

```
1 using System;
2 using System.Collections.ObjectModel;
3 using System.Net;
4 using OpenQA.Selenium;
5 using OpenQA.Selenium.Chrome;
6 using System.Text;
7 using OpenQA.Selenium.Support.UI;
8
9 namespace selenium
10 {
11     class Program
12     {
13         //Yu Wen
14         static void Main(string[] args)
15         {
16             IWebDriver driver = new ChromeDriver();
17
18             //question3internalLinks(driver);
19
20             //question6TradeMeInternalLinks(driver);
21
22         }
23
24         static void question3internalLinks(IWebDriver driver)
25         {
26             //opens website
27             driver.Navigate().GoToUrl("http://automationpractice.com");
28             //find links with tag a
29             List<IWebElement> allLink = driver.FindElements(By.TagName "a").ToList();
30             //List<IWebElement> imgLink = driver.FindElements(By.TagName "img").ToList();
31             //allLink.AddRange(imgLink);
32             String homepage = "http://automationpractice.com";
33
34             for (int i = 0; i < allLink.Count; i++)
35             {
36                 string url = allLink[i].GetAttribute("href");//get links
37
38                 if (url == string.Empty || url == null)
39                 {
40                     //when it is empty (broken)
41                     Console.Write((i + 1) + " ");
42                     Console.WriteLine("Empty :" + allLink[i].Text);
43                 }
44                 else if (url.StartsWith("http://") || url.StartsWith("https://"))
45                 {
46                     if (!url.StartsWith(homepage))
```

```
47         {
48             //external links
49             Console.Write((i + 1) + " ");
50             Console.WriteLine("external : " + url);
51         }
52         else
53         {
54             //valid internal links
55             Console.Write((i + 1) + " ");
56             Console.WriteLine("Valid : " + url);
57         }
58     }
59     else
60     {
61         //for broken links
62         Uri uriResult;
63         bool result = Uri.TryCreate(url, UriKind.Absolute, out uriResult) && (uriResult.Scheme == Uri.UriSchemeHttp || uriResult.Scheme == Uri.UriSchemeHttps);
64         if (result != true)
65         {
66             Console.Write((i + 1) + " ");
67             Console.WriteLine("Broken link : " + url);
68         }
69     }
70 }
71
72 Console.ReadKey();
73 driver.Quit();
74 }
75
76
77 static void question6TradeMeInternalLinks(IWebDriver driver)
78 {
79     //open the webpage
80     driver.Navigate().GoToUrl("https://www.trademe.co.nz/a/");
81
82     //find links with tag a
83     List<IWebElement> allLink = driver.FindElements(By.TagName("a")).ToList();
84
85     //find links with tag img(images)
86     List<IWebElement> imgLink = driver.FindElements(By.TagName("img")).ToList();
87     //allLink.AddRange(imgLink);
88     string homepage = "https://www.trademe.co.nz/";
89
90     for (int i = 0; i < allLink.Count; i++)
91     {
```

```
92         string link = allLink[i].GetAttribute("href");
93
94         if (link == null || link == string.Empty)
95         {
96             //when the url is not abiliabile or empty
97             //string.empty is leading to somewhere in memory, null ↗
98             //is not
99             continue;
100         }
101     else
102     {
103         if (link.StartsWith(homepage))//from internal
104         {
105             //find different links
106             if (link.Contains("property"))
107             {
108                 Console.WriteLine("property link " + link);
109             }
110             else if (link.Contains("service"))
111             {
112                 Console.WriteLine("service link " + link);
113             }
114         }
115     }
116
117     //output
118     Console.ReadKey();
119     driver.Quit();
120 }
121
122 }
123 }
124
```