

Today's Agenda

- Unit 7 – Intro to programming – Part3
 - New:
 - User, User-Agent, Servers
 - Client vs. server side scripting
 - Concatenating numerical and string data in print statements
- Bonus exercise(s) + marking explained
- If time permits, midterm solutions - Review #3

CMPT 165

Unit 7 – Intro to Programming Part 3

July 20th, 2015

Summary of key concepts & terms

Fundamentals

Developer

Interface

GUI

Shell

Program

Statements

Client/server

Fetching a resource

Dynamic HTML

Programming essentials

Variables

Data Types

Numeric

Strings

Assignment (shorthand)

Operations/Operator

Arithmetic

Concatenation

Overloaded symbols

Functions

Data

Input/Output (I/O)

Process

Misc.

- “wildcard” (*)

Statements

...An instruction to IDLE

Execute the statement when you
press ENTER

```
>>> var1=4  
>>> var2=var1*2  
8
```

Output
(blue text)

A Python Shell
(green background)

S.G.: statements that are arithmetic in nature aka numeric expressions

- Can store these statements to form a program; e.g.

myfirstprogram.py

```
var1=4;  
var2=var1*2;
```

Text editor of IDLE
(peach colour)

- Statements are executed in order provided
- To **execute** all statements, press F5 key to run the saved program

Summary of key concepts & terms

Fundamentals

Developer
Interface
GUI
Shell

Program
Statements

Client/server
Fetching a resource
Dynamic HTML

Programming essentials

Variables
Data Types
 Numeric
 Strings
Assignment (shorthand)

Operations/Operator
 Arithmetic
 Concatenation
 Overloaded symbols

Functions

Data
Input/Output (I/O)
Process

Misc.

- “wildcard” (*)

Overloaded symbols

Example type-dependent operations:

	Numeric	Strings
<code>var1*=m</code>	Multiply <code>var1</code> by <code>m</code> and store result back in <code>var1</code>	Replicate itself <code>m</code> times
<code>var1+var2</code>	Addition e.g. <code>var1=1</code> <code>var2=1</code> <code>var1+var2</code> gives 2	Concatenation e.g. <code>var1='hello '</code> <code>var2='world'</code> <code>var1+var2</code> gives 'hello world'

Practice #1

```
>>> var1=4
>>> var2=6**2+var1
?
>>> x=var1*var2
?
>>> x%=8
?
>>> z=x+var1*2
?
>>> z//=2
?
>>> y=z%3 + 100
?
```

```
A=B=C=3;
A=A*B;
```

```
print A;
print B*A;
print 'B*A' ;
```

```
greeting='Hello '
greeting2=3;
```

```
print greeting2;
print greeting+ 'world'
```

Q1) what is value of **y**?

Q2) what is output of above program?

Practice #1

```
>>> var1=4
>>> var2=6**2+var1
?
>>> x=var1*var2
?
>>> x%=8
?
>>> z=x+var1*2
?
>>> z//=2
?
>>> y=z%3 + 100
?
```

```
>>> var1=4
>>> var2=6**2+var1
40
>>> x=var1*var2
160
>>> x%=8
0
>>> z=x+var1*2
8
>>> z//=2
4
>>> y=z%3 + 100
101
```

Make sure you understand this!
shorthand for **x=x%8**

Q1) what is value of **y**?

Practice #1

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.3 (default, Apr 10 2012, 23:24:47)
[MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" f
or more information.
>>> ===== RESTART =====
>>>
9
27
B*A
3
Hello world
>>> |
Ln: 10 Col: 4
```

```
greeting='Hello '
greeting*=3;
print greeting;
```

```
A=B=C=3;
A=A*B;
```

```
print A;
print B*A;
print 'B*A' ;
```

```
greeting='Hello '
greeting2=3;
```

```
print greeting2;
print greeting+ 'world'
```

Q2) what is output of above program?

Q3) what about code on the left?

NEW: How to print integers & strings together?

Here's the syntax for this:

```
print a_number, a_string
```

```
>>> a_number=16
>>> a_string="Dear visitor, you have won $"
>>> print a_string, a_number
Dear visitor, you have won $ 16

>>> another_string=". Goodbye!"
>>> print a_string, a_number, another_string
Dear visitor, you have won $ 16 . Goodbye!
```

Summary of key concepts & terms

Fundamentals

Developer
Interface
GUI
Shell

Program
Statements

Client/server
Fetching a resource
Dynamic HTML

Programming essentials

Variables
Data Types
 Numeric
 Strings
Assignment (shorthand)

Operations/Operator
 Arithmetic
 Concatenation
 Overloaded symbols

Functions

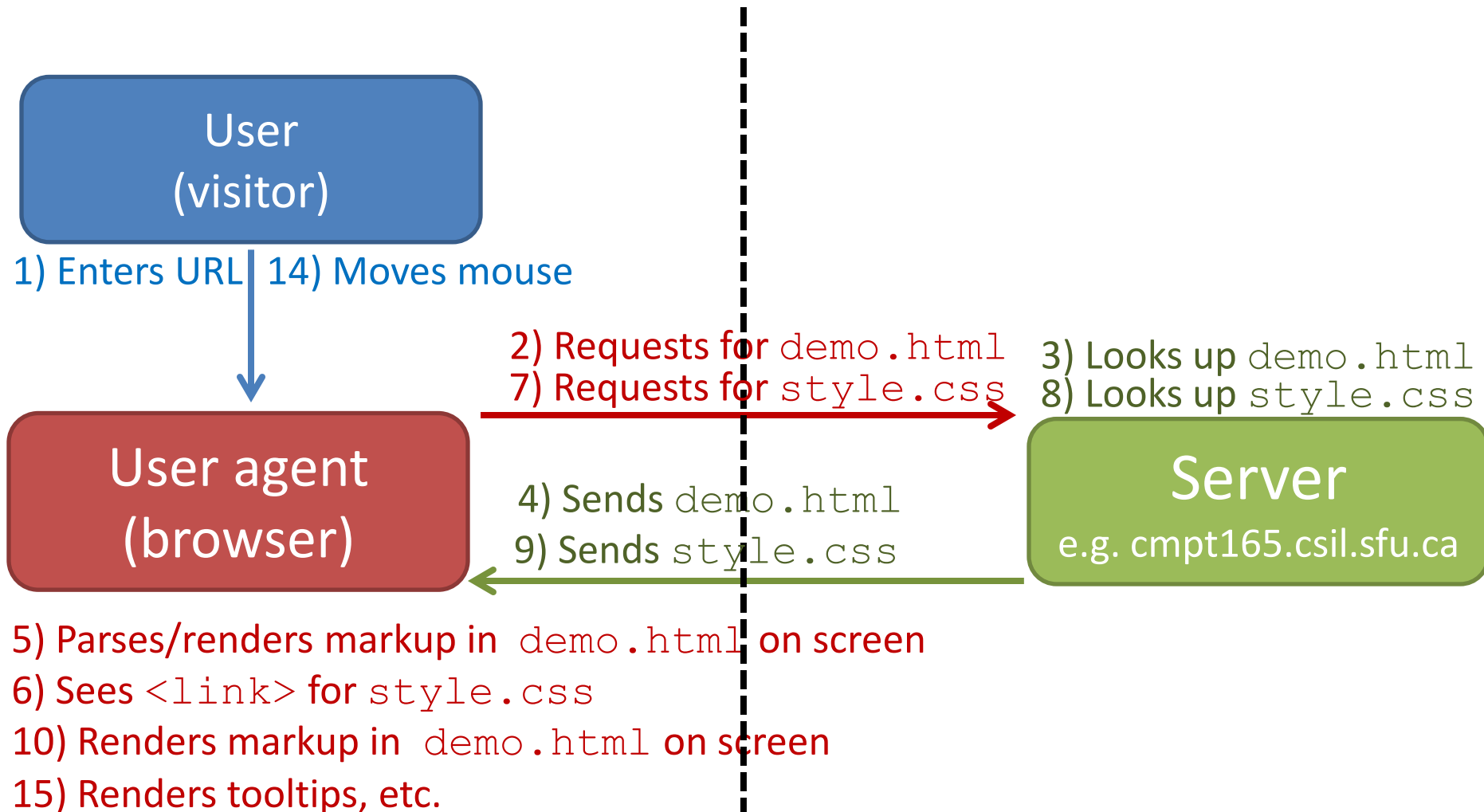
Data
Input/Output (I/O)
Process

Misc.

- “wildcard” (*)

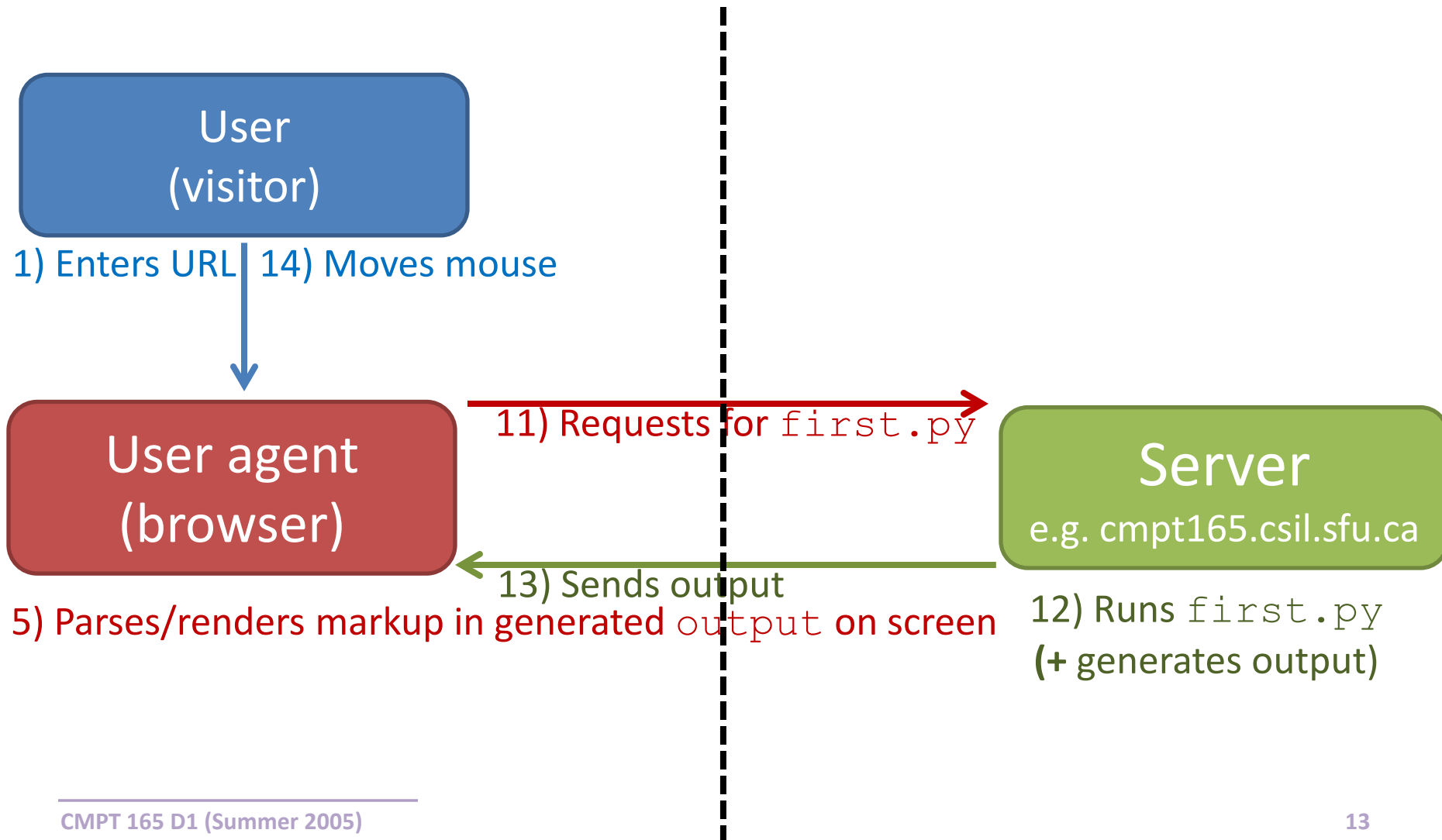
User, User-agent, Server: revisited

Enter URL address bar: `http://cmpt165.csil.sfu.ca/~lisat/demo.html`



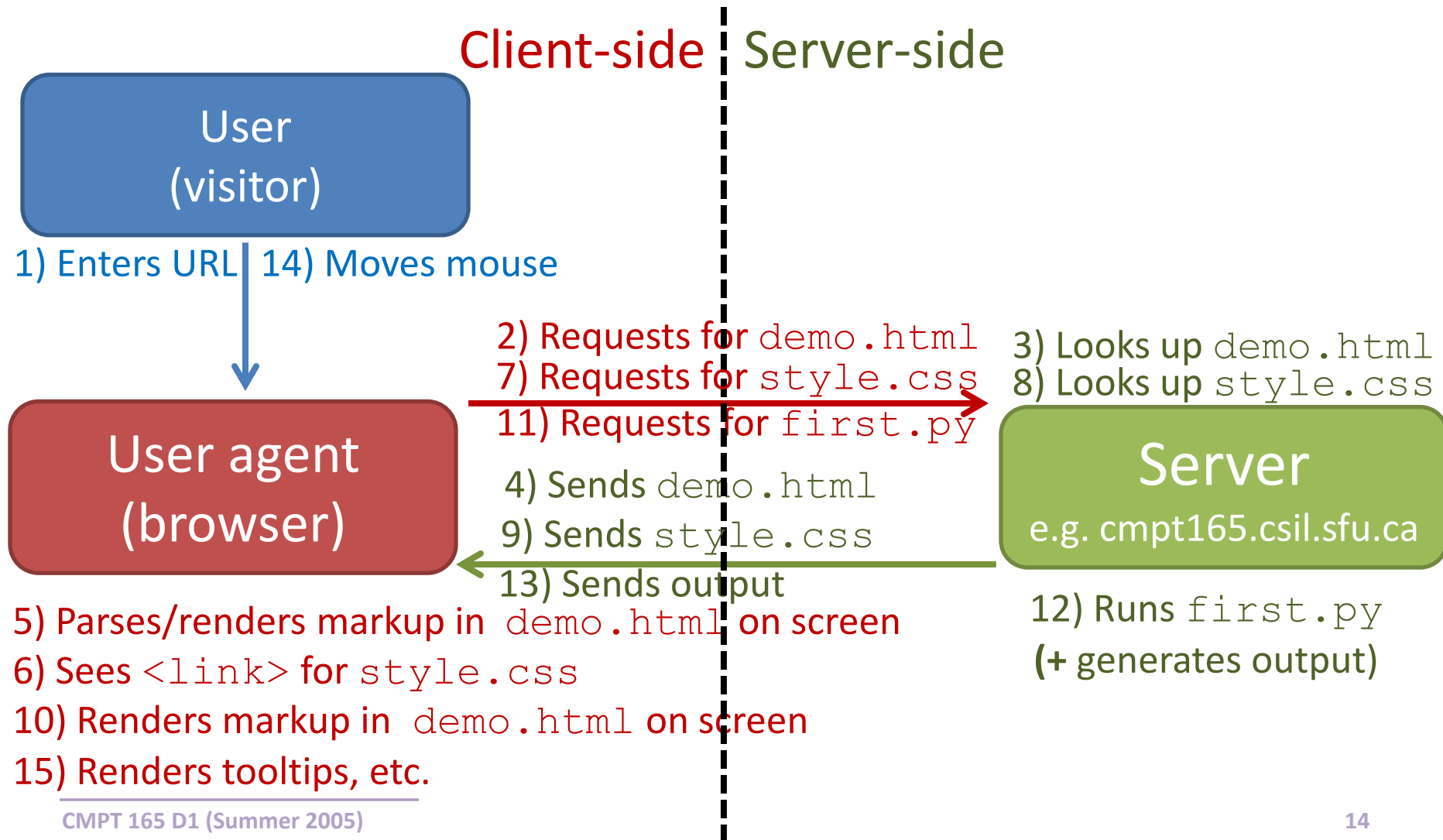
User, User-agent, Server: revisited

Enter URL address bar: ~~http://cmpt165.csil.sfu.ca/~lisat/demo.html~~
http://cmpt165.csil.sfu.ca/~lisat/first.py



User, User-agent, Server: revisited

Enter URL address bar: ~~http://cmpt165.csil.sfu.ca/~lisat/demo.html~~
 http://cmpt165.csil.sfu.ca/~lisat/first.py



Bonus Exercise 1

CMPT 165 - Bonus Exercise 1

2. Modify `first.py` so it prints this instead:

```
I am a Python program...  
...helping my author to finish a lab exercise for "CMPT 165".
```

-0.2 if you...

- Did not use print statements
- Did not write them as numeric expressions, e.g.

Wrong: `print "9929 to the power of 120 "`

Correct: `print 9929**120`

ii. 1650 multiplied by 299

vi. Modulus of 19111 divided by 112

vii. Integer division of 19111 divided by 110

viii. 9929 to the power of 120

4. Save your work (the 6 expressions in the above) in `first.py`.

5. Upload this file to your web space on the course web server and enter its URL to [CourSys](#).
The URL should be:

```
http://cmpt165.csil.sfu.ca/~userid/first.py
```

Don't worry about the output you see on your web browser; we'll examine that in class on Friday.

That's it for this bonus exercise; easy credits!? Life *is* good.

Bonus Exercise 2

cmpt165.csil.sfu.ca/~[redacted]bonus/my_first_dyn_html.py

One of the HTTP headers your script outputted doesn't look right. An HTTP header must look like `name: value`. Did you perhaps forget the blank line between the headers and contents?. You can't have anything on this line, not even a space.

This was the output:

```
<html>
<head>
<title>My first web script</title>
<style> table{font-size:12pt; font-family:Georgia,serif;}
</head>
<body>
<h1>The first web script of [redacted]</h1>
<p>This is a dynamically generated web page.</p>
</body>
</html>
```

☹ Incorrect!

cmpt165.csil.sfu.ca/~[redacted]bonus/my_first_dyn_html.py

The first web script of [redacted]

This is a dynamically generated web page.

How to make different coffees:

Coffee	Amount of milk to add
Latte	

☺ Correct!

MIME type

- How to resolve this?
- Specify MIME type in the output of your Python scripts
- Do so by adding this print statement:

```
print "Content-type: text/html"  
print
```

Prints blank line.
This line is required!

```
print 'I am a Python program...'  
print '...helping my author to finish a lab exercise  
for "CMPT 165".'  
  
print 5012*2988912
```

Dynamic markup from Python scripts

Example similar to one in Fig. 7.2 of Study Guide

```
str="Content-type: text/html"
print str
print

markup("<html><head>"
markup+("<title>Python did this</title>"
markup+("</head><body>"
markup+("<p>Here I am</p>"
markup+("</body></html>"

print markup
```

Summary of key concepts & terms

Fundamentals

Developer
Interface
GUI
Shell

Program
Statements

Client/server
Fetching a resource
Dynamic HTML

Programming essentials

Variables
Data Types
 Numeric
 Strings
Assignment (shorthand)

Operations/Operator
 Arithmetic
 Concatenation
 Overloaded symbols

Functions

Data
Input/Output (I/O)
Process

Misc.

- “wildcard” (*)

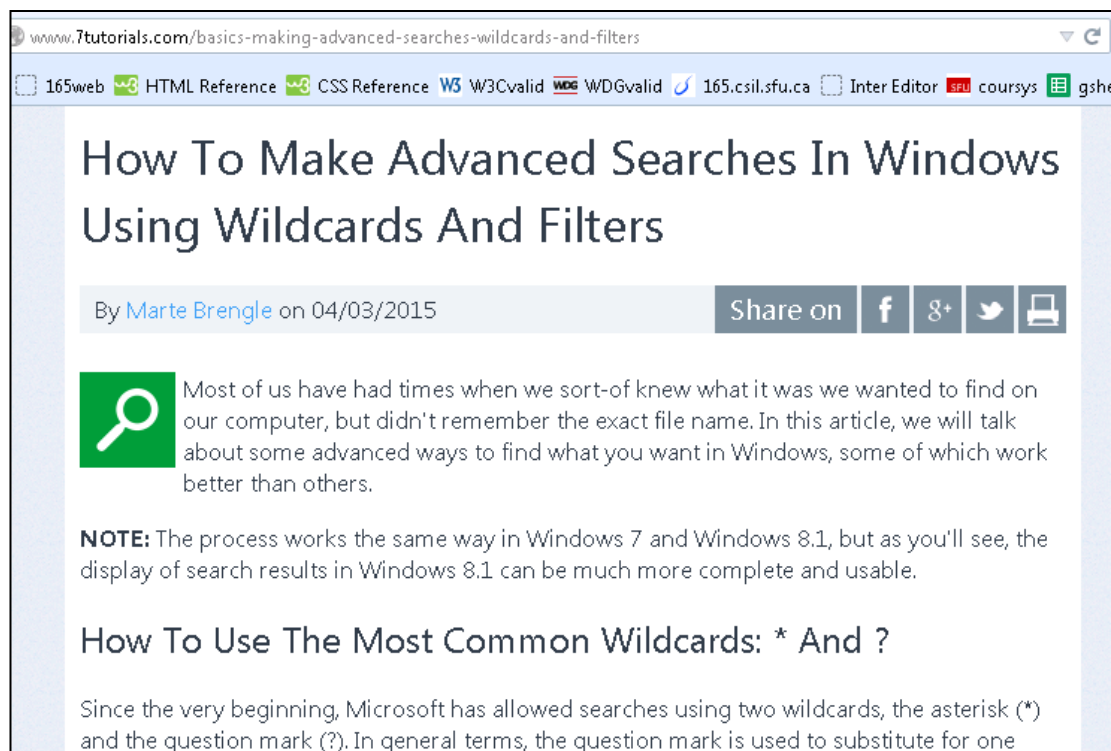
Wildcard reviewed

`*.txt`

`lectures/*.pdf`

`% ls ~/lisat/*.pdf`

Try issuing something like this in a terminal (e.g. Unix via SSH)



<http://www.7tutorials.com/basics-making-advanced-searches-wildcards-and-filters>

Static vs. dynamic resources

Q: What differentiates between these 2 groups of files?

Static resources

* .html

* .htm

* .pdf

* .svg

* .jpg

* .mp3

Files already
saved on
server's hard
disk

Extensions of web scripts

* .py

* .php

* .js

(Javascript)

* .asp

(Active Server Pages)

Scripts that will
generate output
on-the-fly (upon
incoming
requests)

Summary of key concepts & terms

Fundamentals

Developer
Interface
GUI
Shell

Program
Statements

Client/server
Fetching a resource
Dynamic HTML

Programming essentials

Variables
Data Types
 Numeric
 Strings
Assignment (shorthand)

Operations/Operator
 Arithmetic
 Concatenation
 Overloaded symbols

Functions

Data
Input/Output (I/O)
Process

Misc.

- “wildcard” (*)