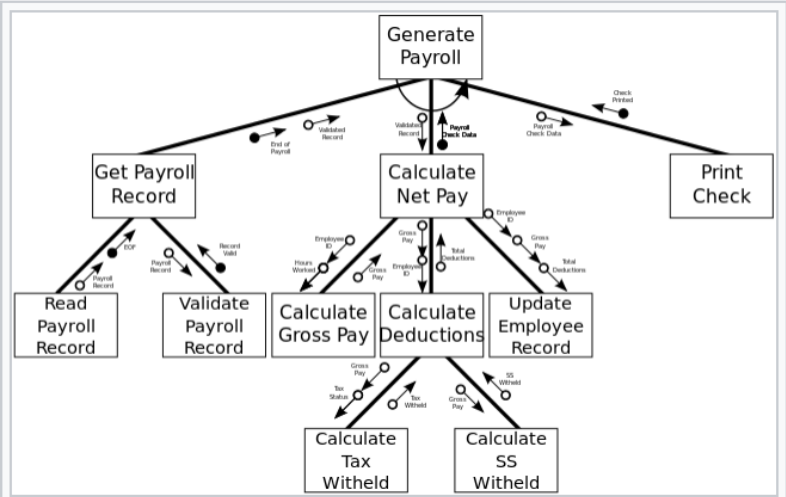
Week 1 - Topic 1: Introduction to the unit and Ruby

What is in this lesson?

- Unit Overview
- Assessment and Submission
- Getting Started
- Tasks for Week 1
- What to do in the first week?

Aims of the unit

This unit of study aims to introduce students to structured programming and design.



Example of a Structured Chart.[1]

Source: wikipedia

That means to teach you how to design and write code professionally in a structured/functional framework.
Directly Relevant languages (eg: C, Go, Lisp etc):

But indirectly also all Object Oriented languages, as the design principles largely carry across from Structured programming to Object Oriented programming (OO).

More on these two approaches later.

Unit Assessment

Each week there are tasks you do both in tutorials and outside tutorials.

Introduction to Programming - Tutorial and Task Timeline

Commencement weeks for tasks						Task Descriptions						
Week	Tut. Tasks	Pass Task	Credit Tasks	D grade	HD (Band 1	HD (Band 2	Tutorial Focus	Tutorial Tasks	Pass Tasks	Credit Tasks	Distinction Tasks	HD (Band 1)	HD (Band 2)
1	1.1T 1.2T	1.3P					Sequence and basic data types	Hello world Desk Check - Bill Total	Hello User (sequence)				
2	2.1T 2.2T	2.3P					Functions, parameters, return values. Structure charts.	Debug Using functions	Hospital Charges				
3	3.1T	3.2P	33.C				Loops/conditionals, Design	Silly name	SimpleMenu	Shape drawing			
4	4.1T 4.2T		4.3C	4.4D			Arrays, File Handling, Design	File handling Gosu cycle/loop		Shape Moving	Maze creation		
5	5.1T	5.2P	5.3C				Records, record file handling, Design	TrackFileHandling	Music Records	Hover Button			
6	6.1T	6.2P		6.3D			Data, References, arrays and searching	Array Search	AlbumFileHandling		Custom Program design		
7	7T - Test 1	7.1P	7.2C	7.3D			Graphical and Game programming	Test 1 (Class Task)	**MusicPlayer with Menu**	**SimpleGUIMusicPlayer**	**FullGUIMusicPlayer**		
8	8.1T			8.2D			Design Patterns/Coupling/Cohesion	Concept map			Food Hunter		
9	9.1T			9.2D	9.3HD		Testing/Debugging Strategies and	Fix-it			D Custom Code	HD Custom Program	
10	10T -Test 2 10.1T		10.2C		10.3HD	10.4HD	Recursion, Algorithms (sorting, stacks, Big O), benchmarking, profiling.	Test 2 (in Workshop) Python/C Hello World		Recursive Factorial		Maze search	Custom Project
11	11.1T	11.2P	11.3C	11.4D	11.5D	11.6HD	Other languages - Static Typing, Compiling	Learning Summary Draft	Python Program	PythonShapeMoving	C Program	Custom Program video	Custom Project Video
12	Task Sign Offs		12.1C				Testing, tools and project packaging	Task Sign Offs		Test Harness			
13	Reflection and Portfolio Submission							Reflection and Portfolio Submission					

Requirements for Pass Grade: All Pass Tasks Completed Satisfactorally
Requirements for Credit Grade: All Pass Tasks and All Credit Completed Satisfactorally
Requirements for Distinction Grade: All Pass Tasks and All Credit and All Distinction Tasks Completed Satisfactorally
Requirements for HD (Band 1) Grade: All Pass Tasks and All Credit and All Distinction and all HD (Band 1) Tasks Completed Satisfactorally
Requirements for HD (Band 2) Grade: All Pass Tasks and All Credit and All Distinction and all HD (Band 1) and all HD (Band 2) Tasks Completed Satisfactorally

NB: The Music Player is the Major Task at each grade level.

These tasks often build up your skills so you can put together more complex code with good design and coding practices.

Assessment overviews are on Canvas:

COS10009 Portfolio Process and Assessment Criteria.pdf

MarkingRubricSummary(4)-1.pdf

Task specifications are obtained and submitted through Ed

Lets have a quick look at these ....

Ruby - the main Programming Language used for the unit

Ruby is an interpreted, object oriented language which we will be using in a mostly structured way.

A fairly popular language due to Ruby On Rails (which both Doubtfire and Canvas are written in).

It is a dynamically typed language (as opposed to statically typed) – we learn more about what that means later.

Popular Sites that use Ruby

- Kickstarter

- AirBnB

- GitHub

- Shopify

- Hulu

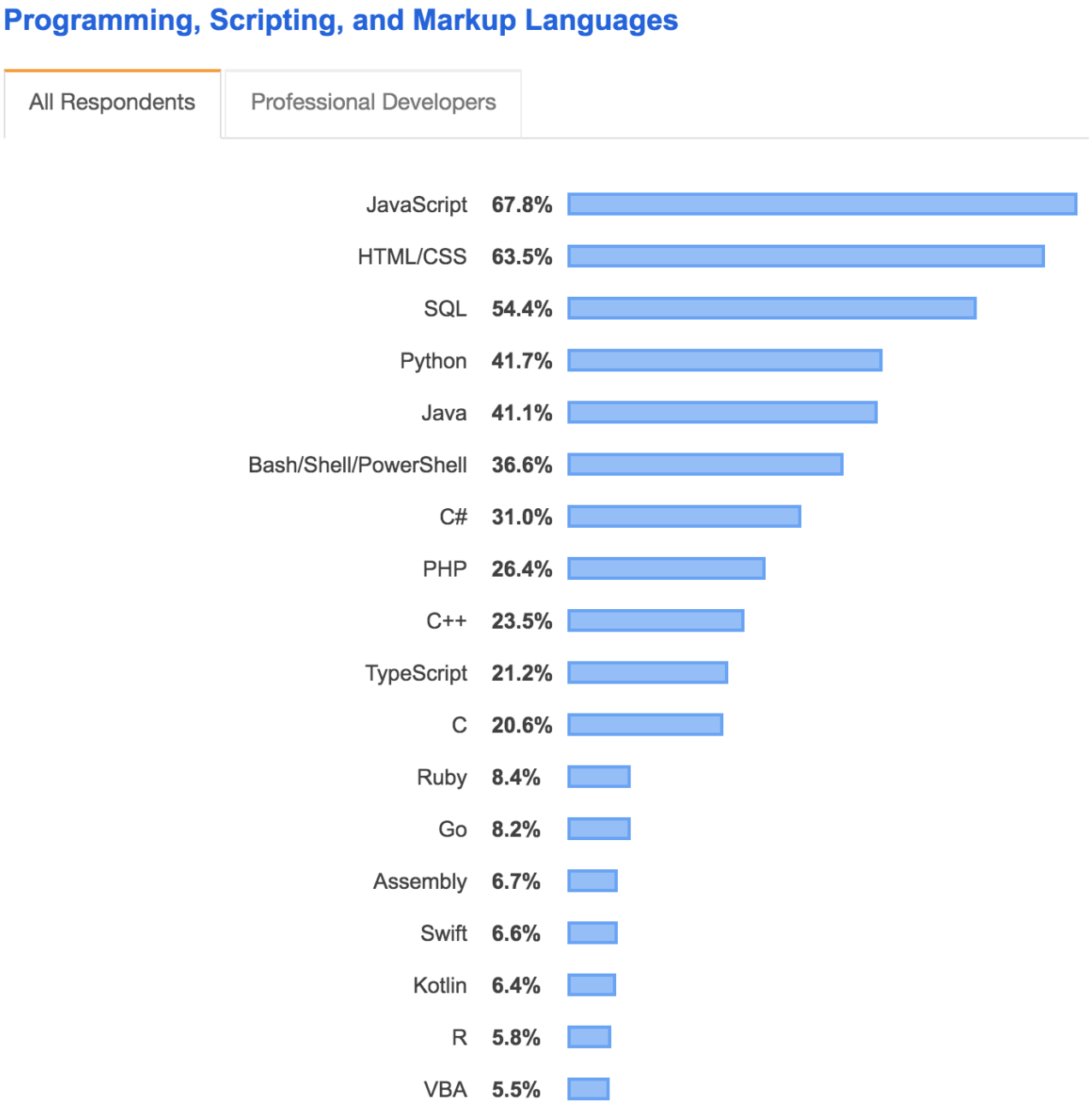
- Canvas

Where does Ruby fit?

Source: https://www.fullstackacademy.com/blog/nine-best-programming-languages-to-learn

Popularity of Ruby

Source: https://insights.stackoverflow.com/survey/2019:



Ruby behind the Scenes

According to Cascade Business News (Nov, 18, 2019):

'The healthcare industry is using Ruby to build advanced electronic health records (EHRs). This type of software is the backbone of the modern healthcare industry. Every single patient's appointment, diagnosis, prescription, and test is recorded in this system. This information is stored on outside servers and can be shared with other healthcare providers at the patient's discretion.

This type of electronic system removed a huge burden from patients, who previously had to carry around a single copy of their medical records to every new doctor.'

also:

'Medical technology companies and Ruby development services are using the language to build cutting-edge software as a medical device (SaMD).

SaMDs are any type of software that is used for medical reasons but is not part of a medical device itself. This includes mobile applications, image processing software, and tools that download diagnostic information onto computers'.

Why Ruby?

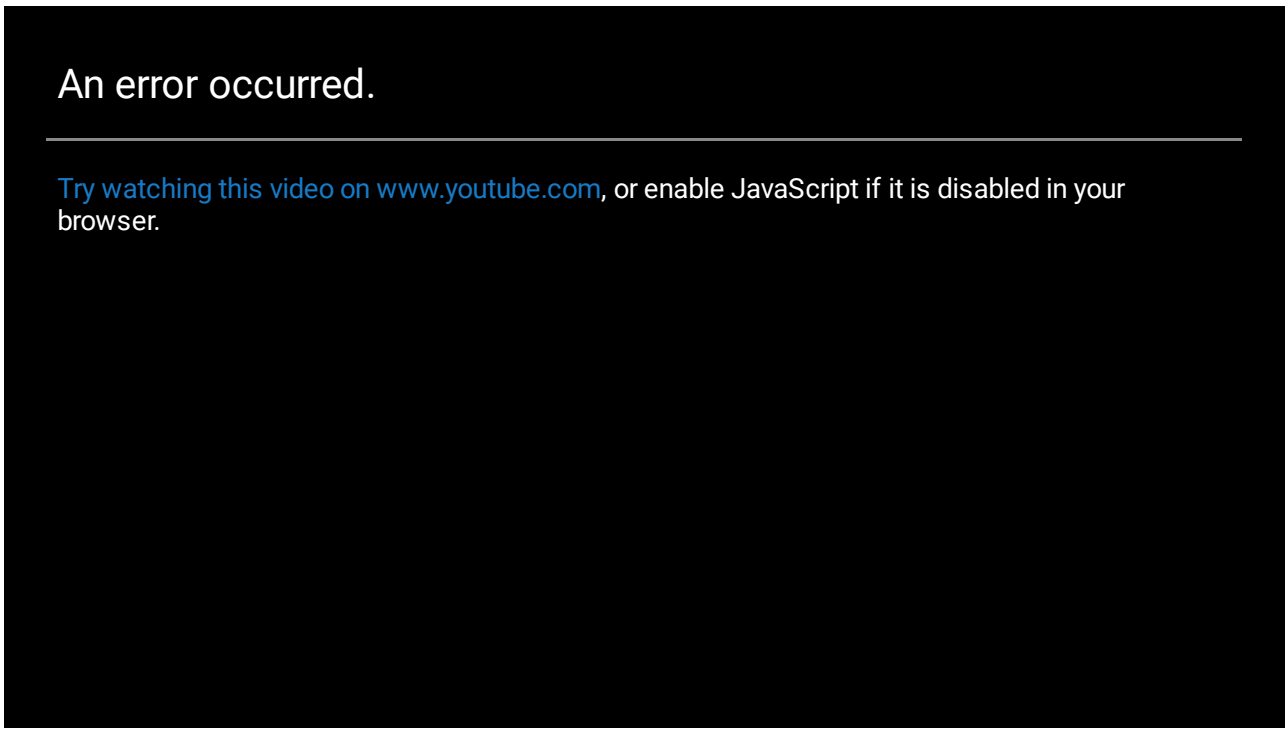
Leaving aside the popularity of Ruby, we are primarily concerned with using a language that is going to help you learn, keep you motivated and get you productive quickly whilst allowing for a wide range of different projects.

We are not specifically learning Object Oriented programming at this stage, so a language that supports structured programming is desirable.

Ruby is Object Oriented but we will be using it in a mostly structured/procedural way.

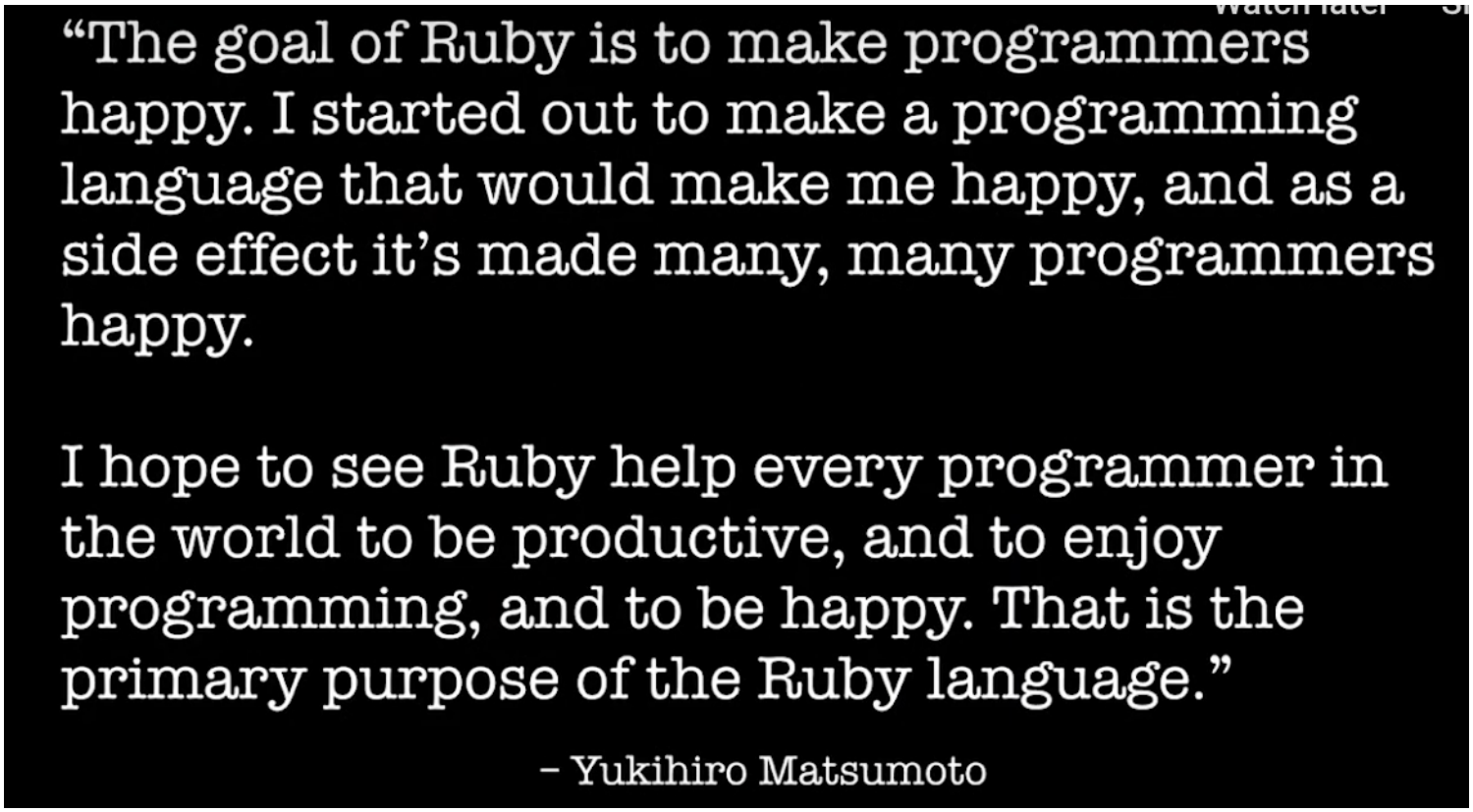
Ruby runs from the DOS window or the Terminal (Mac, OSX, Linux).

About Ruby (and Rails - which is built on Ruby)



This talks through motivations for writing Rails and using Ruby.

Watch this for about 5 to 10 minutes if you have time. One slide is below:

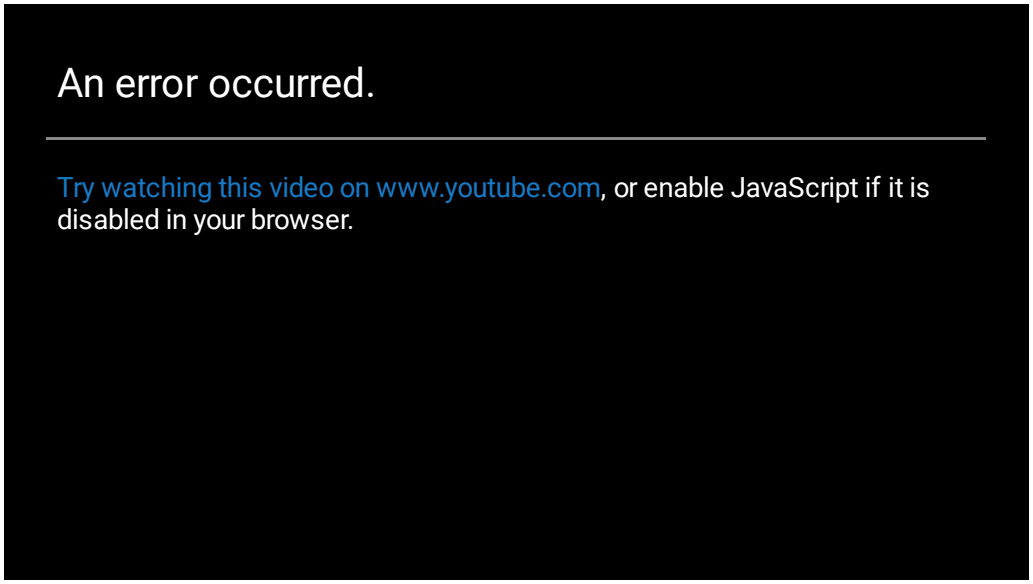


Using the DOS and Terminal Windows

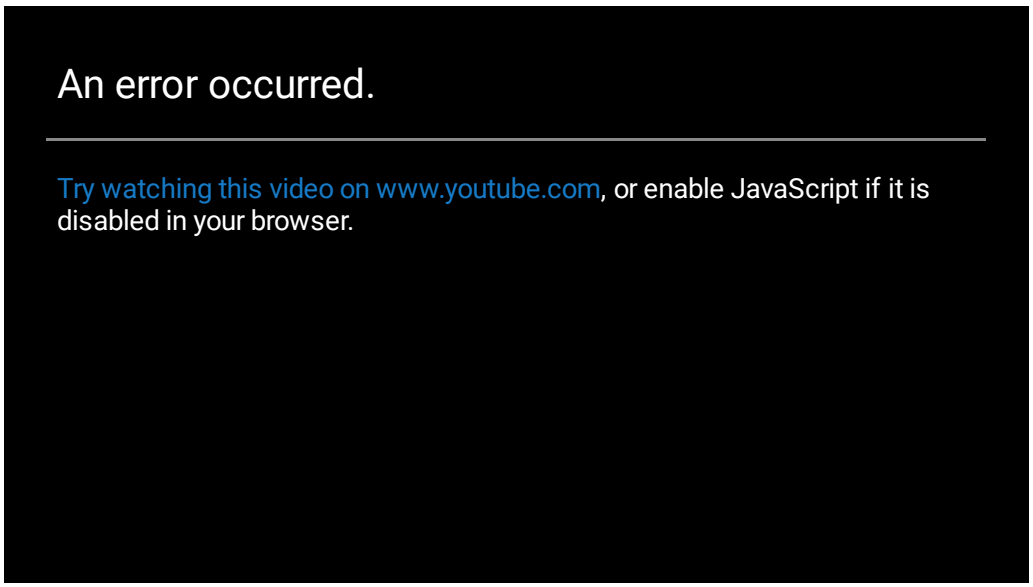
Windows: Go to Start->run type ‘cmd’, this will open the DOS window.
Basic DOS commands are:

- cd: change directory
- pwd: path to current directory
- dir: list files in directory

See the following youtube (also on Canvas):



If you are using Mac OS or Linux, see the following video:



Also see: Basic Unix Commands

Using the terminal

Try out the terminal commands below in the terminal window on the bottom right:

type 'pwd' and press enter

type 'ls - l' and press enter

type 'touch myfile' and press enter

type 'ls - l' and press enter again

What do you see?

Getting Started - installing Ruby

Windows

For Windows use the Ruby-installer here.

Select all the options in the installer when it runs.

Mac-OS

For Mac-OS build the code from source:

- Install x-code (developer tools) - this is free from the Apple App Store -
- Run the following command from the terminal window: 'xcode-select --install' (if you get an xcrun error - use the following command: 'sudo xcode-select -switch /')
- Install homebrew (if you do not have it already)
- Run homebrew install for ruby in the Terminal: *brew install ruby*

Install Homebrew as suggested at the top of the guide, then after installing rvm (follow the rvm install instructions) type the following:

brew install sdl2

Test Ruby

These instructions and links are also on Canvas – see the Software section.

Test your install: In Windows open a DOS prompt (windows->run "cmd") in Mac-OS run a terminal window.
type: *ruby -v*
This should tell you what version of Ruby you are running.

Install GOSU

In the Terminal (or DOS command window) type:

gem install gosu

Find the Ruby folder and then the sub-folder with the examples in it, run some of the examples from the terminal or prompt. You may want to create a folder in your documents called Ruby-code and put

it in there.

A Gosu tutorial is available here: https://onecore.net/gosu-ruby-2d-game-development.htm.

The API is here: http://www.rubydoc.info/github/gosu/gosu/

Getting Started: Trying Ruby online

https://www.tutorialspoint.com/execute_ruby_online.php

Use the API (Application Programmer Interface) documentation:
https://ruby-doc.org/

Or try Ruby below:

type `puts("Hello Everyone!")` then click Run.

Unit Assessment

This unit uses a Portfolio assessment strategy. This means:

- [ ] You can attempt a task and get feedback before submitting again
- [ ] There are no tests
- [ ] All the tasks can be submitted in the last two weeks
- [ ] You need to complete all the tasks up to, and including, the grade level you are aiming for.