Personal Study Notes

Software Engineer

Post Grad Notes

Victoria, British Columbia, Canada

September 7, 2019

Name: David Li
Student Number: V00818631
Email: davidli012345@gmail.ca

In partial fulfillment of the never ending quest to learn.

Contents

List of Figures

Listings

Chapter 1

Example chapter

1.1 A section

This is some text which is indexed.

1.1.1 A subsection

See fig. 1.1.

SENG475 Example Code Listing

All for the sake of glory city. We see that in **??** that mdframed makes environments look nice

SENG475 Example Code Listing

All for the sake of glory city. We see that in **??** that mdframed makes environments look nice

SENG475 Example Code Listing

All for the sake of glory city. We see that in **??** that mdframed makes environments look nice

Listing 1.1: Cpp Testing

```
#include <iostream>
```

Figure 1.1: A figure with text

Listing 1.2: Golang Program

```
package main

import (
"fmt"
"io/ioutil"
)
```

Listing 1.3: Python Script

```
#!/usr/bin/env python
### Adding prism style highlighting in docs
import os
# os.chdir('../')
import sys
print(os.getcwd())
lwarpFiles = []
dir_list =  os.listdir()
for full_file_name in dir_list:
    base_nameTemp, extensionTemp = os.path.splitext(full_file_name)
    if extensionTemp == '.html': # then .pdf file --> convert to image!
        lwarpFiles.append(full_file_name)

import re
directory = 'docs'
if not os.path.exists(directory):
    os.makedirs(directory)
for outputFile in lwarpFiles:
    try:
        base_nameTemp, extensionTemp = os.path.splitext(outputFile)
        finalOutputName = base_nameTemp + extensionTemp
        htmlFile = open(outputFile ,'r',encoding='utf-8')
        htmlSyntaxFile = open(directory + r'/'␣+finalOutputName,'w',enco
    except OSError:
        print('Cannot␣open␣files,␣probably␣because␣they␣are␣being␣used.␣
        pass
    # include prismCss and prismjs in the final html final, consider mer
    prismCss = r'<link rel="stylesheet" href="prism.css" type="text/css"
␣␣␣␣prismJs␣␣=␣r'<script src="prism.js" type="text/javascript"> </script

␣␣␣␣#␣Replace␣the␣lwarp␣<pre␣class="programlisting"␣>
␣␣␣␣#␣with␣something␣that␣works␣for␣prism
␣␣␣␣lwarpCodeSyn␣=␣r'<pre class="programlisting">'
␣␣␣␣lwarpCodeVerb␣=␣r'<pre class="verbatim">'
␣␣␣␣matlab␣="language-matlab"
␣␣␣␣latex␣=␣"language-latex"
```

```
    python = "language-python"
    cplusplus = "language-cpp"
    json = "language-json"
    bash = "language-bash"
    yaml = "language-yaml"
    golang = "language-go"
    js = "language-javascript"
    # Go through each even entry in replacements and then check if a replacement
    replacementTerms = ["Cpp", cplusplus, "Latex Code", latex, "Python Script", p
        "Bash Script", bash, "Matlab Script", matlab, "Yaml File", yaml, "JSON Output
    prismVerbCodeSyn = r'<pre><code class = "' + latex + r'">'
    prismCodeSyn = r'<pre><code class = "' + cplusplus + r'">'

    # Determines if the next pre tag should be changed
    changeNextPre = False
    #replacementLine = ""
    for line in htmlFile:
        #Include prism.js and prism.css after the title in the html document
        newline = re.sub(r'</title>',r'</title>' + '\n' + prismCss + '\n' + prism

        # CHange if the next pre tag should be changed, if it is not true already
        if changeNextPre == False:
            for i in range (0, int(len(replacementTerms)/2)):
                # match only matches from the beginning of the string. Your code
                pattern = re.compile(replacementTerms[i*2])
                changeNextPre = bool(re.search(pattern,newline))
                #print(r'<p>'+replacementTerms[i*2])
                if changeNextPre:
                    #print("Match Found")
                    replacementLine = r'<pre><code class = "' + replacementTerms
                    #print(replacementTerms)
                    break

        ## get code syntax highlighting
        if changeNextPre == False:
            # Assume matlab is being used
            newline = re.sub(lwarpCodeSyn, prismCodeSyn, newline)
        else:
            templine = re.subn(lwarpCodeSyn, replacementLine, newline)
            newline = templine[0]
            # if a match occurs reset changeNextPre to false
            if templine[1] > 0:
                changeNextPre = False
        ## since a new code tag is introduced it must be closed
        newline = re.sub(r'</pre>',r'</code>' + '\n' + r'</pre>' + '\n', newline)
        # account for new problem of <pre class="verbatim" >
```

```
        #newline = re.sub( lwarpCodeVerb,  prismVerbCodeSyn, newline)
        htmlSyntaxFile.write(str(newline))
        #print(newline)

    htmlFile.close()
    htmlSyntaxFile.close()
print('Script␣is␣Done␣creating␣files')
```

Using references is [1]

1.2 Some math

Inline math: $r = r_0 + vt - \frac{1}{2}at^2$ followed by display math:

$$a^2 + b^2 = c^2 \tag{1.1}$$

Chapter 2

Real Content

QPU Versus GPU: Some Common Characteristics

- It is very rare that a program will run entirely on a QPU. Usually, a program running on a CPU will issue QPU instructions, and later retrieve the results.
- Some tasks are very well suited to the QPU, and others are not.
- The QPU runs on a separate clock from the CPU, and usually has its own dedicated hardware interfaces to external devices (such as optical outputs).
- A typical QPU has its own special RAM, which the CPU cannot efficiently access.
- Here's a list of pertinent facts about what it's like to program a QPU: It is very rare that a program will run entirely on a QPU. Usually, a program running on a CPU will issue QPU instructions, and later retrieve the results. Some tasks are very well suited to the QPU, and others are not. The QPU runs on a separate clock from the CPU, and usually has its own dedicated hardware interfaces to external devices (such as optical outputs). A typical QPU has its own special RAM, which the CPU cannot efficiently access.

Listing 2.1: Javascript Program: Quantum random spy hunter

```
Example 2-4. Quantum random spy hunter
qc.reset(3);
qc.discard();
var a = qint.new(1, 'alice');
var fiber = qint.new(1, 'fiber');
var b = qint.new(1, 'bob');
```

Name	Category	Priority
examin pyalgotrade for stock selling and buying	low	investing
work on web scrapper experiment felgo	webscrap	high
gas station network	high	dapps
explore ipfs solutions such as pinata and textile		
add dash auth to dashboard	finance	high

Table 2.1: **Todo List 2019/8/5**

```
function random_bit(q) {
    q.write(0);
    q.had();
    return q.read();
}

// Generate two random bits
var send_had = random_bit(a);
var send_val = random_bit(a);

// Prepare Alice's qubit
a.write(0);
if (send_val)  // Use a random bit to set the value
    a.not();
if (send_had)  // Use a random bit to apply HAD or not
    a.had();

// Send the qubit!
fiber.exchange(a);

// Activate the spy
var spy_is_present = true;
```

Listing 2.2: Cpp Testing

```
#include <iostream>
```

Listing 2.3: Golang Program

```
package main
```

```
import (
        "fmt"
        "io/ioutil"
)
```

Bibliography

[1] Valentina Porcu. *Python for Data Mining Quick Syntax Reference*. 1st ed. Apress, 2019. ISBN: 1484241126, 978-1484241127. URL: http://gen.lib.rus.ec/book/index.php?md5=719FA57E15A5FAAE9996594FF491F0E1.

[2] Igor Livshin. *Artificial Neural Networks with Java - Tools for Building Neural Network Applications*. Apress, 2019. ISBN: 9781484244203. URL: http://gen.lib.rus.ec/book/index.php?md5=1a97921ae2ebdb2acbce0ef06e29667d.

[3] Mercedes Gimeno-Segovia Eric R. Johnston Nic Harrigan. *Programming Quantum Computers: Essential Algorithms and Code Samples*. 1st ed. O'Reilly Media, 2019. ISBN: 1492039683, 978-1492039686. URL: http://gen.lib.rus.ec/book/index.php?md5=71fa12e84983bfe4b6b39c80879077a4.

[4] Mercedes Gimeno-Segovia Eric R. Johnston Nic Harrigan. *Programming Quantum Computers: Essential Algorithms and Code Samples*. https://oreilly-qc.github.io/. 2019.