Ruby is a very high level language. It follows similar syntax to Java and Python because of this and the fact that it is object oriented. However, to make it easier to pick up, it has some additional features that make it unique to any other language.

Ruby was created in the 1990s by Yukihiro Matsumoto. It was created for just-starting-out programmers so they could improve their knowledge on coding. In order to do this, Ruby intends to eliminate the idea of programming being a difficult task. This means that for people who have a background in programming typically can pick the language up rather quick.

Ruby has many different control structures. Most of which are relatively typical of modern languages. One that stands out, however, is the many different ways one can use loops.

Ruby has many legal ways to do loops. This includes simple loops, while loops, until loops, do/while loops, for loops, iterators, and recursion. We’re going to be talking in depth on a few of these looping mechanisms.

The simplest loop in Ruby can be done as shown. This type of loop will run indefinitely without the user manually stopping it, so it is common to add in conditionals with break statements. By doing so, once the condition is reached, the break statement can break your program out of this loop.

Ruby also has the standard while and do/while loops as shown here. This gives the programmer the option of running code up until the condition is no longer met.

Until loops work exactly opposite of a while loop, so they will run until the condition is met.

For loops are beneficial in avoiding infinite loops because they iterate i over a range of values, and end once each iteration has passed.

Ruby supports essentially all standard data types. In particular, we’re going to focus on Arrays, Hashes, and the many different types of Numbers Ruby has. Uses the nil value instead of null.

Arrays are implemented in Ruby as most languages do. This means indexing will begin at 0, and getting to the next element can be done in increments of one. However an interesting feature of arrays that is slightly less common, is that Ruby allows for negative indexing. This means accessing an element at -1 will return the last element in the array.

Hashing is a common way to store data in many languages. In Ruby, one can declare a hash as shown in the example, or by adding values into an empty hash once at a time.

Ruby has many different subclasses of Numeric, this is how it maximizes the space of numbers in memory.

First off, the Integer class is a parent to Bignum and Fixnum. Fixnum is used for integers that are either 32 or 64 bits, while Bignum is used for big numbers.

Next is Floating point numbers. Floats in Ruby are considered to be imprecise, this means that saving it in memory may result in the value slightly changing. If we want a precise decimal number, we’ll have to use BigDecimal.

Ruby is object oriented and acts as such. It supports Classes, methods inside those classes, and Libraries that hold everything together.

Ruby is a perfectly object oriented language. This means everything you do revolves around classes. Within these classes, Ruby allows Data Encapsulation, Data Abstraction, Polymorphism, and Inheritance. One can initialize an object with the keyword “new”.

Methods are the standard way a programmer can break up parts of their Ruby code. Methods in Ruby can take in 0 or more parameters and always return some value.

With the many different subprogramming strategies in Ruby, it is important to note the different variable accessibilities in different situations. Local variables can only be seen within the method they are declared, while class variable can be seen by any object, but belong to the class they are declared in. Instance variables  are defined in object declaration and can only be seen to that object and any of its subclasses.

Methods always return something in Ruby, so it is common to end them with a return statement. Surprisingly however, this is not required. If a programmer decides not to return anything from a method, Ruby will automatically return the last statement’s value.

One is also allowed to return multiple values from a method. To do this, you can just list the values you want returned with commas in between each value.

In conclusion, Ruby is an ideal language for projects that need quick fixes. In general, the high level functionality of Ruby allows for programmers to get a lot done in very few lines of code. This is good for someone just beginning to learn how to code, in addition to being used by experienced programmers to get simple programs finished in a few hours of work. Both of these reasons to use Ruby is exactly why the language was created.