PizzasOnly

Red Opal Innovations

Version 1.1

Table of Contents

[Scripting vs Programming Languages 3](#_Toc82250387)

[Background 3](#_Toc82250388)

[Framework and Integrated Development Environment 3](#_Toc82250389)

[Usage 3](#_Toc82250390)

[Object Model and Protocols 4](#_Toc82250391)

[Application Overview 4](#_Toc82250392)

[Current Model: 4](#_Toc82250393)

[Corrected pseudocode 5](#_Toc82250394)

[User Documentation 6](#_Toc82250395)

[How to use PizzaOnly script 7](#_Toc82250396)

[Content Control 8](#_Toc82250397)

[Content control 8](#_Toc82250398)

# 

# Scripting vs Programming Languages

## Background

Scripting and programming languages differ in the way they perform tasks. Programming languages use a set of instructions are usually complied and run as stand-alone programmes. A compiler is used to create a program from the written code. Traditional programming is general based on low level languages, while scripting prefers high level languages.

Scripting languages take their set of instructions and with the aid of an interpreter execute them. They are written in one language and are interpreted by another. They are usually executed line by line. For example, JavaScript must be incorporated with HTML which will then be interpreted by the Internet browser. Scripting languages are primarily used for web applications where speed of development is more that speed of execution

In a nutshell, languages used for traditional programming are complied, while languages used for scripting are interpreted rather than compiled. An interpreted language facilitates the implementation phase and allows quickly prototyping new functions. Complied languages can provide faster applications than interpreted languages.

# Framework and Integrated Development Environment

## Usage

For the project that is to be undertaken by PizzasOnly the use of a stand-alone application is not required. The premise is that a stand-alone application would require the building of said application and the resources required for it would be outside of the scope of this project. Therefor the idea of this project would be to use a simple JavaScript application that would run on any computer that has an Internet browser install, hence requiring no further hardware or extra software to be install.

A rapid application development approach to this project would be applied. The development of a JavaScript script written in Notepad ++ or Visual Studio Code would be advantageous as the required applications are already installed, take little to no time to load and only require the programmer to understand JavaScript. The two examples above are interpreters and only require the use of the Internet browser to execute the script, any errors that arise or are discovered through testing or normal running can be changed and the application could be reinterpreted to ensure that the application runs.

Whereas the use of writing a stand-alone application would require the installation of an Integrated Development Environment, for example Visual Studio, and would write the application on lower level. The use of the example above is an example of where the program would need to be compiled Any errors that are identified or that arise throughout testing or in execution will need to be made in the complier, then the program would need to be recompiled, and redeployed to the end user. Requiring more time to reinstall the application.

# Part 1 Question 3 here…

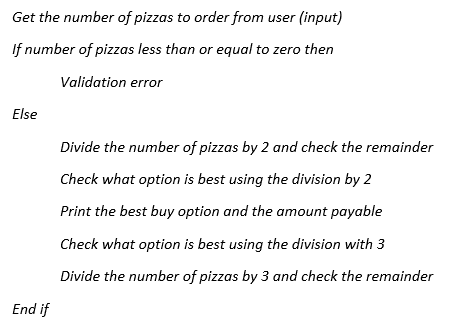
# Object Model and Protocols

For this project I will be using JavaScript as a script behind a HTML file hosted so we can determine the number of pizzas and the best deal.

AS mentioned above JavaScript is very versatile and quick to deploy. So I will be writing the code for this script in Visual Studio Code as I have it already available to me and the ease of use for creating the script for a quick turnaround for PizzasOnly. I will take the already generated pseudocode written, fix the errors that are noted below, and generate an working example of the pseudocode and then in turn generate the JavaScript to fit.*.*

# Application Overview

## Current Model:

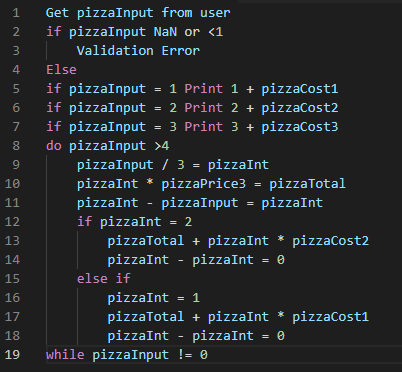


If we use the pseudocode above, we can see there are numerous errors.

The division of the number of pizzas should be started with three. This will take the highest possible number out, take a remainder if there is one, calculate the number of three’s that are in the count times the cost of the pizzas, then take the remainder and divide by 2. Calculate the number of two pizzas, and finally if we are left with one. Once all of the divisions have been performed we then will print the “best buy option”.

I propose that the following algorithm be used in creating the PizzasOnly webpage.

# Corrected pseudocode



*Fig 1 Pseudocode of script*

AS you can see from the algorithm that we take the user input if its bigger than 4, divide it by 3. The division of 3 will give us the number of 3’s in the pizza. We multiply that by 14 (being the cheapest option available) then we see if we have a remainder or 2 or 1. If we have a remainder of 2 we add the pizza cost for the 2 pizza deal to the number of 3 pizza deals we have, and if we have a remainder of 1 we add the 6.45 to the number of the 3 pizza deals we have.

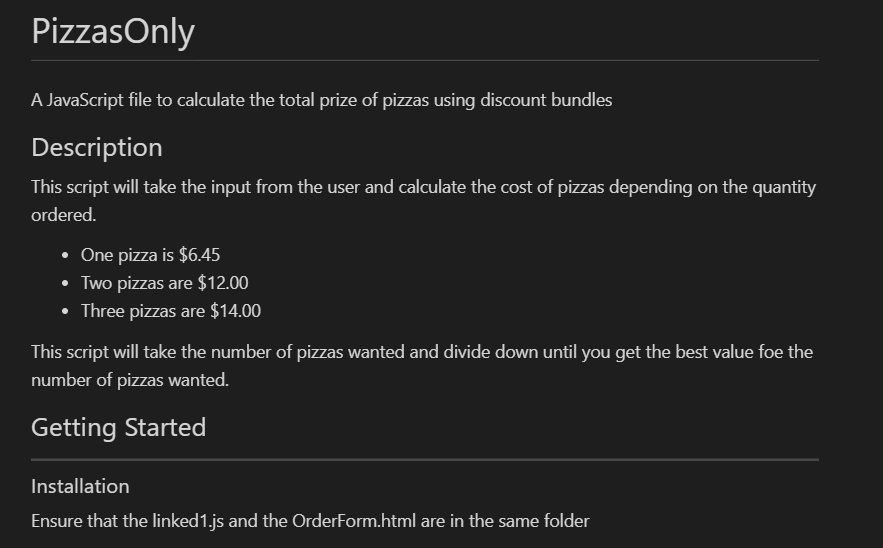
Text

Description automatically generated

*Fig 2 JavaScript code getting developed within the IDE*

# User Documentation

User documentation is provided in the attached README.md file.



*Above is a sample of the readme.md. See file for full readme.*

# Content Control

**Content Control:** Andrew Mills

## Content control

**Date**: 9/10/2021 – Version 1.2