (E0119008)
- APARNA S (E0119020)
- KARAN V (E0119039)
- SATHISHKUMAR M (E0119052)
- SIVANT M (E0119004)

FROM

B.TECH CSE (AI & ML)

JULIA LEARNING APP

CSE 220 WEB PROGRAMMING & SCRIPTING

II YEAR V QUARTER

JULIA LEARNING APP

MODULES:

- *Declarative Routing*
- *Julia emulator(Repl)*
- *Conditional Rendering*
- *State Management*
- *Progressive web app (PWA)*
- *Bootstrap 5*

TECHNOLOGIES UTILISED:

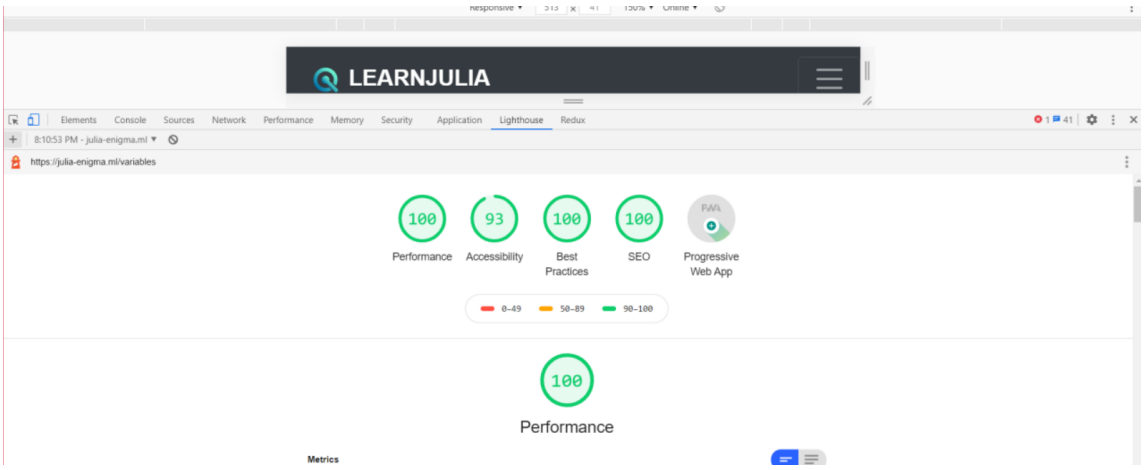
- *Javascript (ES 8)*
- *Vue JS*
- *Vuex*
- *Vue Router*
- *Bootstrap*
- *Firebase*

FEATURES & PERFORMANCE:

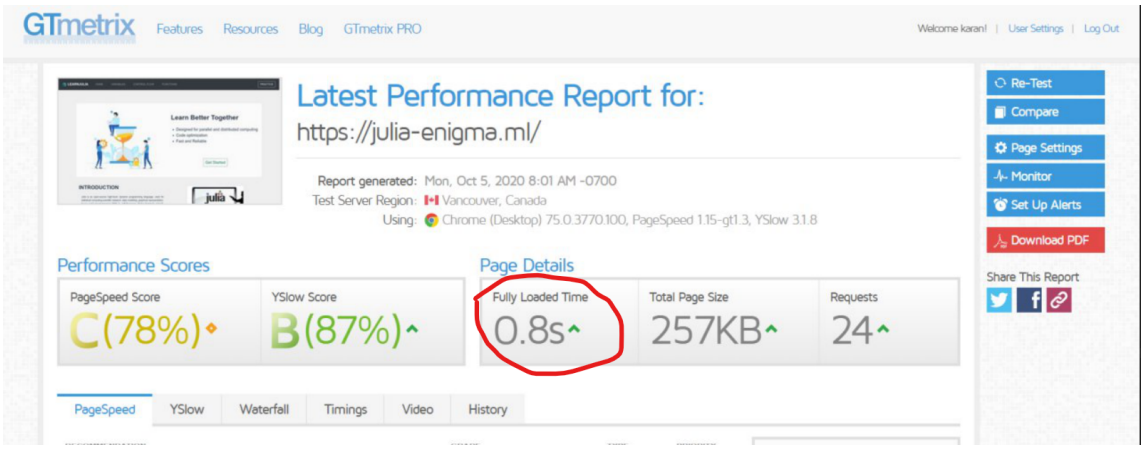
- *Responsive UI design*
- *User friendly*
- *Search engine optimization maxed*
- *Fast load time*
- *PWA offline support*

BENCHMARK

LIGHTHOUSE:



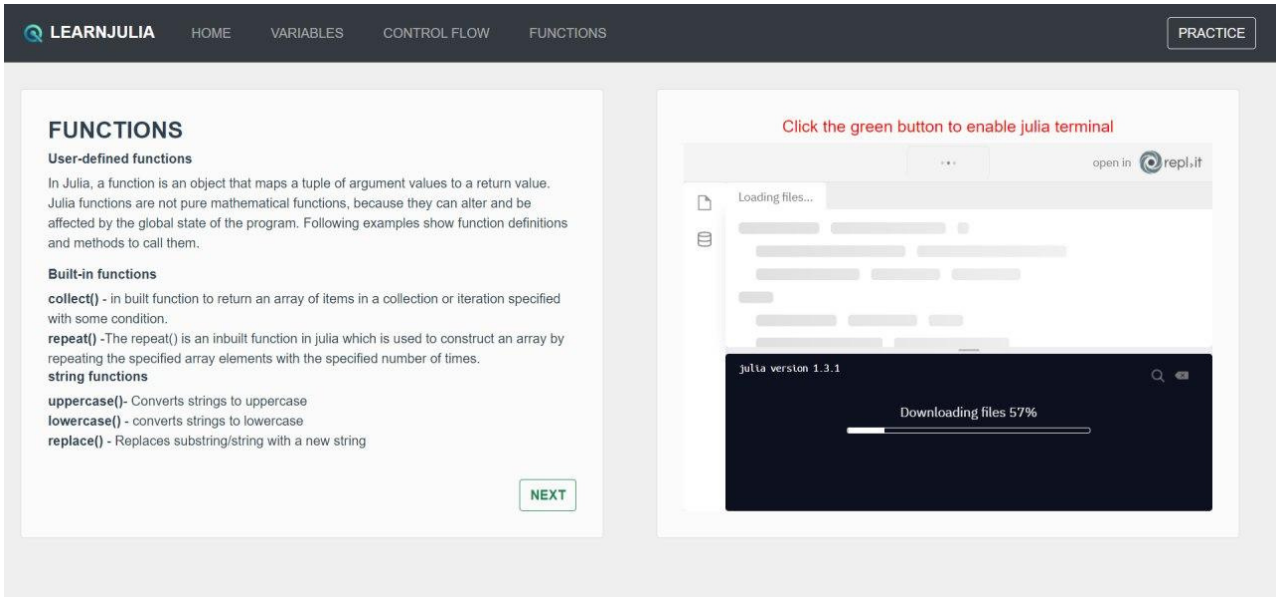
GTMETRIX:

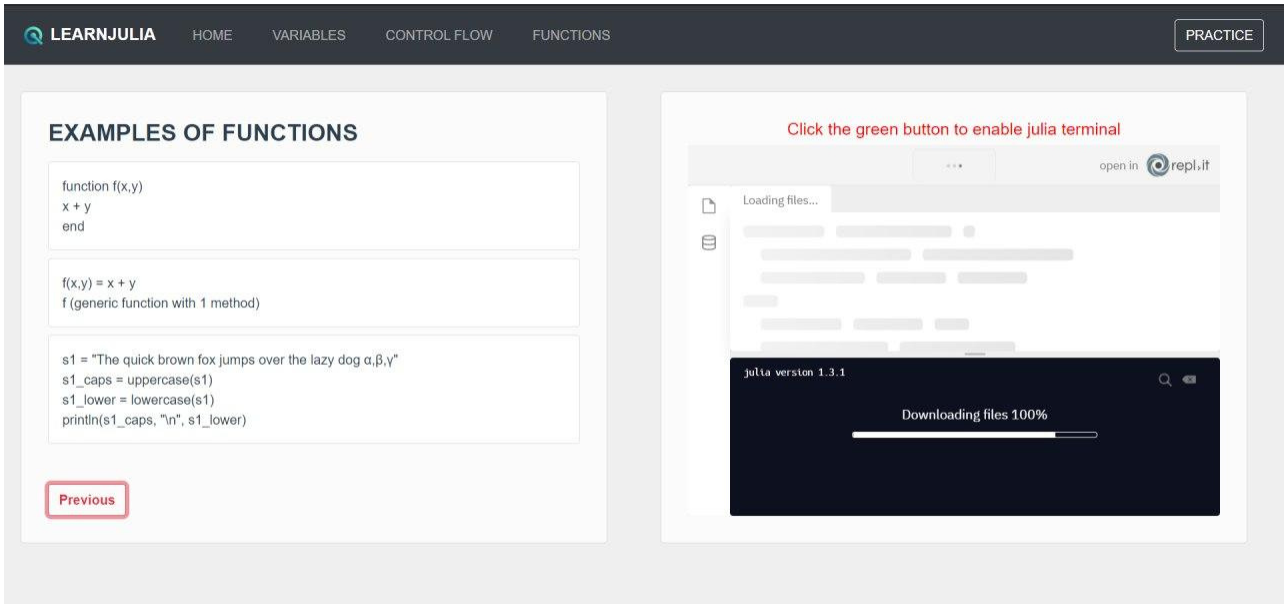


DEMO

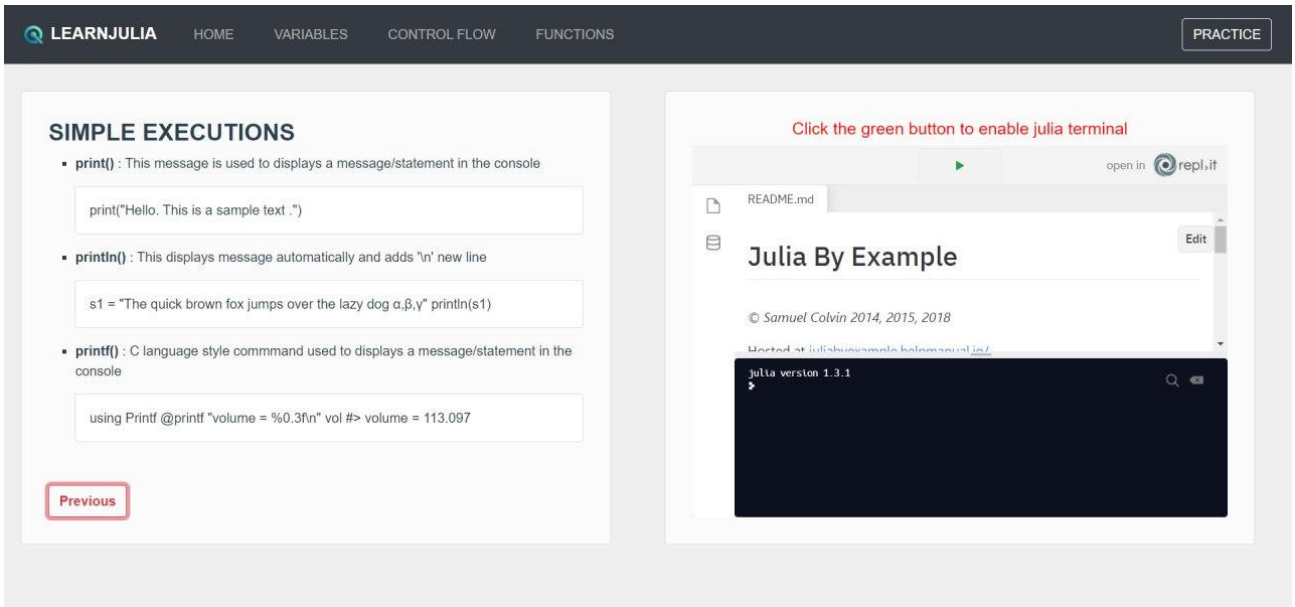
BROWSER VIEW

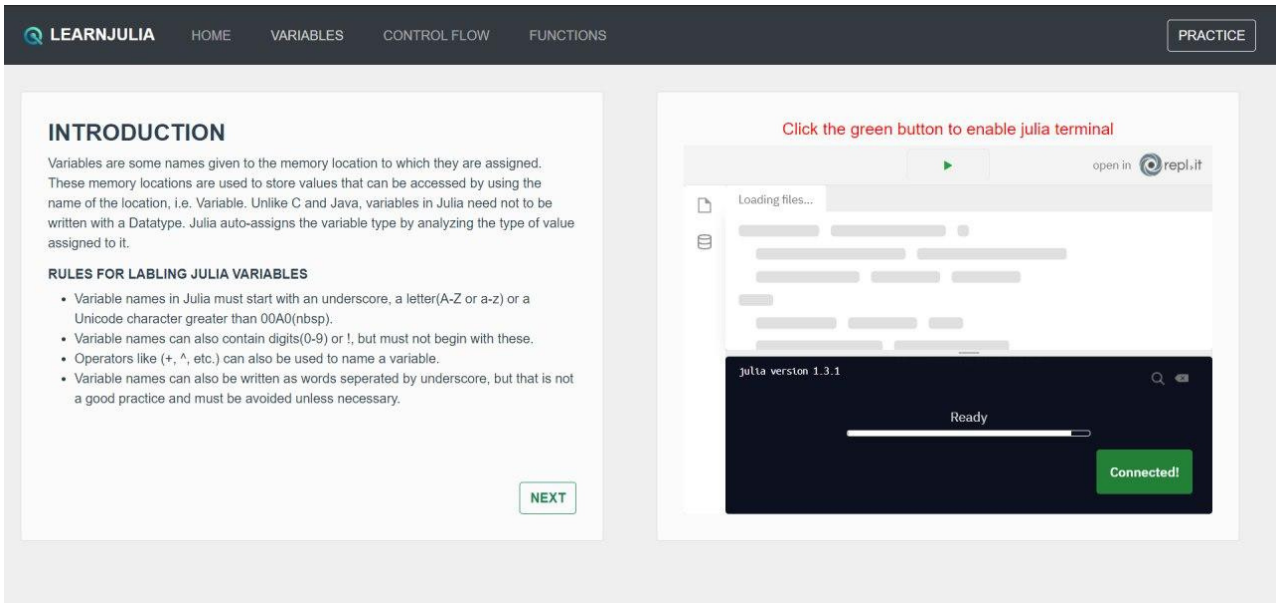
FUNCTIONS


LEARNJULIA
HOME
VARIABLES
CONTROL FLOW
FUNCTIONS
PRACTICE
FUNCTIONS
User-defined functions
In Julia, a function is an object that maps a tuple of argument values to a return value. Julia functions are not pure mathematical functions, because they can alter and be affected by the global state of the program. Following examples show function definitions and methods to call them.
Built-in functions
collect() - in built function to return an array of items in a collection or iteration specified with some condition.
repeat() -The repeat() is an inbuilt function in julia which is used to construct an array by repeating the specified array elements with the specified number of times.
string functions
uppercase()- Converts strings to uppercase
lowercase() - converts strings to lowercase
replace() - Replaces substring/string with a new string
NEXT
Click the green button to enable julia terminal
open in repl.it
Loading files...
julia version 1.3.1
Downloading files 57%

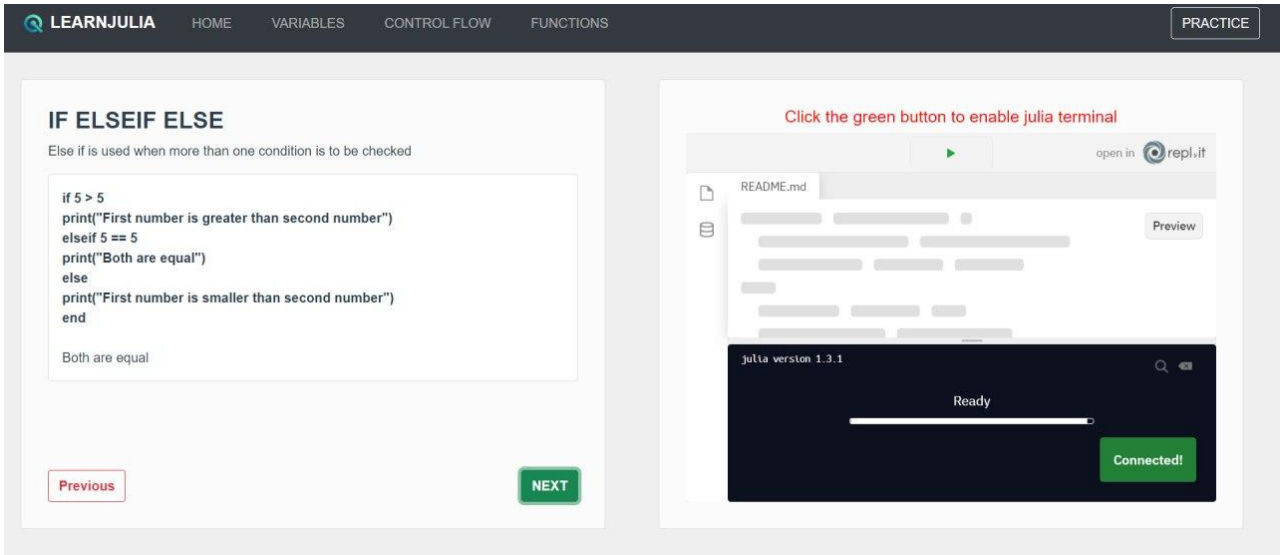

LEARNJULIA
HOME
VARIABLES
CONTROL FLOW
FUNCTIONS
PRACTICE
EXAMPLES OF FUNCTIONS
function f(x,y)
x + y
end
f(x,y) = x + y
f (generic function with 1 method)
s1 = "The quick brown fox jumps over the lazy dog α,β,γ"
s1_caps = uppercase(s1)
s1_lower = lowercase(s1)
println(s1_caps, "\n", s1_lower)
Previous
Click the green button to enable julia terminal
open in repl.it
Loading files...
julia version 1.3.1
Downloading files 100%

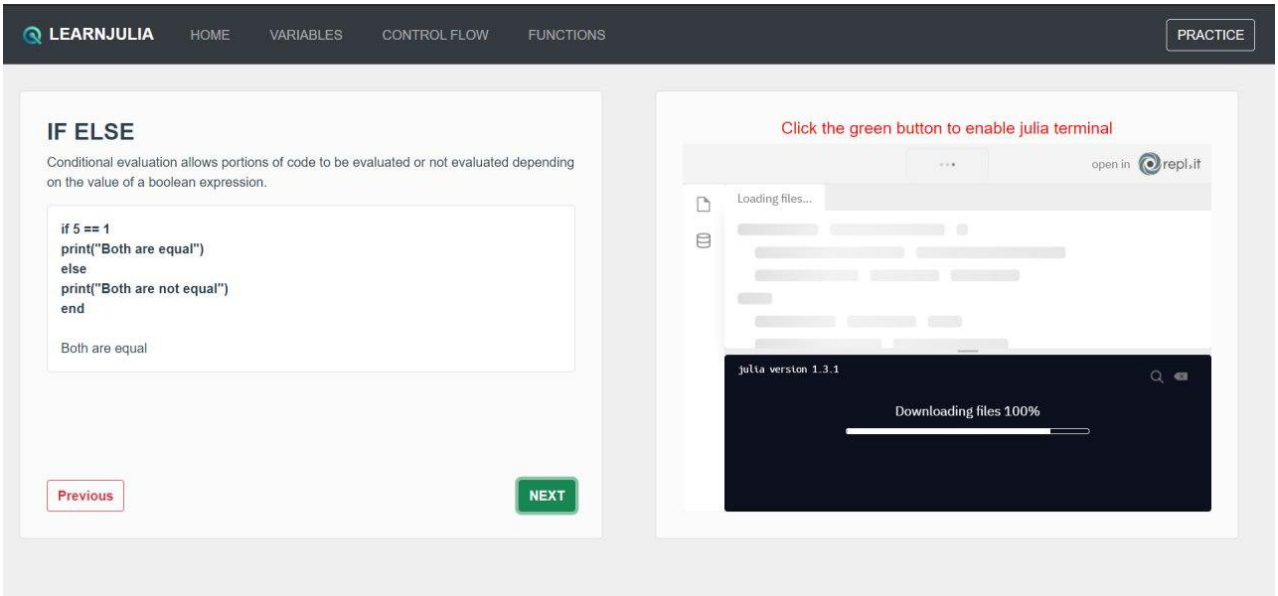
VARIABLES


LEARNJULIA
HOME
VARIABLES
CONTROL FLOW
FUNCTIONS
PRACTICE
SIMPLE EXECUTIONS
print() : This message is used to displays a message/statement in the console
print("Hello. This is a sample text .")
println() : This displays message automatically and adds '\n' new line
s1 = "The quick brown fox jumps over the lazy dog α,β,γ" println(s1)
printf() : C language style commmand used to displays a message/statement in the console
using Printf @printf "volume = %0.3f\n" vol #> volume = 113.097
Previous
Click the green button to enable julia terminal
open in repl.it
README.md
Edit
Julia By Example
© Samuel Colvin 2014, 2015, 2018
julia version 1.3.1

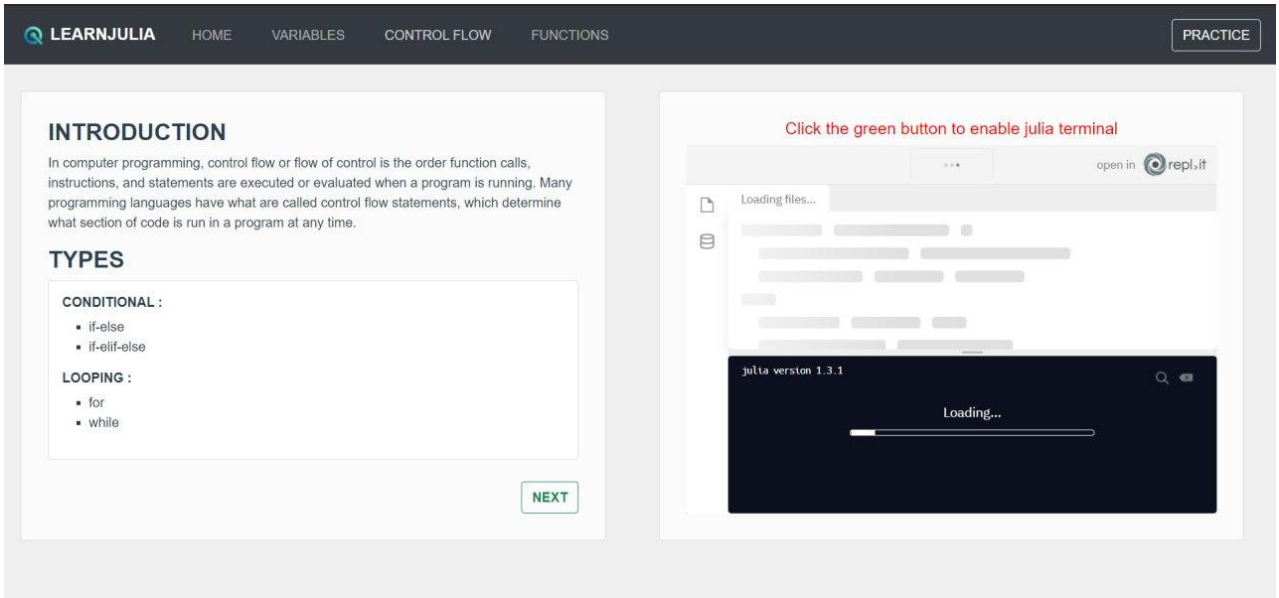

LEARNJULIA
HOME
VARIABLES
CONTROL FLOW
FUNCTIONS
PRACTICE
INTRODUCTION
Variables are some names given to the memory location to which they are assigned. These memory locations are used to store values that can be accessed by using the name of the location, i.e. Variable. Unlike C and Java, variables in Julia need not to be written with a Datatype. Julia auto-assigns the variable type by analyzing the type of value assigned to it.
RULES FOR LABLING JULIA VARIABLES
Variable names in Julia must start with an underscore, a letter(A-Z or a-z) or a Unicode character greater than 00A0(nbsp).
Variable names can also contain digits(0-9) or !, but must not begin with these.
Operators like (+, ^, etc.) can also be used to name a variable.
Variable names can also be written as words seperated by underscore, but that is not a good practice and must be avoided unless necessary.
NEXT
Click the green button to enable julia terminal
open in repl.it
Loading files...
julia version 1.3.1
Ready
Connected!

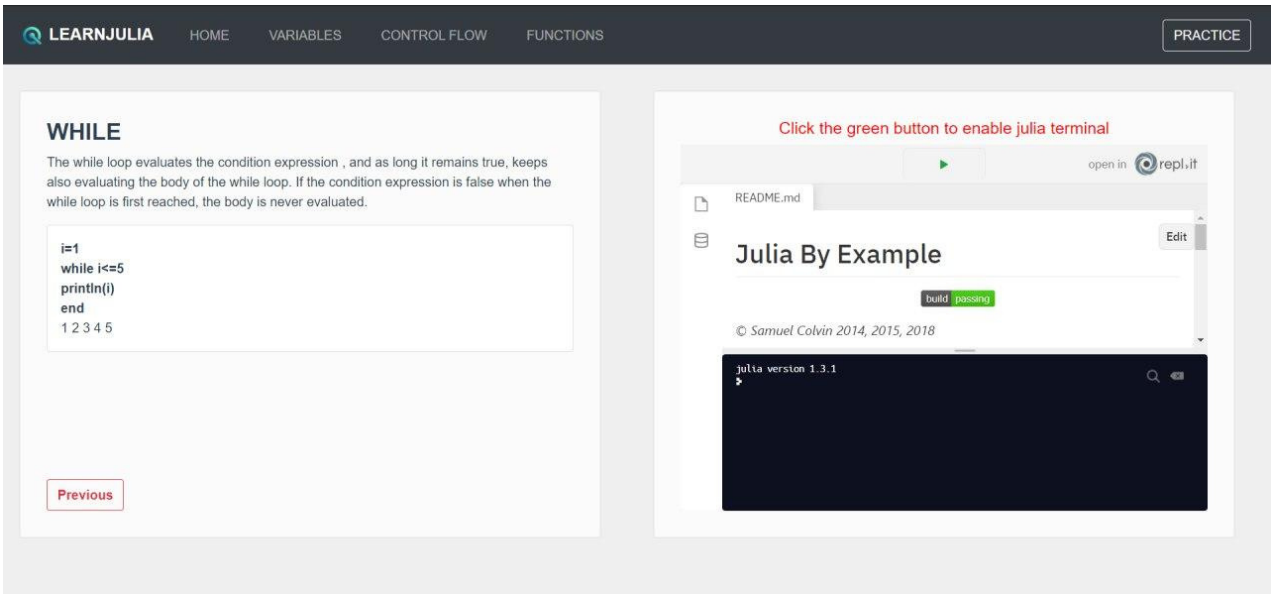
CONDITIONAL STATEMENTS

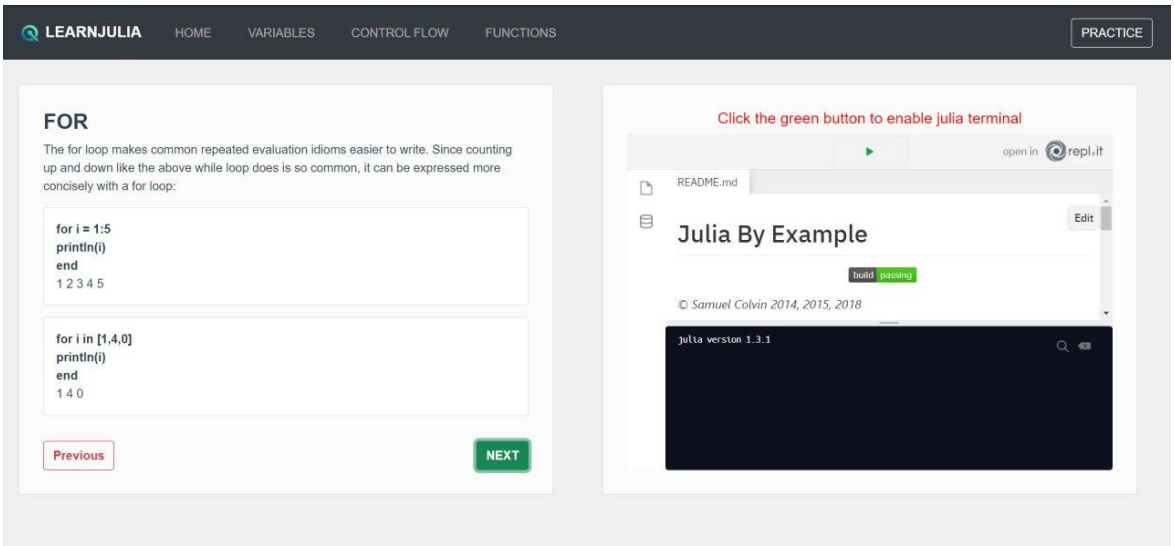




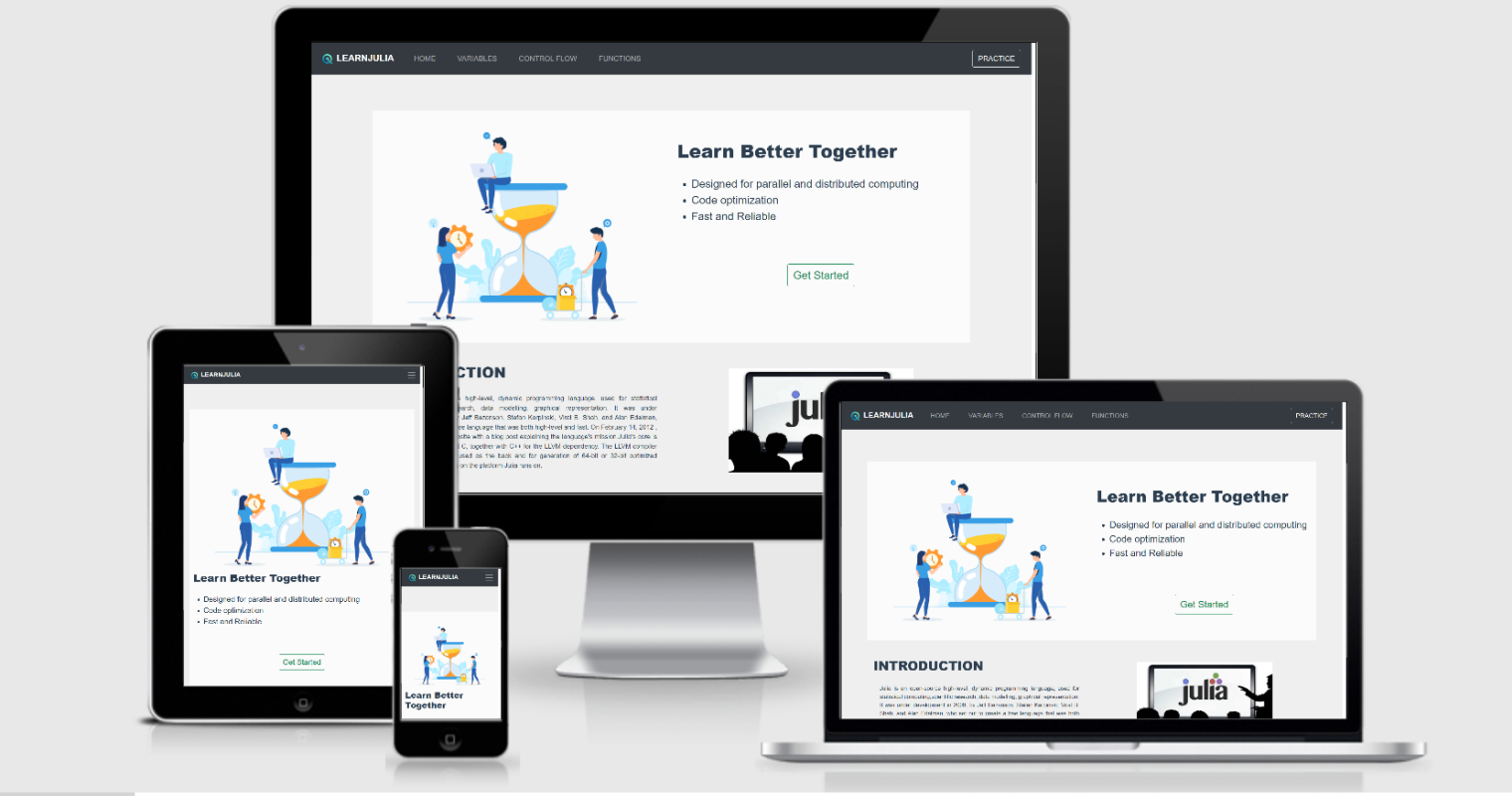
CONTROL FLOW







RESPONSIVE LAYOUTS



LEARN JULIA @ julia-engima.ml