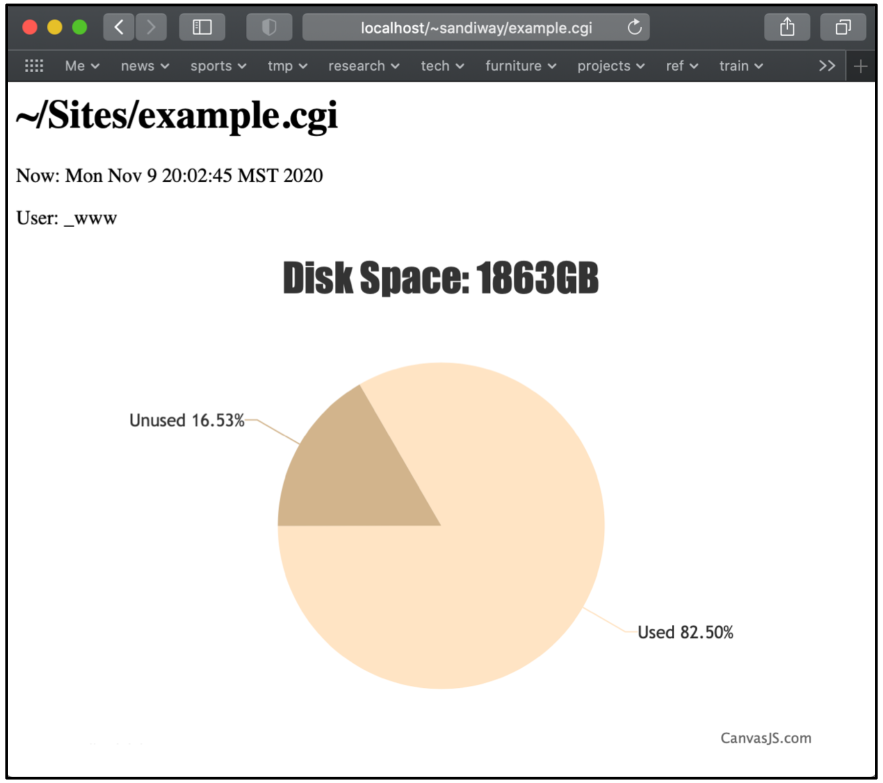
LING 408/508: Computational Techniques for Linguists

Lecture 24

Last Time

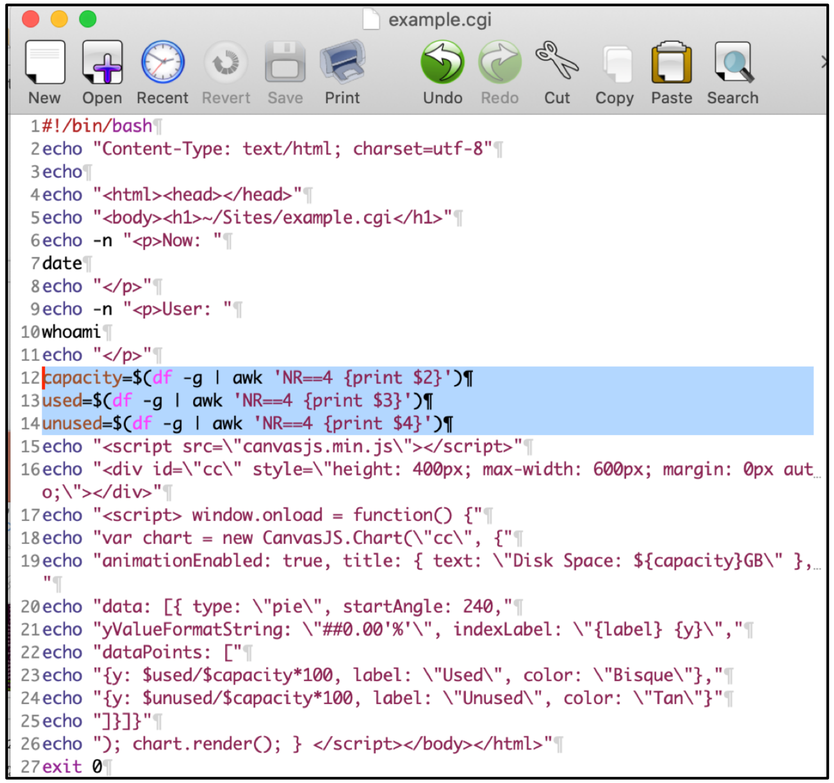

localhost/~sandiway/example.cgi
Me news sports tmp research tech furniture projects ref train
Unused 16.53%
Used 82.50%
CanvasJS.com

~/Sites/example.cgi

Now: Mon Nov 9 20:02:45 MST 2020

User: _www

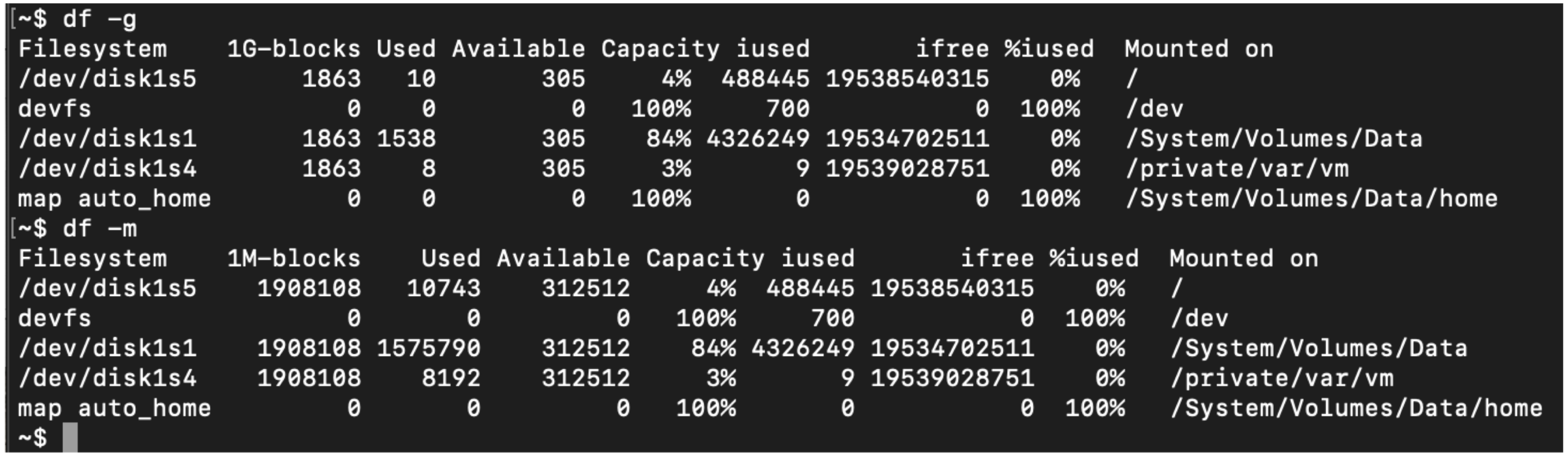
Disk Space: 1863GB


example.cgi
New Open Recent Revert Save Print Undo Redo Cut Copy Paste Search

```
#!/bin/bash
echo "Content-Type: text/html; charset=utf-8"
echo
echo "<html><head></head>"
echo "<body><h1>~/Sites/example.cgi</h1>"
echo -n "<p>Now: "
date
echo "</p>"
echo -n "<p>User: "
whoami
echo "</p>"
capacity=$(df -g | awk 'NR==4 {print $2}')
used=$(df -g | awk 'NR==4 {print $3}')
unused=$(df -g | awk 'NR==4 {print $4}')
echo "<script src=\"canvasjs.min.js\"></script>"
echo "<div id=\"cc\" style=\"height: 400px; max-width: 600px; margin: 0px auto;\"></div>"
echo "<script> window.onload = function() {"
echo "var chart = new CanvasJS.Chart(\"cc\", {"
echo "animationEnabled: true, title: { text: \"Disk Space: ${capacity}GB\" },"
echo "data: [{ type: \"pie\", startAngle: 240,"
echo "yValueFormatString: \"##0.00'%'\", indexLabel: \"{label} {y}\","
echo "dataPoints: ["
echo "{y: $used/$capacity*100, label: \"Used\", color: \"Bisque\"},"
echo "{y: $unused/$capacity*100, label: \"Unused\", color: \"Tan\"}"
echo "]}]"
echo "); chart.render(); } </script></body></html>"
exit 0
```

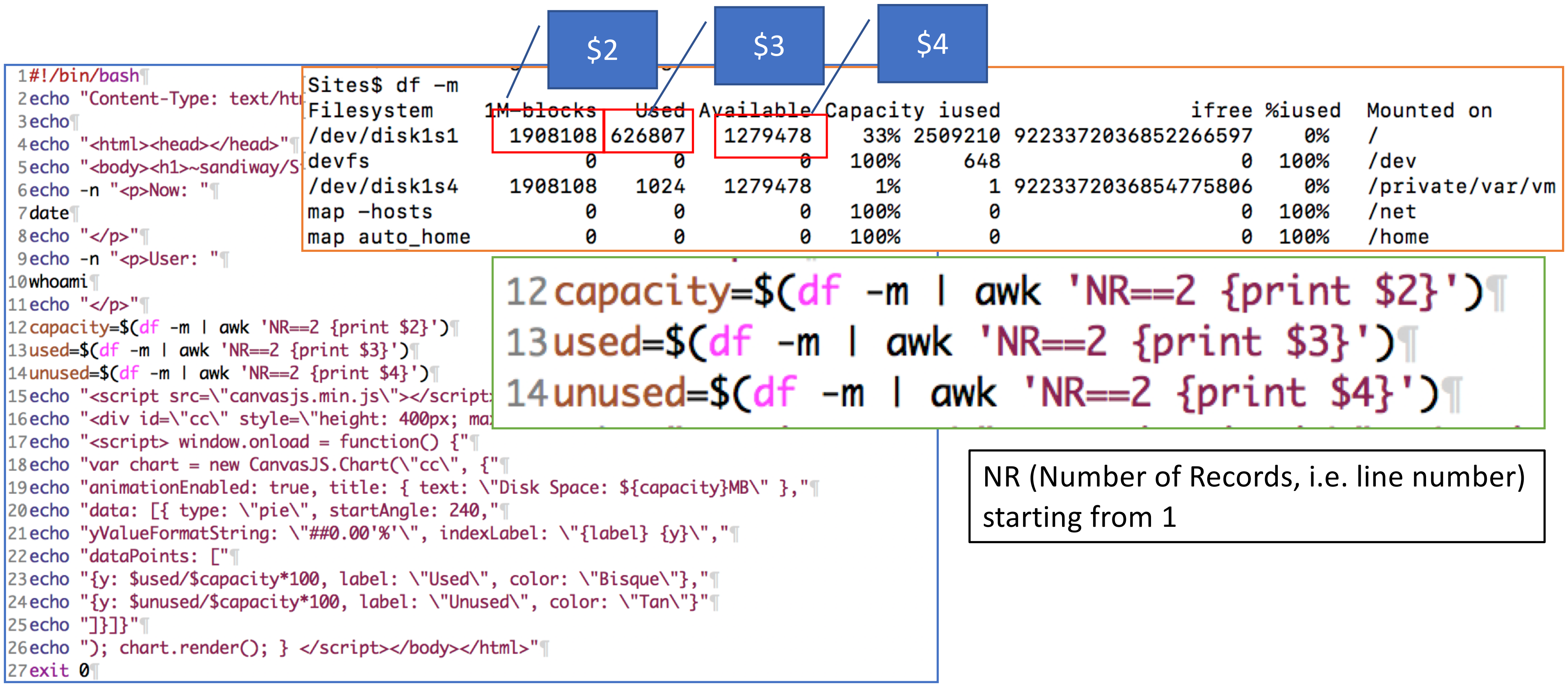
df command

- has various options: –m (megabytes) –g (gigabytes) –H ("Human-readable" output).
  Use unit suffixes: Byte, Kilobyte, Megabyte, Gigabyte, Terabyte and Petabyte
  in order to reduce the number of digits to three or less using base 10 for sizes.



```
[~$ df -g
Filesystem    1G-blocks Used Available Capacity iused       ifree %iused  Mounted on
/dev/disk1s5       1863   10       305     4%  488445 19538540315    0%   /
devfs                 0    0         0   100%     700           0  100%   /dev
/dev/disk1s1       1863 1538       305    84% 4326249 19534702511    0%   /System/Volumes/Data
/dev/disk1s4       1863    8       305     3%       9 19539028751    0%   /private/var/vm
map auto_home         0    0         0   100%       0           0  100%   /System/Volumes/Data/home
[~$ df -m
Filesystem    1M-blocks    Used Available Capacity iused       ifree %iused  Mounted on
/dev/disk1s5    1908108   10743    312512     4%  488445 19538540315    0%   /
devfs                 0       0         0   100%     700           0  100%   /dev
/dev/disk1s1    1908108 1575790    312512    84% 4326249 19534702511    0%   /System/Volumes/Data
/dev/disk1s4    1908108    8192    312512     3%       9 19539028751    0%   /private/var/vm
map auto_home         0       0         0   100%       0           0  100%   /System/Volumes/Data/home
~$ 
```

Example: `example.cgi`



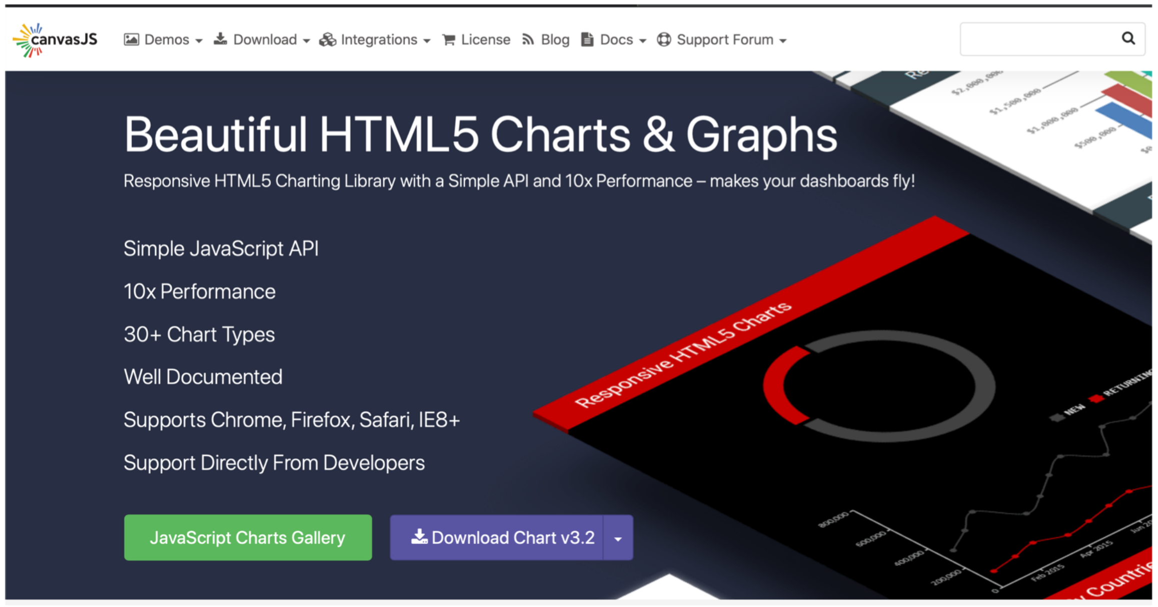
```
#!/bin/bash
echo "Content-Type: text/ht
echo
echo "<html><head></head>"
echo "<body><h1>~sandiway/S
echo -n "<p>Now: "
date
echo "</p>"
echo -n "<p>User: "
whoami
echo "</p>"
capacity=$(df -m | awk 'NR==2 {print $2}')
used=$(df -m | awk 'NR==2 {print $3}')
unused=$(df -m | awk 'NR==2 {print $4}')
echo "<script src=\"canvasjs.min.js\"></script
echo "<div id=\"cc\" style=\"height: 400px; ma
echo "<script> window.onload = function() {"
echo "var chart = new CanvasJS.Chart(\"cc\", {"
echo "animationEnabled: true, title: { text: \"Disk Space: ${capacity}MB\" },"
echo "data: [{ type: \"pie\", startAngle: 240,"
echo "yValueFormatString: \"##0.00'%'\", indexLabel: \"{label} {y}\","
echo "dataPoints: ["
echo "{y: $used/$capacity*100, label: \"Used\", color: \"Bisque\"},"
echo "{y: $unused/$capacity*100, label: \"Unused\", color: \"Tan\"}"
echo "]}]}"
echo "); chart.render(); } </script></body></html>"
exit 0
```

```
Sites$ df -m
Filesystem     1M-blocks   Used Available Capacity  iused               ifree %iused  Mounted on
/dev/disk1s1     1908108 626807   1279478    33% 2509210 9223372036852266597    0%   /
devfs                  0      0         0   100%     648                   0  100%   /dev
/dev/disk1s4     1908108   1024   1279478     1%       1 9223372036854775806    0%   /private/var/vm
map -hosts             0      0         0   100%       0                   0  100%   /net
map auto_home          0      0         0   100%       0                   0  100%   /home
```

```
12 capacity=$(df -m | awk 'NR==2 {print $2}')
13 used=$(df -m | awk 'NR==2 {print $3}')
14 unused=$(df -m | awk 'NR==2 {print $4}')
```

NR (Number of Records, i.e. line number) starting from 1

Example: `example.cgi`



Beautiful HTML5 Charts & Graphs

Responsive HTML5 Charting Library with a Simple API and 10x Performance – makes your dashboards fly!

Simple JavaScript API

10x Performance

30+ Chart Types

Well Documented

Supports Chrome, Firefox, Safari, IE8+

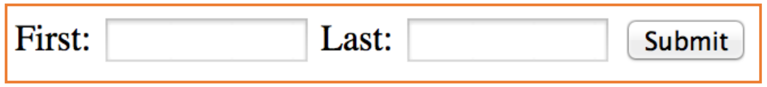
Support Directly From Developers

- Other free toolkits also available
  - Used the free(?) `canvasjs.min.js` library from www.canvasjs.com
  - They have source code for many examples of javascript charts.
  - Wondering what happens after 30 days…

Today's Topics

- Sending form data to the webserver using:
  1. `GET` method
  2. `POST` method
- There are also (many) other methods to communicate information depending on the kind of webserver we run:
  - e.g. Apache Tomcat (for Java)
  - e.g. WebSocket interface (for bidirectional data passing)
  - etc.

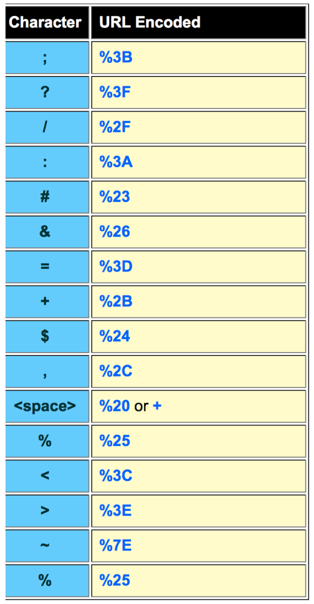
Sending information using **GET**



- HTML form:

```
<form action="http://localhost/cgi-bin/get.cgi" method="GET">
First: <input type="text" name="first" size=12>
Last:  <input type="text" name="last" size=12>
<input type="submit">
</form>
```

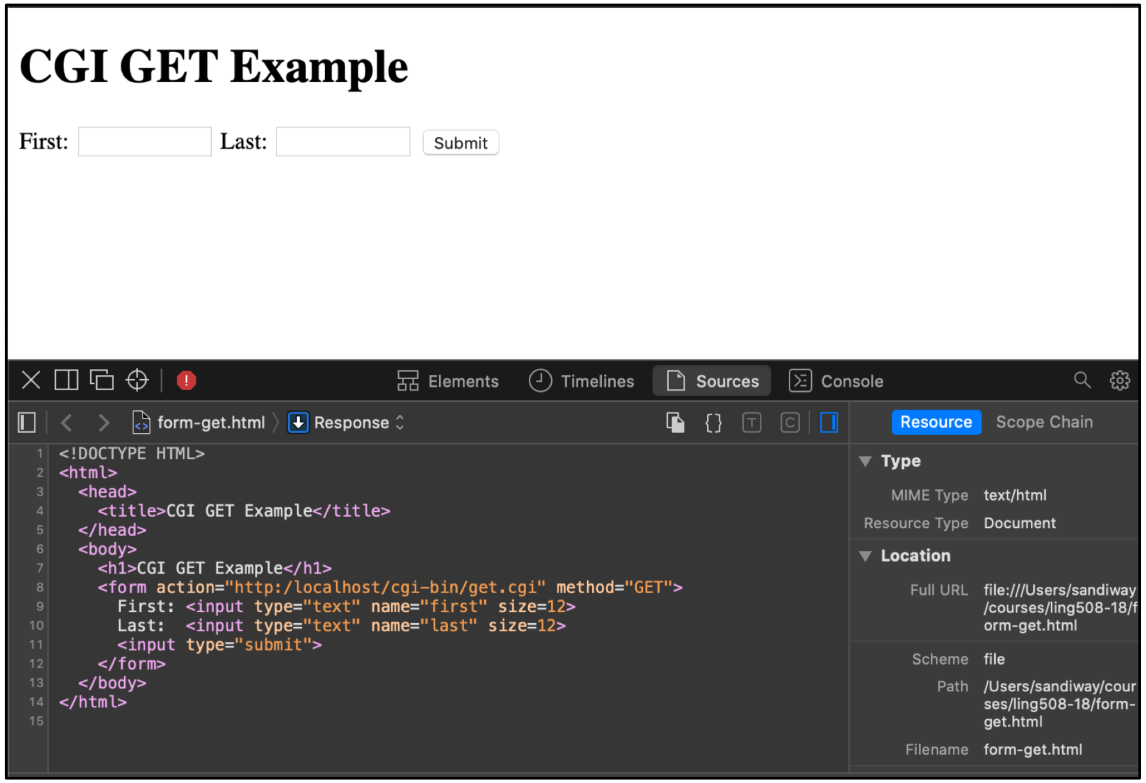
```
http://localhost/cgi-bin/get.cgi?first=Sandiway&last=Fong
```



Character	URL Encoded
;	%3B
?	%3F
/	%2F
:	%3A
#	%23
&	%26
=	%3D
+	%2B
$	%24
,	%2C
<space>	%20 or +
%	%25
<	%3C
>	%3E
~	%7E
%	%25

- Information encoded using alphanumeric characters: why?
- URLs are restricted to alphanumeric characters only
- **bash** accesses the URL-encoded string via the environment variable **QUERY_STRING**

Client side: sending information using **GET**



CGI GET Example

First: Last: Submit

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>CGI GET Example</title>
  </head>
  <body>
    <h1>CGI GET Example</h1>
    <form action="http:/localhost/cgi-bin/get.cgi" method="GET">
      First: <input type="text" name="first" size=12>
      Last:  <input type="text" name="last" size=12>
      <input type="submit">
    </form>
  </body>
</html>
```

Type

MIME Type	text/html
Resource Type	Document

Location

Full URL	file:///Users/sandiway/courses/ling508-18/form-get.html
Scheme	file
Path	/Users/sandiway/courses/ling508-18/form-get.html
Filename	form-get.html

Server side: sending information using **GET**

- **get.cgi**:

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
#echo $QUERY_STRING
origIFS=$IFS
IFS='=&'
set -- $QUERY_STRING
IFS=$origIFS
echo "1:<$1> 2:<$2> 3:<$3> 4:<$4>"
```

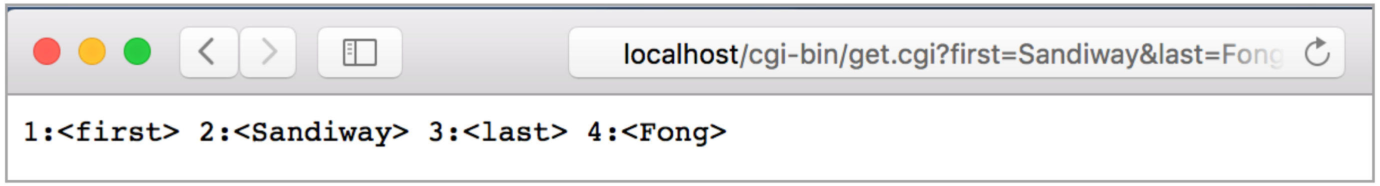
= *and* &

```
http://localhost/cgi-bin/get.cgi?first=Sandiway&last=Fong
```

In **bash**:

- IFS = **internal field separator** (for arguments)
- default: space newline tab
- **`set -- `*`String`***
- -- option: positional parameters **`$1,$2`**,..etc. are set after splitting *String*

Server side: sending information using **GET**

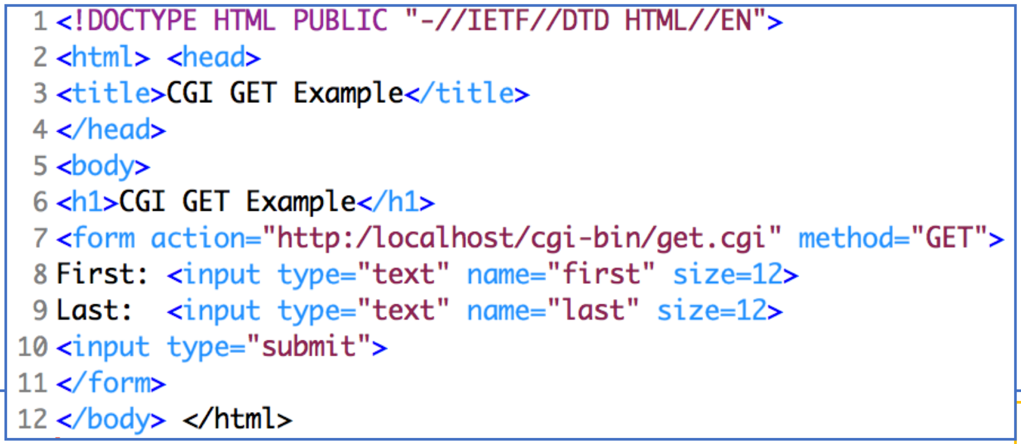


```
1:<first> 2:<Sandiway> 3:<last> 4:<Fong>
```

- **get.cgi**:

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
#echo $QUERY_STRING
origIFS=$IFS
IFS='=&'
set -- $QUERY_STRING
IFS=$origIFS
echo "1:<$1> 2:<$2> 3:<$3> 4:<$4>"
```

Server side: sending information using GET



```html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html> <head>
<title>CGI GET Example</title>
</head>
<body>
<h1>CGI GET Example</h1>
<form action="http:/localhost/cgi-bin/get.cgi" method="GET">
First: <input type="text" name="first" size=12>
Last:  <input type="text" name="last" size=12>
<input type="submit">
</form>
</body> </html>
```

MacOS:

- `/Library/WebServer/CGI-Executables/`
- `$ls -l get.cgi`
- `-rwxr-xr-x  1 root  wheel  161 Oct 16  2014 get.cgi`
- `sudo chmod 755 get.cgi`

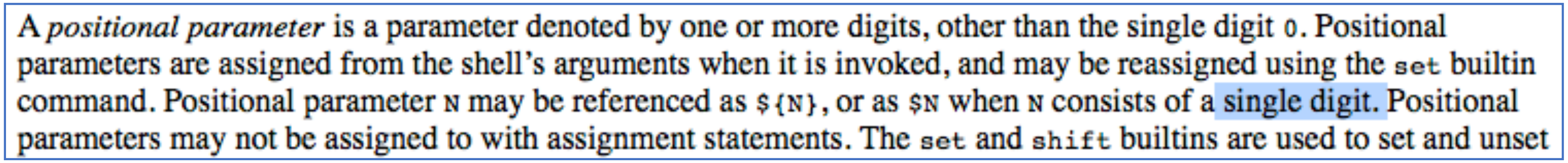
Ubuntu:

`/usr/lib/cgi-bin/`

Limitations of positional parameters

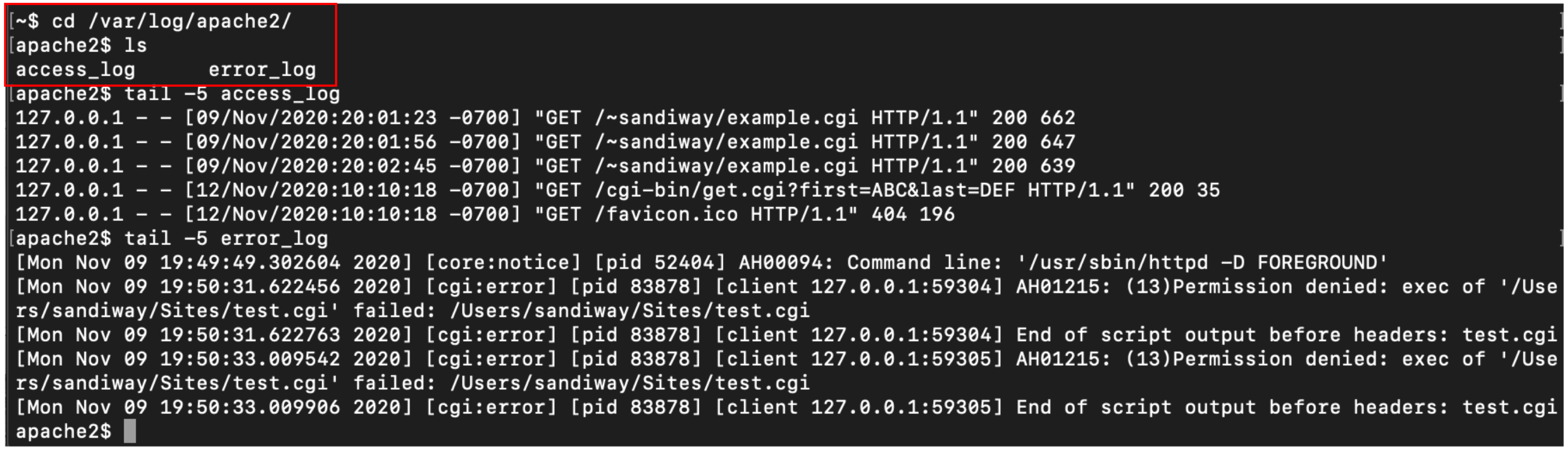
- set values:

```
origIFS=$IFS
IFS='=&'
set -- $QUERY_STRING
IFS=$origIFS
echo "1:<$1> 2:<$2> 3:<$3> 4:<$4>"
```



A *positional parameter* is a parameter denoted by one or more digits, other than the single digit `0`. Positional parameters are assigned from the shell's arguments when it is invoked, and may be reassigned using the `set` builtin command. Positional parameter `N` may be referenced as `${N}`, or as `$N` when `N` consists of a single digit. Positional parameters may not be assigned to with assignment statements. The `set` and `shift` builtins are used to set and unset

Webserver logs



```
~$ cd /var/log/apache2/
apache2$ ls
access_log      error_log
apache2$ tail -5 access_log
127.0.0.1 - - [09/Nov/2020:20:01:23 -0700] "GET /~sandiway/example.cgi HTTP/1.1" 200 662
127.0.0.1 - - [09/Nov/2020:20:01:56 -0700] "GET /~sandiway/example.cgi HTTP/1.1" 200 647
127.0.0.1 - - [09/Nov/2020:20:02:45 -0700] "GET /~sandiway/example.cgi HTTP/1.1" 200 639
127.0.0.1 - - [12/Nov/2020:10:10:18 -0700] "GET /cgi-bin/get.cgi?first=ABC&last=DEF HTTP/1.1" 200 35
127.0.0.1 - - [12/Nov/2020:10:10:18 -0700] "GET /favicon.ico HTTP/1.1" 404 196
apache2$ tail -5 error_log
[Mon Nov 09 19:49:49.302604 2020] [core:notice] [pid 52404] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
[Mon Nov 09 19:50:31.622456 2020] [cgi:error] [pid 83878] [client 127.0.0.1:59304] AH01215: (13)Permission denied: exec of '/Use
rs/sandiway/Sites/test.cgi' failed: /Users/sandiway/Sites/test.cgi
[Mon Nov 09 19:50:31.622763 2020] [cgi:error] [pid 83878] [client 127.0.0.1:59304] End of script output before headers: test.cgi
[Mon Nov 09 19:50:33.009542 2020] [cgi:error] [pid 83878] [client 127.0.0.1:59305] AH01215: (13)Permission denied: exec of '/Use
rs/sandiway/Sites/test.cgi' failed: /Users/sandiway/Sites/test.cgi
[Mon Nov 09 19:50:33.009906 2020] [cgi:error] [pid 83878] [client 127.0.0.1:59305] End of script output before headers: test.cgi
apache2$
```

Files and Locations

Server cgi-bin directory:

- MacOS: `/Library/WebServer/CGI-Executables`
- Ubuntu: `/usr/lib/cgi-bin`

Webserver logs:

- MacOS: `/var/log/apache2/error_log`
- Ubuntu: `/var/log/apache2/error.log`

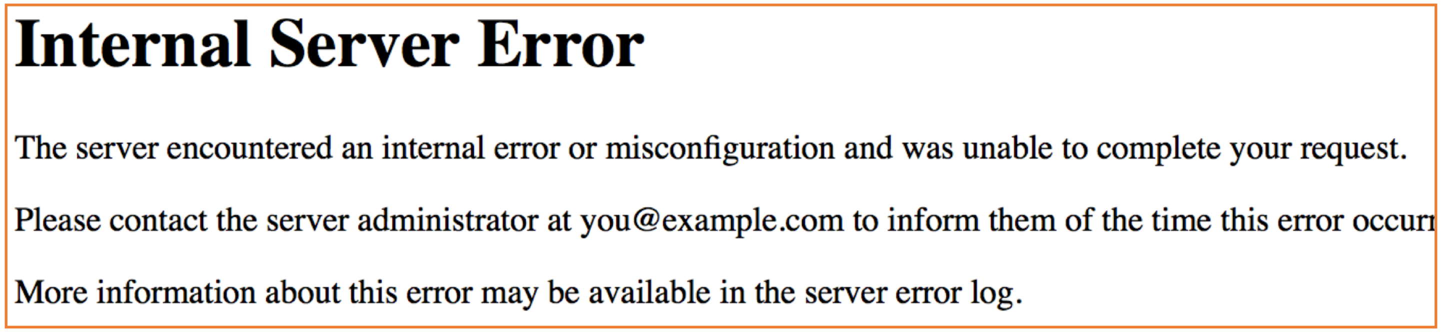
Course webpage:

- `form-get.html`
- `form-post.html`
- `get.cgi` (needs 755 permissions)
- `read.cgi` (needs 755 permissions)

Error Log

Example:

can't access file



Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at you@example.com to inform them of the time this error occurr

More information about this error may be available in the server error log.

error log:

```
…
[Wed Oct 31 22:08:01.555932 2018] [cgi:error] [pid 4600]
[client ::1:51340] AH01215: (13)Permission denied: exec of
'/Users/sandiway/Sites/get2.cgi' failed:
/Users/sandiway/Sites/get2.cgi
[Wed Oct 31 22:08:01.556059 2018] [cgi:error] [pid 4600]
[client ::1:51340] End of script output before headers: get2.cgi
```

Ungraded Exercise

- Make the GET example work on your computer.
- Next time, we'll review the POST method …