Quick Overview

This lesson gives a quick overview of the concepts discussed in this course.

Concept	Explanation
Variable	holds data or reference to data
Data type	defines the type of value assigned to a variable. It can be primitive, or non-primitive. Primitives includes, integers, floats, character, boolean, arrays, tuples and string.
Array	sequence of elements of the same type
Tuple	sequence of elements of different types
Operators	tell the compiler to perform specific operations
Conditional Statements	statements that execute if the associated condition evaluates to true
Loops	blocks of code that keep on executing until a specific condition becomes true
Functions	a resuable piece of code that is used to perform a set of tasks
Strings	a sequence of characters; primitive, i.e., String literal(&str) or non-primitive i.e., String object (String)

Vector	A resizable array
Structs	A composite data type which contains a number of key value pairs, key being the item name and value the data type of the item
Enum	composite data type which contains definite values called its variants
Trait	define an interface for multiple structs
Generics	generalize a data type for struct, enum, trait, functions , arrays and collections
Stack	holds variables having primitive data type (size is known)
Heap	holds variables having non-primitive data type (size is unknown)
Ownership	defines which variable will hold the value. Primitive data type copy their ownership and non-primitive data move their ownership during assignments
Borrowing	share a variable value or share and mutate a variable value
Lifetime	defines the scope for which the reference is valid

The next lesson gives you a quick guide about your future prospects after learning Rust in the programming field.

Or, you can go back to the Learn Rust from Scratch course homepage.