CS262 Unit 1

24

Gundega

Contents

1 CS262 - Unit 1: Web Page Source

Contents

1.1 1. Building a Web Browser

In this course you will learn the theory and practice of programming languages that will culminate in the construction of a web browser. Your web browser will take HTML and java script as input – the primary languages of the web– and use it to produce an image of the webpage.

You are probably familiar with HTML, which describes the basics of web pages. However, you may be less familiar with JavaScript, which allows you to describe computations so that you can have a lot of power and flashy graphics.

For example, sites use JavaScript to animate tabs, so that when you scroll over them a drop down appears or some floating text appears (http://www.podcast-tuneup.com/). When you look at the source code for pages with these features you will see that they use both HTML and JavaScript.

Course and Project Overview

- Step 1: Start with the web page source, and then break that source into important words.
- Step 2: Next, understand the structure of the words you have found.
- Step 3: Finally, you will look for and find the meaning of the structure.

The goal of this course is to use a browser to structure your learning.

1.1.1 1.1. Quiz: Breaking up Strings

Consider:

```
<b>Hello 1¶
¶
```

One approach to breaking this up would be to use Python’s **string.find** function, which would find the space between the “Hello” and the “1” and then split up the string into everything to the right of the space as well as everything to the left of the space.

Python's **string.find** function is often described as finding a needle in a haystack.

Example:

```
"Mifune Toshiro".find("fun")¶
¶
```

"fun" is the needle you are looking for in "Mifune Toshiro" – the haystack string on the left.

The result you should expect is the string index of the beginning of the fun in "Mifune," which in this case is two. Remember, Python starts counting positions at zero; so that the "M" is in position zero, "i" is in position one, and "f" is in position two.

Do you know who Mifune Toshiro is?

Example:

```
"Hello world".find(" ")¶
5¶
¶
```

Example:

```
"1 + 1=2".find("1", 2)¶
4¶
¶
```

This expression defines a starting position, two. Therefore, you want to find the first occurrence of *1*, starting at position two. Notice that there is a *1* in position zero and position four, however, since we are counting the *1s* after position two, there is just one in position four.
Example:

```
"haystack".find("needle")¶
¶
```

What happens when the needle you are look for does not occur? In this case Python will return negative one (-1)

1.1.2 1.2. Quiz: Breaking Up Strings

What answer would the Python interpreter return, given the following expressions:

```
"Ada Lovelace".find(" ")¶
"Alan Turing".find("n", 4)¶
¶
```

1.2 2. Selecting Substrings

So far, you can find positions (indices) in strings. Now, what you want to do is chop up those strings into substrings. Once you know where the spaces are you can start splitting a sentence up into words. The Python syntax for isolating strings is to use square brackets. The elements within the brackets include, the starting position, followed by a semi-colon, and then the end position.

```
"hello"[1:3]¶
'el'¶
¶
```

This statement says to start at position one and include all of the elements in the string up to but NOT including the element in position three. Therefore, the resulting string is *el*.
You can also leave out one of the number specifiers. In the example below, leaving the position to the right of the semi-colon empty means to go as far as possible – to the end of the string.

```
'hello'[1: ]¶
'ello'¶
¶
```

Now that you know how to find strings and chop them up, see if you can combine them together to write a Python procedure.

1.2.1 2.1. Quiz: Selecting Substrings

Let *p* and *q* be strings containing two words separated by a space. Example: *"bell hooks"*, *"grace hopper"*, *"alonzo church"*. Write a procedure called **myfirst_yoursecond(p,q)** that returns *True* if the first word in *p* equals the second word in *q*.

Example:

```
myfirst_yoursecond("bell hooks", "curer bell”)¶
True¶
¶
```

1.2.2 2.2. Split

Splitting words by spaces in computer science is such a common task that Python has a built in function to do just that.

- string.split( )

For example:

```
"Jane Eyre".split( )¶
["Jane", "Eyre"]¶
¶
```

1.2.3 2.3. Quiz: Split

What is the number of elements in the list returned by split( ) expression in each of the following.

```
"Python is fun".split( )¶
"July-August 1842".split( )¶
"6*9==42".split( )¶
¶
```

1.3 3. Regular Expressions

For strings with elements such as hyphens and operational symbols, like the second two above, you may be inclined to want to split them up too. Therefore, you need more control over splitting strings than Python’s built in string function can offer.

Regular expressions are tools that give you more control over splitting up strings!

Suppose you want to find all the numbers in a string. You could make ten different calls to **s.find**:

```
s.find(“1”)¶
s.find(“2”)¶
s.find(“3”)¶
…¶
¶
```

This gets really tedious, really fast. Regular expressions are a better way to do this.

The term "regular" has special meaning in mathematics and computer science. For now it means, simple strings. And an expression is concise notation.

The following equations correspond to some possible value for x:

- x = sqrt(4)   x =2 or x =-2
- 5<x<9   x =6 or 7 or 8

These mathematical equations are concise notations for potentially large sets of values. Especially an equation such as:

- 50< x < 90

Regular expressions are going to be very concise and will let you describe a large number of simple strings.

Regular expressions:

1. [1-3], matches or denotes these three strings: “1”, “2”, “3”

The basic idea is that some symbol on the left and some symbol on the right, so that the regular expression matches each one of those and everything in between

1. [4-8]   “4”, “5”, “6”, “7”, “8”
2. [a-b]   “a” “b”

Regular expressions are very popular. They are very useful online and in computing in general. Credit cards, phone numbers, addresses and emails are all handled by regular expressions on websites you use everyday. For example, the form that you fill out when you apply for a U.S. Passport relies upon regular expressions. Regular expressions are very common when you want to enter structured data. Within the form, notice that different items; your birthday, your birthplace, your social security number, your email address, require different ways of writing this information. Regular expressions allows you to make sense of this type of data and process it when you see it on webpages.

1.3.1 3.1. Quiz: Single Digits

1.4 4. Import

1.4.1 4.1. Quiz: Find All

1.4.2 4.2. Quiz: Concatenation

1.4.3 4.3. Quiz: One or More

1.5 5. Finite State Machines

1.5.1 5.1. Quiz: Accepting States

1.5.2 5.2. Quiz: FSM Evolution

1.6 6. Disjunction

1.6.1 6.1. Quiz: Disjunction in FSM

1.6.2 6.2. Quiz: Disjunction Construction

1.7 7. Options

1.8 8. Escape Sequences

1.8.1 8.1. Quiz: Hyphenation

1.9 9. Tricky

1.9.1 9.1. Quiz: Re-Challenges

1.10 10. Quoted Strings

1.11 11. Structure

1.11.1 11.1. Quiz: Escaping the Escape

1.12 12. Representing a FSM

1.12.1 12.1. Quiz: FSM Simulator

1.12.2 12.2. Quiz: FSM Interpretation

1.12.3 12.3. More FSM Encoding

1.12.4 12.4. mis MSF

1.13 13. Epsilon and Ambiguity

1.13.1 13.1. Quiz: Phone It In

1.13.2 13.2. Quiz: Inverting the Problem

1.14 14. Nondeterminism

1.14.1 14.1. Quiz: Nondet to det

1.15 15. Save the World

1.15.1 15.1. Keywords

Strings - sequence of characters **Regular Expressions** - concise notation for specifying sets of strings. More flexible than fixed string matching. **Finite State Machine (FSM)** - pictorial equivalent to regular expression

1.16 16. World of Westley's Whimsical References

1.16.1 16.1. From Unit 1-4 : Toshiro Mifune, Hello World, Ada Lovelace, Alan Turing

Toshiro Mifune was a Japanese actor in films such as **Seven Samurai** and **Rashomon**

Hello World is a traditional first program or first example text.

Ada Lovelace is widely regarded as the first computer programmer.

Alan Turing was highly influential in the development of computer science and in code breaking during World War II.

1.16.2 16.2. From Unit 1-5 : Grace Hopper, Currer Bell, bell hooks

Grace Hopper was a computer scientist who developed the first compiler — or translator — for a computer programming language.

Currer Bell (note spelling — the video has a typo) was a pen name used by Charlotte Brontë when she wrote Jane Eyre.

bell hooks (intentionally uncapitalized) is a pen name used by Gloria Jean Watkins, an author and social activist.

1.16.3 16.3. From Unit 1-6 : What do you get if you multiply six by nine, Douglas Adam's The Hitchhiker's Guide to the Galaxy

What do you get if you multiply six by nine? was a fairly important question in **The Hitchhiker's Guide to the Galaxy**, a science fiction comedy by Douglas Adams.

1.16.4 16.4. From Unit 1-8 : Isak Dinesen, Out of Africa

Isak Dinesen was a pen name for Karen Dinesen, the author of **Out of Africa**.

1.16.5 16.5. From Unit 1-9 : Barbara Liskov, Turing Award

Barbara Liskov is a computer scientist and recipient of the Turing Award.

1.16.6 16.6. From Unit 1-10 : Mir Taqi Mir, Khwaja Mir Dard, Urdu Poets

Mir Taqi Mir and **Khwaja Mir Dard** were 18th century Urdu poets.

1.16.7 16.7. From Unit 1-12 : Peru's Independence from Spain

Peru declared independence from **Spain** on July 28, 1821.

1.16.8 16.8. From Unit 1-14 : Murasaki Shikibu's The Tale of Genji, American Independence

Murasaki Shikibu wrote **The Tale of Genji**, often called the first historical or psychological novel.

Thirteen colonies declared independence from the British Empire in **1776** and became the United States of America.

1.16.9 16.9. From Unit 1-16 : Brave New World

Aldous Huxley wrote **Brave New World**, a popular science fiction novel.

1.16.10 16.10. From Unit 1-18 : Goethe, Faust

Johann von Goethe wrote **Faust**.

1.16.11 16.11. From Unit 1-19 : Václav Havel

Václav Havel was a Czech politician and author.

1.16.12 16.12. From Unit 1-21 : Greek Letter Epsilon, Rabindranath Tagore, Where the Mind is Without Fear

The lowercase Greek letter **Epsilon** ( ) is often used to mean "the empty string" (when talking about languages) or "a very small number" (in some branches of mathematics).

Rabindranath Tagore is a Bengali poet and Nobel Prize laureate who wrote the oft-quoted **Where the mind is without fear**.

1.16.13 16.13. From Unit 1-22: Pride and Prejudice, Jane Austen

Pride and Prejudice is an 1813 novel written by **Jane Austen**.

1.16.14 16.14. From Unit 1-27 : Do-Re-Mi, The Virtue of Selfishness, MIDI

Do-Re-Mi is a song from the musical **The Sound of Music**.

The Virtue of Selfishness is a 1964 collection of philosophy essays.

MIDI, the Musical Instrument Digital Interface, is a standard for digital music.

1.16.15 16.15. From Unit 1-28 : The Beatles' Hello, Goodbye, Turtles All The Way Down, Don Knuth's The Complexity of Songs, Ninety-Nine Bottles of Beer

Hello, Goodbye is a 1967 song by **The Beatles**.

Turtles all the way down refers to problems of infinite regression, such as when something is defined in terms of itself.

Don Knuth, a Turing-award winning computer scientist, wrote **The Complexity of Songs**.

Ninety-nine bottles of beer is an American children's song with repetitive lyrics.

1.16.16 16.16. From Unit 1-29 : Back to the Future, Romeo & Juliet, Taj Mahal

In the movie **Back to the Future**, traveling at 88 miles per hour is related to traveling back in time.

"What's in a name? That which we call a rose / By any other name would smell as sweet." is spoken by Juliet in Shakespeare's play **Romeo and Juliet** (II,ii,1-2).

The **Taj Mahal** is a **UNESCO World Heritage Site** in India.

1.16.17 16.17. From Unit 1-35 : History of the Telephone

The **history of the telephone** is actually fairly interesting.

1.16.18 16.18. From Unit 1-37 : Newtonian Mechanics, Free Will & Determinism, Subjective Experience

Newtonian mechanics is a reasonable description for a **billiard ball** universe (at least, for objects that are neither **too small** nor **too fast**).

Free will and **determinism** are the subjects of much debate in philosophy.

For an interesting read on subjective experience and consciousness, you may enjoy **Thomas Nagel's** paper **What is it like to be a bat?**.

1.17 17. End