

Term One Coder Academy Workbook

1. Components and concepts of web development and markup languages

Web development is generally referred to as the tasks and associated operations with building and maintaining functional web sites. Web developers use markup languages as the fundamental building blocks to create pages and sites on the internet (Kyrnin, 2021). Content in general needs structure to be presented. Markup languages give individuals a standard and formal way of structuring documents for machine-based processing. Each markup has its own set of rules for structuring content and documents defined by the language or some framework

(https://www.researchgate.net/publication/252519961_Structuring_Content_with_XML). Several markup languages have contributed to the growth of the internet and the development of the world wide web. Hypertext (HTML) is a markup language that is one of the web's fundamental building blocks. It serves the purpose of providing a method to access and link different types of information in a document on the internet. Each document acting as a web of information nodes which can be inspected at the will of the user (<http://info.cern.ch/hypertext/WWW/Proposal.html>). One of the main driving principles behind Hypertext is to provide the ability of uninterrupted movement between information sources. Hypertext basically provides the means of navigating information on the internet. Leading on from this, Hypertext provides a standard way of structuring and formatting documents so that web browsers, applications and users can read the content. This is achieved by using tags and attributes to define elements and describe the documents' logical structure or how it should be styled

(<https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/markup-languages>).

2. Features of technologies essential to the development of the internet (Packets, IP, Routing, DNS)

Packet switching is one of the underlying foundational technologies of the internet. The concept of packet switching arose from the need to have a system of communications that avoids a single point of failure and could operate reliably if damaged. Packet switching is a method used for communicating over a network (<https://www.carritech.com/news/packet-switching-explained/>). Packets are small chunks of information reasonably sized for ease of transmission over the physical network. Each packet is responsible for carrying data of variable length. On top of this, each packet has a source and a destination in the form of a unique identifying address. Packets are used to break large pieces of data down into smaller separable blocks so it can be transmitted over different routes. Following on from this, packets are an integral part of being able to transmit data without having to prearrange a connection to the device intended to communicate with (Univeristy, 2022). A network based on packet switching works better, faster and is cheaper for communicating data than traditional dedicated circuit switched networks. Because network bandwidth is shared, multiple simultaneous communications can occur. This is the key concept that makes the internet possible.

(https://web.archive.org/web/20140702224203/http://www.livinginternet.com/i/iw_packet_inv.htm).

IP addressing is the key to realising connections of separate LANs (local area networks), so long as hosts have a valid IP address and logical address planning, communication is possible. (<https://blog.router-switch.com/2021/01/the-development-of-network-hardware/>) IP (Internet Protocol) addresses are used

to distinctly identify devices connected to a network. An IP address is a string of numbers used to define the network and device locations for sending data. IP addresses are vital in communication over networks as they give a digital address for packets to be sent and received. Internet Protocol version 4 or IPv4 utilises 4 numbers separated by periods ranging from 0 to 255. As the number of devices connecting to the internet has exponentially increased since the inception of IP and IP addresses the need for more addresses has led to the creation of IPv6. IPv6 utilises 8 hexadecimal numbers allowing for a much larger sum of total addresses (Fox, 2020). IP addresses are classified as public, private, static and dynamic. Static IP addresses are created manually and do not change while dynamic IP addresses are assigned and are subject to change. Private IP addresses are allocated to a host for use inside a private network for communicating with other devices on the network is not assigned for communication with devices on another network

(https://isaacomputerscience.org/concepts/net_internet_ip_addresses?examBoard=all&stage=all).

The concept of using a router provides a system independent interface to the internet and is one of the driving factors contributing to its growth. A router is a piece of peripheral hardware that extends the capabilities of a network of computer systems. Its basic function serves to allow computers with different system configurations to communicate successfully (

<https://apps.dtic.mil/sti/pdfs/AD0686811.pdf>). Routing is defined as the process of moving packets from a source to a destination. In order to achieve this, the router holds a table in its memory that is comprised of a list of all the networks the device is connected to. The list also contains other important information relating to how busy each path is between networks at any current time. When data arrives as a packet, the router will open the header and read the destination. Following on from this, it then checks all the available routes to the specified address. Additionally, each path is examined for congestion and the packet is then sent on its fastest calculated path. Routers can also perform additional tasks such as exchanging protocol information over networks and filtering traffic. There are three primary classifications of routing which include Default, Static and Dynamic. Default routing is generally performed by a stub router which relies on a single route to all other networks. In static routing, a network administrator manually adds paths to routing tables and packets travel fixed routes. Dynamic routing is a process by which data is forwarded via routes that are calculated based on current network activity, routing tables are automatically updated based on the fastest route. (

<https://www.webopedia.com/definitions/routing/#:~:text=In%20networking%2C%20routing%20is%20the,device%20known%20as%20a%20router.>)

The Domain Name System or DNS for short, is a decentralised system for naming and identifying computers on the internet or other IP networks. Its purpose is to translate human readable host names into numerical IP addresses (https://en.wikipedia.org/wiki/Domain_Name_System). Information online is generally accessed using domain names which is a unique string of human readable text that refers to a specific numerical IP address. Domain names are divided into three sections separated by a period. Each identifying section is a descendant of the domain to its right. Domain names are an integral component of URLs or uniform resource locators (What is DNS?, n.d.).

3. Features of technologies essential to the development of the internet (TCP, HTTP, HTTPS, Browsers)

TCP or transmission control protocol is a set of rules defining how data is transferred over a network. TCP is one of the most used protocols for sending data. TCP works by establishing a connection between the client and server. Following on from this, it ensures the connection remains open until

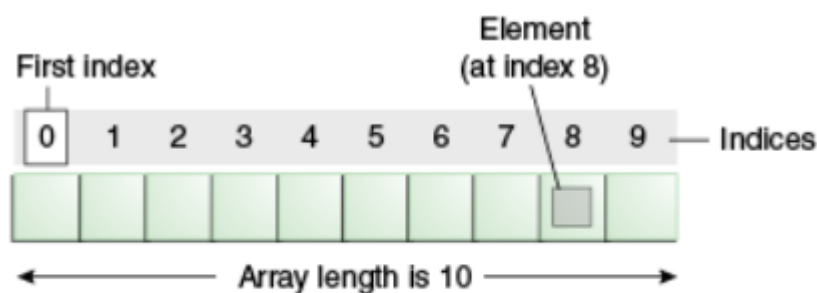
transmission is initiated. Lastly, TCP breaks down large chunks of data into smaller packets whilst still maintaining data integrity in the process (Fox, n.d.). Once the connection is established, the client and server are required to send each other synchronisation and acknowledgment packets. The purpose of this is to engage both ends for communication and to negotiate the limits of the network connection at either end before any meaningful data is sent (Shalom, 2020).

HTTP or Hypertext Transfer Protocol is the integral structure underpinning the delivery of data for the World Wide Web. HTTP is nested on top of TCP and has continued to evolve since its inception (MDN, n.d.). HTTP defines interactions between the client and web server by specifying permitted client requests and server response. The function of HTTP is to initiate a connection between the user's browser and the server for sending of html pages. When using HTTP, the connection between client and server is only maintained for the duration of each request/response. In addition to HTTP is HTTPS (Neeva, 2021). HTTPS or Hypertext Transfer Protocol Secure is an encrypted version of HTTP based on the sending and receiving of security certificates. This ensures users and clients identities are consistent and data remains secure (Cloudflare, n.d.).

Web browsers are a foundational part of the World Wide Web. Browsers for short, let users access files on a local network or the web. A browser is an application designed to process and render HTML into human readable content. Most browsers contain common features such as browsing history, private-mode and bookmarks (Techterms, 2014). Web browsers use HTTP or HTTPS to retrieve content from the web or from a local storage device for displaying to the user. In doing this, the browser is essentially acting as the client to contact a server and request information. (<https://smartbear.com/blog/history-of-web-browsers/>)

4. Data Structures in Ruby

An array is a fundamental data structure of programming and is useful for performing operations on whole collections of elements rather than individual variables. An advantage of using an array is once they are declared they can be reused multiple times. (Miralles, Arrays are data structures that represent a set of elements that can be selected by indices and computed upon at runtime during a programs execution. An array is used to store multiple elements of similar type together so they can be called upon by a common name (<https://www.geeksforgeeks.org/ruby-arrays/>). An array is a fundamental data structure of programming and is useful for performing operations on whole collections of elements rather than individual variables. An advantage of using an array is once they are declared they can be reused multiple times. (Miralles,). Arrays store data contiguously and unlike other low level languages Ruby uses dynamic arrays.



An array of 10 elements.

```
odd = [1, 3, 5, 7]

even = [2, 4, 6, 8]

numbers = [odd[0], even[0], odd[1], even[1], odd[2], even[2], odd[3], even[3]]

p odd,
  even,
  numbers
```

```
[1, 3, 5, 7]
[2, 4, 6, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
```

A hash is a data structure that is generally like an array. A Hash differs from an array in that, instead of indexing by integer a hash allows the use of key/value pairs. This is useful for dealing with large or nested arrays and aids in fast access of elements if the key is known. (Miralles, 2018). Hashes are the fastest storage structure for data and makes them essential for situations in which a computer system is interacting with data rather than a human (https://everythingcomputerscience.com/discrete-mathematics/Data_Structures/Hash_Table.html).

Hash Table

Like an array, except instead of indices you can set arbitrary keys for each value.

"apple"	4
"cherry"	2
"pecan"	5
"blueberry"	1

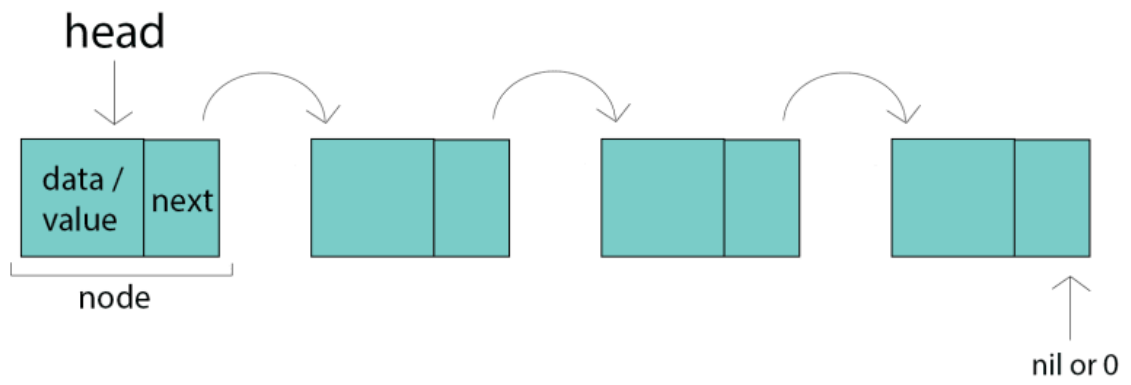
```
peoples_ages = { John: 38, Mary: 40, Jennifer: 25, Bill: 30 }

puts peoples_ages.map{|k,v| "Name: #{k}\t Age: #{v}" }.sort
```

```
Name: Bill      Age: 30
Name: Jennifer  Age: 25
Name: John      Age: 38
Name: Mary      Age: 40
```

A linked list is a data structure consisting of an element (node) which holds two key pieces of information, the data and a reference to the next node. It is a linear set of elements and is useful for storing and managing collections of data. <https://medium.com/analytics-vidhya/implement-a-linked-list-in-ruby-2aae925acd9c> . It is similar to an array however, the key difference lies in memory allocation. A linked list can be stored as just its nodes in separate blocks of memory. Which results in greater flexibility of memory allocation for other objects and doesn't use any extra space than is

needed.



```
3 # linked lists are typically created using two classes
4 # a node class and a list class
5
6 class LinkedList
7   # the starting point of the linked list
8   attr_accessor :head
9
10  # method to initialise and set head to nil(empty)
11  def initialize
12    self.head = nil
13  end
14
15  # method to add value,
16  def add(value)
17    # checks if head is empty(nil) then sets head to new node and sets value in new node
18    if head.nil?
19      self.head = Node.new(value, nil)
20    else
21      last_node = head
22      last_node = last_node.next_node until last_node.next_node.nil?
23      # end of list
24      last_node.next_node = Node.new(value, nil)
25    end
26  end
27
28  def delete(value)
29    return if head.nil?
30
31    node = head
32    prev_node = nil
33    until node.nil?
34      if node.value == value && !prev_node.nil?
35        prev_node.next_node = node.next_node
36      else
37        self.head = nil
38      end
39    end
40    prev_node = node
41    node = node.next_node
42  end
43
44  def find(value)
45    node = head
46    until node.nil?
47      return true if node.value == value
48    end
49    # go to next node
50    node = node.next_node
51  end
52  # otherwise the condition is false
53  false
54 end
```

```

56 # this is a node class
57
58 class Node
59   # the data or value and the pointer to the next node
60   attr_accessor :value, :next_node
61
62   # method to initialise a node
63   def initialize(value, next_node)
64     # set the value to value passed in
65     self.value = value
66     self.next_node = next_node
67   end
68 end
69 end

```

5. Compilers & Interpreters

An interpreter translates source code into binary. It does this sequentially by reading, translating and executing each instruction before moving on to the next line. Because of this, interpreted code is generally slower to run. Each time the program is run the script must be translated all over again. However, because of this errors may be easier to catch. A compiler is a program that translates source code from one programming language into a target language. Usually from a high-level language into a low-level language that is executable on a specific system. It uses the source code as its set of instructions which it performs a series of processes on to produce machine code. These processes include lexical, syntactic and semantic analysis, code generation and optimisation. Once the program has finished compiling, errors are reported and upon fixing will need to be compiled again. This cycle repeats until there is no errors in the program. While this may take time to compile, it results in faster run times as no further translation is needed.

6. Benefits and drawbacks of Ruby & C

Ruby is a popular programming language currently in use and its benefits include easily readable/writable syntax, open source with lots of easily accessible documentation and frameworks. One of the drawbacks of ruby is its scalability due to it being an interpreted language (McAllister, 2012). Ruby is a free, high-level, general-purpose language. Ruby is object orientated, simplistic in its design and emphasizes productivity. Ruby is a scripting language that can be used for a variety of applications such as front and back-end web development, data processing and automation(<https://developer.oracle.com/ruby/what-is-ruby/>). Some of it's features include portability, easy syntax and a rich set of built in functions just to name a few. (https://www.tutorialspoint.com/ruby/ruby_overview.htm). Because Ruby is a scripting language and must be interpreted, it may perform slower than a lower-level language (<https://ruby-doc.com/docs/ProgrammingRuby/html/trouble.html>). C is a programming language that is procedural in nature that is compiled to gain access to low-level memory. C is a structured language and is procedure orientated. Often referred to as a mid-level programming language due to it's code being suitable for high and low-level programs. C is based on ASCII characters; this gives the benefit of being able to run on a wide range of systems making it machine independent(<https://prosancons.com/computer/pros-and-cons-of-c-programming-language/>). Lack of exception handling is a disadvantage for C as bugs and errors may be harder to catch. Once the program has been written it must be compiled and all errors are shown at the same time. This can make checking

code problematic and time consuming (<https://www.includehelp.com/c/advantages-and-disadvantages-of-c-programming-language.aspx>).

7. nil

8. Control Flow

A script interpreter starts processing statements sequentially, starting at the first line of code. The sequential flow of execution can be determined with control statements (<https://www.ibm.com/docs/en/tivoli-monitoring/6.3.0?topic=language-flow-control-statements>). A control statement provides the means to alter and control the sequence of applied statements. There are many kinds of control flow some of those being sequencing, selection and iteration. Sequencing is the order in which statements are executed and refers to executing statements one after another. Selection is a decision-making process and the result changes the path of the program. Iteration is the repetition of some process and is often referred to as looping (<https://www.bbc.co.uk/bitesize/guides/z433rwx/revision/4>).

```

1  # Control flow examples
2
3  # this simple calculator shows all three examples Sequential, Selection and Iteration.
4  continue = true
5  while continue == true
6      puts 'enter 2 integers'
7      num1 = gets.chomp.to_f
8      num2 = gets.chomp.to_f
9      puts 'select an operation'
10     calculate = gets.chomp
11     case calculate
12     when '+'
13         puts "#{num1} + #{num2} = #{num1 + num2}"
14     when '-'
15         puts "#{num1} - #{num2} = #{num1 - num2}"
16     when '*'
17         puts "#{num1} * #{num2} = #{num1 * num2}"
18     when '/'
19         if num2.zero?
20             puts 'Undefined operation - cannot divide by zero'
21         else
22             puts "#{num1} / #{num2} = #{num1 / num2}"
23         end
24     when 'exit'
25         continue = false
26     else
27         puts 'Invalid operation try again'
28     end
29 end
30

```

9. Type Coercion

Type coercion happens because data is stored as one type but the context it is required in is a different data type. Type coercion is converting a datum to a different type of data in an expression (<http://faculty.salina.k-state.edu/tmertz/Java/041datatypesandoperators/07typecoercionandconversion.pdf>). In general, coercion refers to implicit conversion which is an automatic process run by the compiler to avoid loss of data when one or more data types are present in an expression (<https://www.geeksforgeeks.org/implicit-type-conversion-in-c-with-examples/>). Explicit conversion is

stated within a program and defined by the user.

```

1  # Type coercion examples
2
3  # Ruby will try to operate in the same data type
4
5  5 + 5 # This returns an integer as both operands are integers
6
7  1.0 * 9.0 # This returns a float as both operands are floats
8
9  6.66 - 1 # This returns a float, Ruby converts to match when possible
10
11 # explicit conversion
12
13 number = 5 # this is an integer
14 number.to_s # converted to a string

```

10. Data types

A data type defines the nature of a variable and its range of values. Data types also define memory allocation for each type. Data types are allocated different sized chunks of memory based on their type. In Ruby, data types are referred to as classes. Like other object orientated languages, a class in Ruby is a description of objects (Computer Science Programming Basics in Ruby).

Data types in Ruby include strings, numbers, symbols, arrays, hashes and Booleans. Each representing a specific type of data and what operations can be performed on them.

```

1  # Data types in Ruby, some examples;
2
3  # The variable "textual" is assigned the class string
4  textual = "I am a string of characters representing textual data"
5  # the variable "integer" is assigned the class integer
6  integer = 1
7  # "float" is a floating point number
8  float = 5.0
9  # "array" is an array containing numbers of different types
10 array = [integer, float]
11 # "hash" is a hash containing string: as the symbol and textual as the value
12 hash = {string: textual}
13 # Boolean true/false
14 array.include?(integer)

```

11. Code Error


```

1  class CustomerOrder
2
3      def initialize
4          time
5          date
6          order_number
7      end
8
9      def add_to_order
10         + menu_item
11     end
12
13     def delete_from_order
14         - menu_item
15     end
16
17     def add_notes
18         add_notes_to_order
19     end
20
21     def total_order
22         sum_cost
23     end
24
25     def account
26         paid_unpaid?
27     end
28
29     def status
30         time_created?
31         estimated_wait_time
32         preparing or cooking or ready
33     end
34
35     def print_order
36         all_menu_items_selected
37         total_price
38         time
39         date
40         order_number
41         order_notes
42         account
43     end
44
45 end
46
47

```

```

108 class Accounts
109     def initialize
110         new_account
111     end
112

```

```

48 class Menu
49     def initialize
50         name
51         ingredients
52         drinks
53         sides
54         entree
55         mains
56         kids
57     end
58
59     def add_ingredient
60         ingredient
61     end
62
63     def delete_ingredient
64         ingredient
65     end
66
67     def drink
68         size
69         Brand_name
70         price
71         image
72     end
73
74     def side
75         name
76         image
77         size
78         price
79     end
80
81     def main
82         item
83         price
84         image
85     end
86
87     def entree
88         item
89         price
90         image
91     end
92
93     def kid
94         item
95         price
96         image
97     end
98
99     def delete()
100         delete_menu_item
101     end
102     def show_all
103         print everything nicely
104     end
105
106 end

```

```
112
113   def add_amount
114     variable
115   end
116
117   def tax
118     add_tax
119   end
120
121   def give_discount
122     minus_from_total
123   end
124
125   def amount_total
126     total order
127   end
128
129   def split_bill_payment
130     divide_bill
131     get card_number
132     get card_num2
133   end
134
135   def pay_total
136     get card_number
137   end
138   def paid
139     payment_recieved
140     confirm_order
141   end
142
143   def unpaid
144     unpaid order, confirm before continuing
145   end
146
147
148 end
```

12. Code Snippet

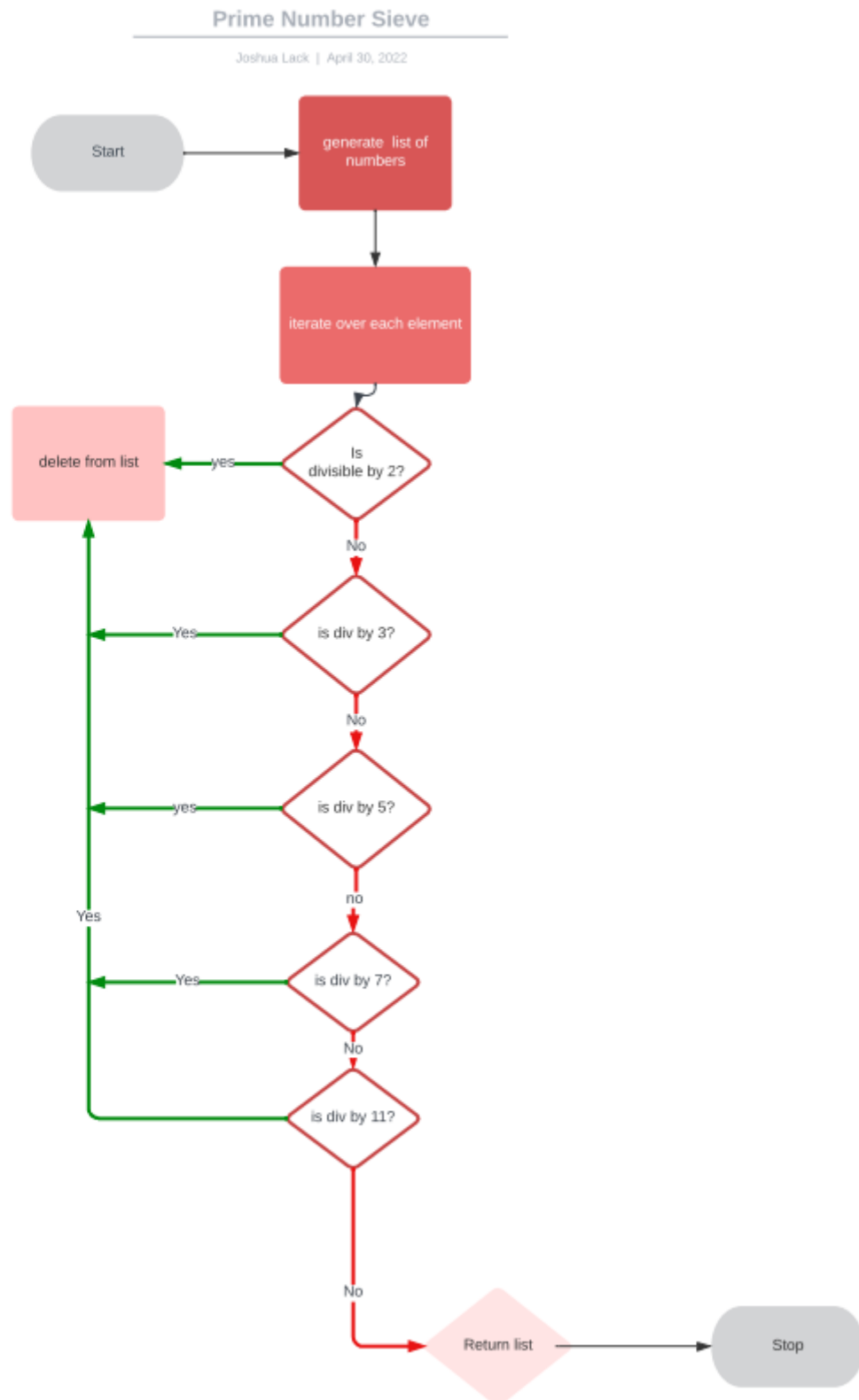
```
1  # temperature is a continuous variable, remove whitespaces
2  celsius = gets.chomp.to_f
3  # when dividing two integers, the result is truncated to an integer unless,
4  # changed to a floating point.
5  # changed the parenthesis to include the fraction on its own,
6  # use the celsius variable as float for the expression
7  fahrenheit = celsius * (9.0 / 5) + 32
8  print 'The result is: '
9  print fahrenheit
10 puts '.'
```

13. Code Snippet Two

```
# solution one - use multiple assignment instead of assigning both elements to arr[i + 1]
arr = [5, 22, 29, 39, 19, 51, 78, 96, 84]
i = 0
while i < arr.size and arr[i] < arr[i + 1]
  i = i + 1 end
puts i
arr[i], arr[i + 1] = arr[i + 1], arr[i]
```

```
1  arr = [5, 22, 29, 39, 19, 51, 78, 96, 84]
2  # finds largest numbers, swaps first two out of order,
3  # disregarding the last elements out of order
4  def bubbles(arr)
5    n = arr.size
6    (n - 2).times do |x|
7      is_swapped = false
8      (n - x - 2).times do |index|
9        if arr[index] > arr[index + 1]
10         arr[index], arr[index + 1] = arr[index + 1], arr[index]
11         is_swapped = true
12       end
13     end
14     break unless is_swapped
15   end
16   arr
17 end
18 p bubbles(arr)
```

14. Prime Sieve



```

1  # Find all primes between 1 and 100
2
3  # numbers 1..100
4
5  # is integer dvisible by 2 3 5 7 11
6  # what is remainder
7  # if 0
8  # then is divishle by itself?

```

```

9   #   is divisible by 1?
10  # when yes then is_prime true
11  # when no then not_prime
12
13  # arr = Array.new(100, i)
14
15  # surely I can do this with a regexp? grep?
16  # p arr
17  # integers = [*2..100]
18  # p integers
19  require 'prime'
20  Prime.each(100) do |prime|
21    p prime
22  end

```

```

1  # I know this uses way more memory than is necessary but I wanted to see if I could do it :)
2
3  multiples = []
4  (1..100).each do |x|
5    multiples << x if (x % 2).zero? && (x / 2) != 1
6    multiples << x if (x % 3).zero? && (x / 3) != 1
7    multiples << x if (x % 5).zero? && (x / 5) != 1
8    multiples << x if (x % 7).zero? && (x / 7) != 1
9    multiples << x if (x % 11).zero? && (x / 11) != 1
10 end
11
12 integers = []
13 (2..100).each do |i|
14   integers << i
15 end
16
17 primes = integers - multiples
18 print primes

```

15. Raining Pseudocode

```
2  case temperature
3  when (0..15)
4    if raining == true
5      print 'It\'s wet and cold'
6    else
7      print 'It\'s not raining but is cold'
8    end
9  when temperature >= 15
10   if raining == false
11     print 'It\'s warm and not raining'
12   else
13     print 'It\'s warm and raining'
14   end
15 end
```

REFERENCES

Cloudflare.com. n.d. What is DNS?. [online] Available at: <https://www.cloudflare.com/en-au/learning/dns/what-is-dns/> [Accessed 22 March 2022].

Cloudflare.com. n.d. What is HTTPS?. [online] Available at: <https://www.cloudflare.com/en-au/learning/ssl/what-is-https/> [Accessed 22 March 2022].

Cloudflare.com. n.d. What is routing?. [online] Available at: <https://www.cloudflare.com/en-gb/learning/network-layer/what-is-routing/> [Accessed 22 March 2022].

Developer.mozilla.org. n.d. An overview of HTTP - HTTP | MDN. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> [Accessed 22 March 2022].

Fox, P., 2020. Khan Academy. [online] Khanacademy.org. Available at: <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:addressing-the-internet/a/ip-v4-v6-addresses?modal=1> [Accessed 22 March 2022].

Fox, P., n.d. Transmission Control Protocol (TCP) (article) | Khan Academy. [online] Khan Academy. Available at: <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/transmission-control-protocol--tcp> [Accessed 22 March 2022].

Kyrnin, J., 2022. What Are Markup Languages?. [online] ThoughtCo. Available at: <https://www.thoughtco.com/what-are-markup-languages-3468655#:~:text=For web design and development,HTML%2C XML%2C and XHTML>. [Accessed 22 March 2022].

McAllister, N., 2012. Twitter survives election after Ruby-to-Java move. [online] Theregister.com. Available at: https://www.theregister.com/2012/11/08/twitter_epic_traffic_saved_by_java/#:~:text=Micro-blogging site Twitter experienced,Java for its backend software. [Accessed 22 March 2022].

Miralles, A., 2014. Mastering data structures in Ruby—Graphs. [online] Medium. Available at: [https://medium.com/amiralles/mastering-data-structures-in-ruby-graphs-caa5892d50b1#:~:text=A graph is a data,this is vertex%2Fnode.\)](https://medium.com/amiralles/mastering-data-structures-in-ruby-graphs-caa5892d50b1#:~:text=A graph is a data,this is vertex%2Fnode.)) [Accessed 22 March 2022].

Neeva, T., 2021. What is HTTP? How It Works and Related Safety Concerns. [online] Neeva. Available at: <https://neeva.com/learn/what-is-http> [Accessed 22 March 2022].

Shalom, N., 2020. What is a Three-Way Handshake? - Definition from Techopedia. [online] Techopedia.com. Available at: <https://www.techopedia.com/definition/10339/three-way-handshake> [Accessed 22 March 2022].

Techterms.com. 2014. Web Browser Definition. [online] Available at: https://techterms.com/definition/web_browser [Accessed 22 March 2022].

University, I., 2022. What is a packet?. [online] Kb.iu.edu. Available at: <https://kb.iu.edu/d/anyq#:~:text=Packets evolved due to the,end can transmit to another>. [Accessed 22 March 2022].