Lecture 19

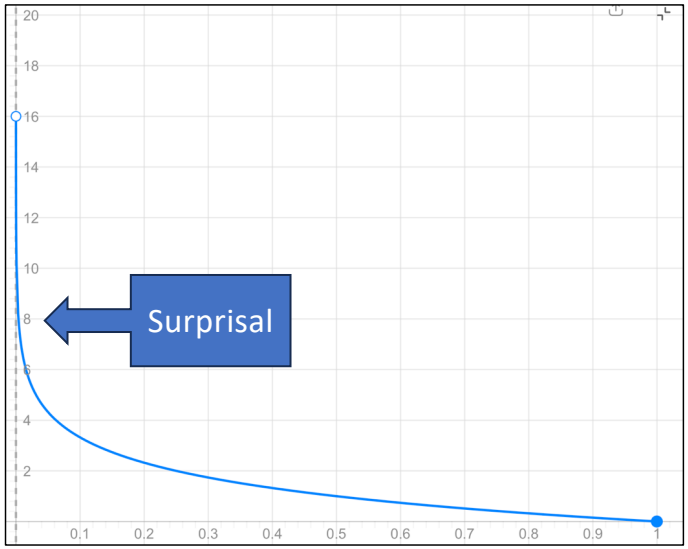
408/508 *Computational Techniques for Linguists*

Today's Topics

- **Term project idea**: interactive **Javascript** and **Plotly.js**
- `sandiway.arizona.edu`: php problem
- `cgi-bin` (*Common Gateway Interface binaries*)

Demo: Surprisal

- What is **surprisal**?
  - *Expectations regarding the next word play a central role in **language comprehension**, such that listeners process **expected words** with greater ease than **less expected ones**. **Surprisal theory** formalizes this relationship* (Krieger e al. 2025)
  - -lg P(*w* |*preceding context*) a measure of the likelihood of a word *w* that can immediately follow a given context
  - High surprisal = lower predictability, and thus some difficulties for perception/cognitive processing
  - **Information Theory**: *highly predictable events* communicate little new information, whereas *unexpected events* have a heavy informational load



Demo: Surprisal

- Difficult to measure:
  - *Critically, the true expectancy of a word should in principle reflect all relevant determinants of what words are likely to appear next — including **distributional**, **linguistic**, and **world knowledge**-based plausibility constraints.* (Krieger e al. 2025)
- **LLM** Proxy:
  - *the traffic cop wrote the bicycle (high surprisal)*
  - *article* (medium surprisal)
  - *ticket* (low surprisal)
  - `llm_surprisal (models)`
    - `openai-community/openai-gpt` `117M`
    - `openai-community/gpt2` `124M`
    - `openai-community/gpt2-large` `774M`
    - `openai-community/gpt2-xl` `1.5B`

Demo: Surprisal

LLM Surprisal 3D Ribbon Plot

Model	Parameter
openai-community/openai-gpt	117M
openai-community/gpt2	124M
openai-community/gpt2-large	774M
openai-community/gpt2-xl	1.5B

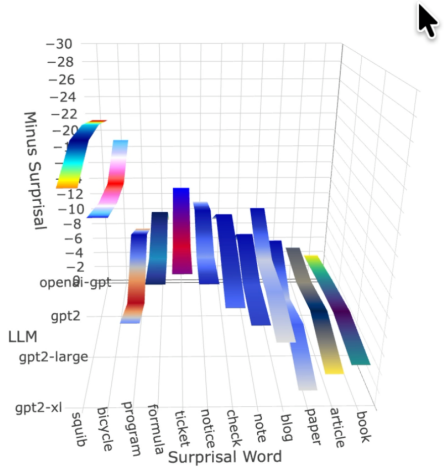
Sentence

The researcher wrote the _____

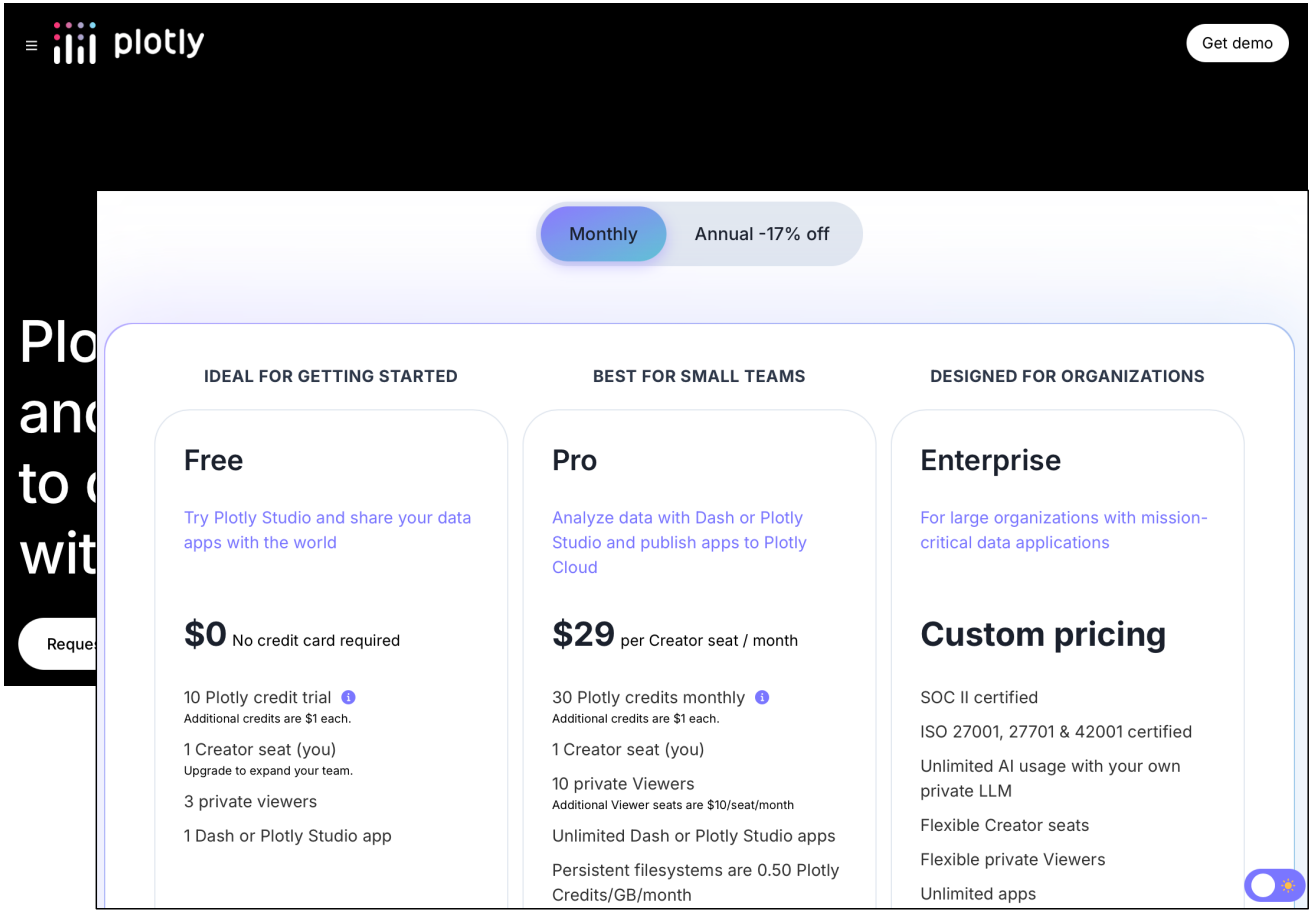
surprisal word

Surprisal threshold: -30

Instructions:
find the word with the smallest surprisal magnitude corresponding to each subject (in the pulldown menu) by sliding the threshold to the right. Rotate the image with the LLM axis in front to see which model performs the best. [*Suggested end threshold: -4*.]
Note: 3D diagram can be rotated, zoomed etc. using pop-up controls (top-right of 3D image)



Plotly.js


plotly
Get demo
Monthly
Annual -17% off
IDEAL FOR GETTING STARTED
BEST FOR SMALL TEAMS
DESIGNED FOR ORGANIZATIONS
Free
Try Plotly Studio and share your data apps with the world
$0 No credit card required
10 Plotly credit trial
Additional credits are $1 each.
1 Creator seat (you)
Upgrade to expand your team.
3 private viewers
1 Dash or Plotly Studio app
Pro
Analyze data with Dash or Plotly Studio and publish apps to Plotly Cloud
$29 per Creator seat / month
30 Plotly credits monthly
Additional credits are $1 each.
1 Creator seat (you)
10 private Viewers
Additional Viewer seats are $10/seat/month
Unlimited Dash or Plotly Studio apps
Persistent filesystems are 0.50 Plotly Credits/GB/month
Enterprise
For large organizations with mission-critical data applications
Custom pricing
SOC II certified
ISO 27001, 27701 & 42001 certified
Unlimited AI usage with your own private LLM
Flexible Creator seats
Flexible private Viewers
Unlimited apps

Plotly.js

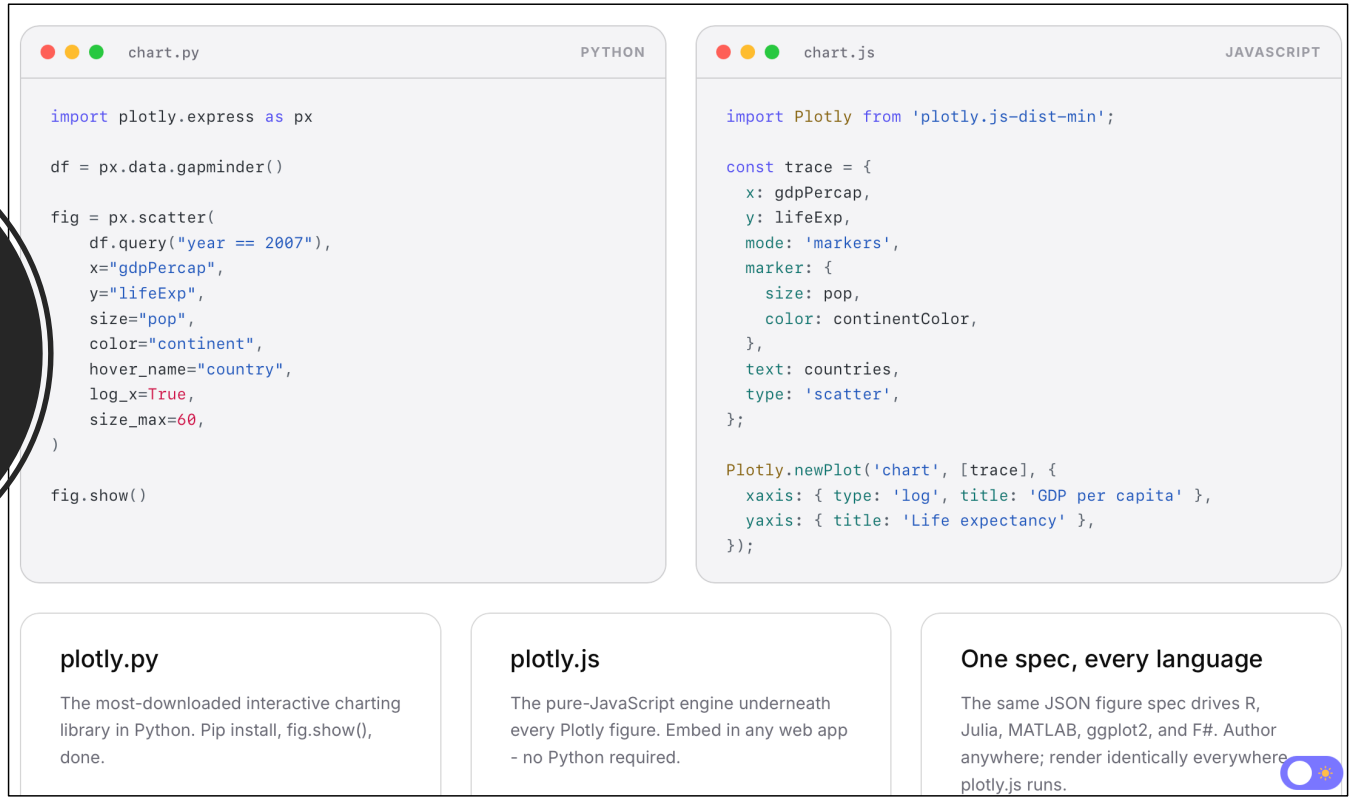


chart.py PYTHON

```python
import plotly.express as px

df = px.data.gapminder()

fig = px.scatter(
    df.query("year == 2007"),
    x="gdpPercap",
    y="lifeExp",
    size="pop",
    color="continent",
    hover_name="country",
    log_x=True,
    size_max=60,
)

fig.show()
```

chart.js JAVASCRIPT

```javascript
import Plotly from 'plotly.js-dist-min';

const trace = {
  x: gdpPercap,
  y: lifeExp,
  mode: 'markers',
  marker: {
    size: pop,
    color: continentColor,
  },
  text: countries,
  type: 'scatter',
};

Plotly.newPlot('chart', [trace], {
  xaxis: { type: 'log', title: 'GDP per capita' },
  yaxis: { title: 'Life expectancy' },
});
```

plotly.py

The most-downloaded interactive charting library in Python. Pip install, fig.show(), done.

plotly.js

The pure-JavaScript engine underneath every Plotly figure. Embed in any web app - no Python required.

One spec, every language

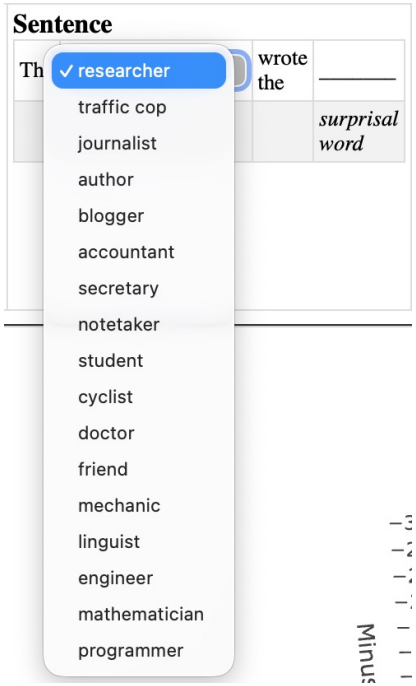
The same JSON figure spec drives R, Julia, MATLAB, ggplot2, and F#. Author anywhere; render identically everywhere plotly.js runs.

How did I build the tool?

- Construct a pull-down menu (`div`)

```
<td><div style="display: inline-block" id="dropdown-container"></div></td>
```

```
<script>
const person = ['researcher', 'traffic cop', 'journalist', 'author', 'blogger',
'accountant', 'secretary', 'notetaker', 'student', 'cyclist', 'doctor', 'friend',
'mechanic', 'linguist', 'engineer', 'mathematician', 'programmer'];
const select = document.createElement("select");
select.id = "dynamic-select";
person.forEach(item => {
        const option = document.createElement("option");
        option.value = item.toLowerCase().replace(/\s+/g, "-"); // Create a safe value
        option.text = item; // Text visible to the user
        select.appendChild(option);
});
const container = document.getElementById("dropdown-container");
container.appendChild(select);
var selected = 0;
</script>
```



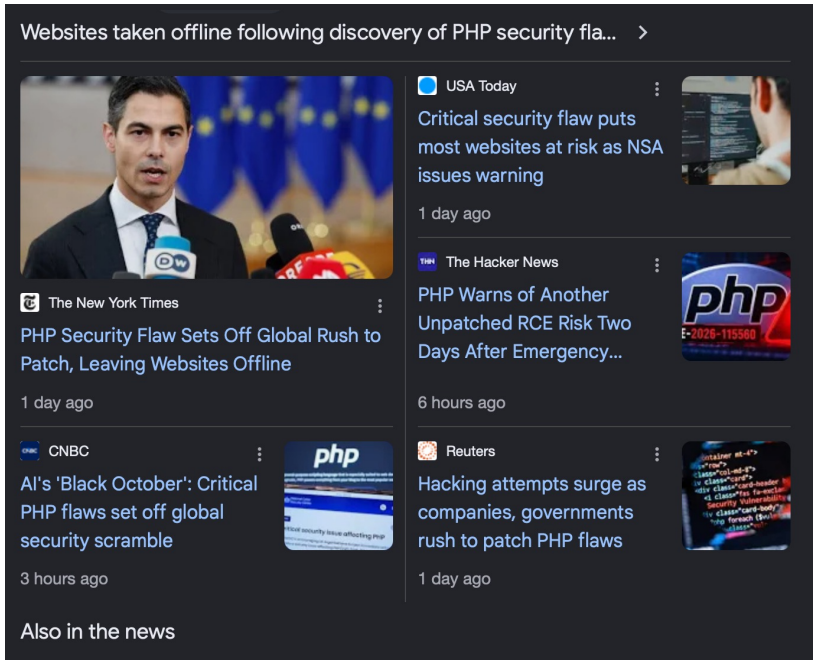
How did I build the tool?

```
const rawData = [
`
book    3.8             4.7             5.4             4.6
article 4.9             4.4             5.5             3.6
paper   5.9             2.2             3.9             1.8
blog    10.3            7.3             7.8             7.1
note    6.0             10.5            8.6             8.9
check   8.9             12.9            12.1            10.7
notice  11.2            13.4            11.7            13.1
ticket  12.1            16.0            12.8            14.1
formula 8.8             13.2            11.6            13.1
program 7.6             10.5            6.5             9.2
bicycle 19.0            16.5            17.5            19.4
squib   21.4            23.0            23.7            22.0
`,
`
book    7.6             8.4             8.6             8.8
article 7.9             9.3             10.9            9.4
paper   7.3             7.9             9.2             9.0
blog    13.9            10.1            12.4            11.5
note    5.0             5.4             6.1             7.0
check   6.7             7.8             8.0             7.8
notice  9.0             8.8             7.8             8.0
ticket  5.8             5.8             1.4             1.6
formula 13.0            13.5            15.1            12.5
program 10.9            12.7            12.2            13.9
bicycle 16.5            11.7            14.8            14.7
squib   21.5            21.9            23.7            26.5
`,
```

- Plot function:

```
function plot() {
    parsed = parseDataToRibbons(rawData[selected], 1.5);
    layout.scene.xaxis.tickvals = parsed.xTickVals;
    layout.scene.xaxis.ticktext = parsed.xTickText;
    Plotly.newPlot('ribbonDiv', parsed.traces, layout);
};
```

Why my page was off-line



Security advisory: Critical remote code execution vulnerabilities, update now 21 Oct 2026

PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35 fix two critical vulnerabilities that can allow unauthenticated remote code execution. All PHP users are strongly encouraged to update immediately.

Who is at risk: all PHP installations serving web requests, including PHP-FPM, Apache with mod_php, php-cgi, FrankenPHP and LiteSpeed. We consider the risk highest for mod_php, with a demonstrated exploit where an unauthenticated request can trigger the attack before any application code runs.

The vulnerabilities:

- **RCE via OOM corruption**, CVE-2026-109618 (GHSA-w8hw-w9fx-j79w): out-of-memory handling can interrupt PHP while internal request data is only partially initialized. Request cleanup may then treat the invalid data as valid.
- **RCE via stale global state**, CVE-2026-109619 (GHSA-6wmf-4rv5-p3rp): request-owned internal state can survive after its backing memory has been freed. A later request handled by the same worker may use that stale state before normal initialization resets it.

How to update:

- Update to PHP 8.5.12, 8.4.27, 8.3.36 or 8.2.35. Source downloads are on our downloads page. Windows source and binaries are on the same page.
- Linux distributions and LTS vendors received the patches in advance, so check your vendor for updated packages.
- **PHP 8.1 and older** no longer receive updates from the PHP project. Some distributions and LTS vendors provide backported fixes, so contact your vendor for details, or upgrade to a supported version.
- Given the severity of these vulnerabilities, the PHP security team provided minimal source-backports for PHP 7.4, PHP 8.0, and PHP 8.1 to help people building from source or without access to an LTS vendor.

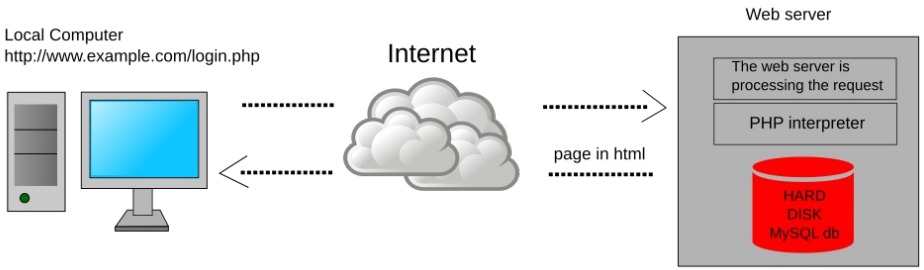
What is PHP?

The following "Hello, World!" program is written in PHP code embedded in an HTML document:

```
<!DOCTYPE html>
<html>
    <head>
        <title>PHP "Hello, World!" program</title>
    </head>
    <body>
        <p><?= 'Hello, World!' ?></p>
    </body>
</html>
```

```
<?php
    $name = 'John'; $age = 18;
?>
```

- Server side scripting embedded in a html file.
- Javascript is client side
- <?php … ?> means execute code
- <?= … ?> means echo (*like Bash*)



CGI-bin

- https://en.wikipedia.org/wiki/Common_Gateway_Interface

> In computing, **Common Gateway Interface** (**CGI**) is an interface specification for web servers to execute programs like console applications (also called command-line interface programs) running on a server that generates web pages dynamically. Such programs are known as *CGI scripts* or simply as *CGIs*. The specifics of how the script is executed by the server are determined by the server. In the common case, a CGI script executes at the time a request is made and generates HTML.[1]

- Again, unfortunately Apple and Linux setups are different …

test.cgi

- On course webpage:

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running a Bash shell script from cgi-bin successfully!"
echo -n "Now it's: "
date
echo -n "User: "
whoami
echo -n "Directory: "
ls -l
exit 0
```

tells the receiving browser what to expect: output of each command is sent to the browser

Boilerplate browser expects is:

```
Content-Type: text/plain
<blank line>
```

Content

date

NAME

date – display or set date and time

DESCRIPTION

When invoked without arguments, the date utility displays the current date and time.

whoami

NAME

whoami – display effective user id

SYNOPSIS

whoami

DESCRIPTION

The whoami utility has been obsoleted by the id(1) utility, and is equivalent to “id -un”. The command “id -p” is suggested for normal interactive use.

The whoami utility displays your effective user ID as a name.

`cgi-bin` on macOS

```
1#!/bin/bash¶
2echo "Content-Type: text/plain"¶
3echo¶
4echo "Running a Bash shell script from cgi-bin successfully!"¶
5echo -n "Now it's: "¶
6date¶
7echo -n "User: "¶
8whoami¶
9echo -n "Directory: "¶
10ls -l¶
11exit 0¶
```

- Put it in directory
  - `/Library/WebServer/CGI-Executables`
  - `sudo` needed (owned by *root*)
- URL:
  - http://localhost/cgi-bin/test.cgi

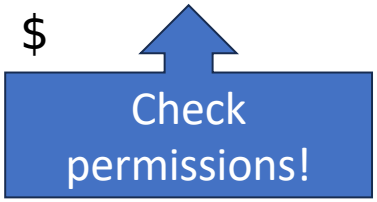
test.cgi must be executable and readable chmod 755



WRONG! *Prints the contents of the script instead of running it*!

`cgi-bin` on macOS

```
$ cd /Library/WebServer/CGI-Executables/
$ ls -l test.cgi
-rwxr-xr-x  1 root  wheel  196 Nov  1  2022 test.cgi
$
```



`cgi-bin` on macOS

- Assuming the webserver is running,

```
$ ps -ax | grep httpd
17614 ??         0:09.45 /usr/sbin/httpd -D FOREGROUND
17618 ??         0:00.00 /usr/sbin/httpd -D FOREGROUND
17619 ??         0:00.01 /usr/sbin/httpd -D FOREGROUND
26312 ttys000    0:00.00 grep httpd
```

- possible: *404 Not Found: some_program.cgi doesn't exist!*

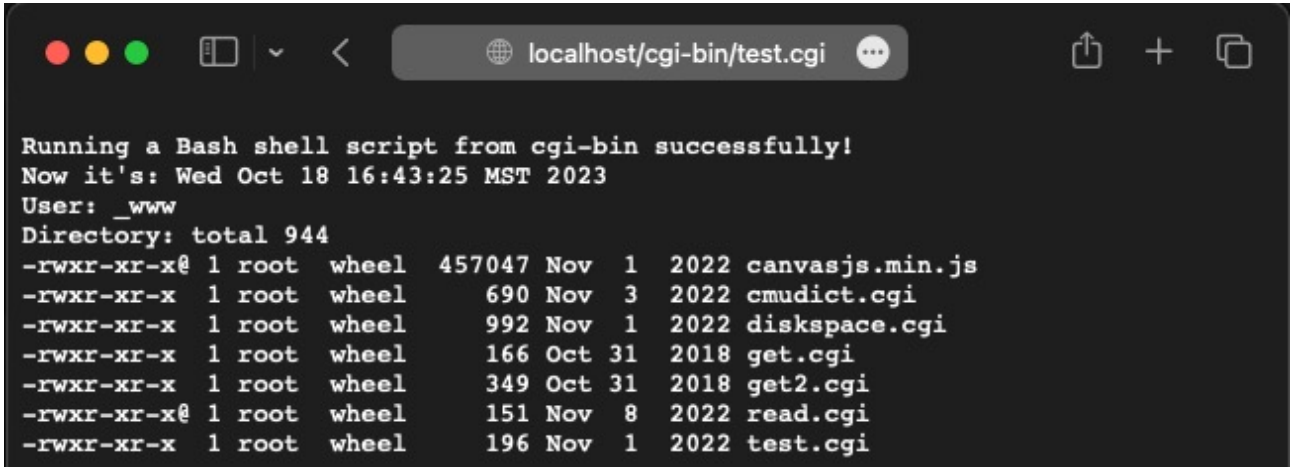


`cgi-bin` on macOS

- By default, the cgi module is turned off
- Instructions:
  1. in file `/etc/apache2/httpd.conf`
     - `sudo nano /etc/apache2/httpd.conf`
     - and uncomment line shown ☞
  2. restart apache2:
     - `sudo apachectl –k restart`

```
167 LoadModule autoindex_module libexec/apache2/mod_autoindex.so
168 #LoadModule asis_module libexec/apache2/mod_asis.so
169 #LoadModule info_module libexec/apache2/mod_info.so
170 <IfModule !mpm_prefork_module>
171         #LoadModule cgid_module libexec/apache2/mod_cgid.so
172 </IfModule>
173 <IfModule mpm_prefork_module>
174         #LoadModule cgi_module libexec/apache2/mod_cgi.so
175 </IfModule>
176 #LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
177 #LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
```

`cgi-bin` on macOS



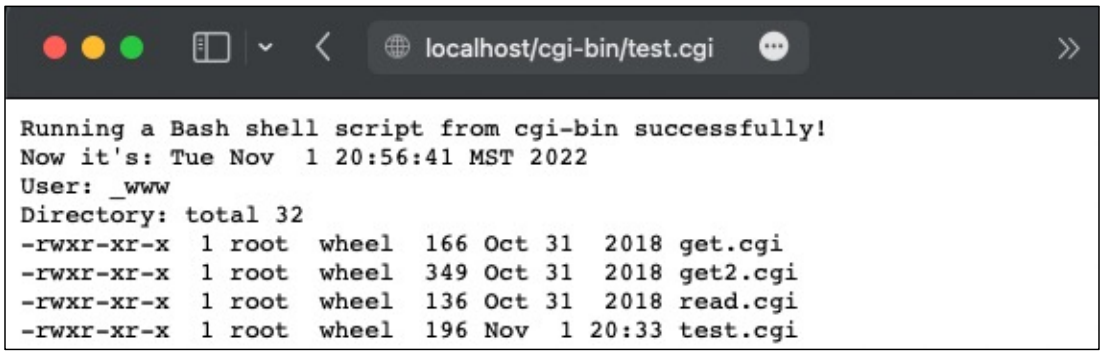
`cgi-bin` on macOS

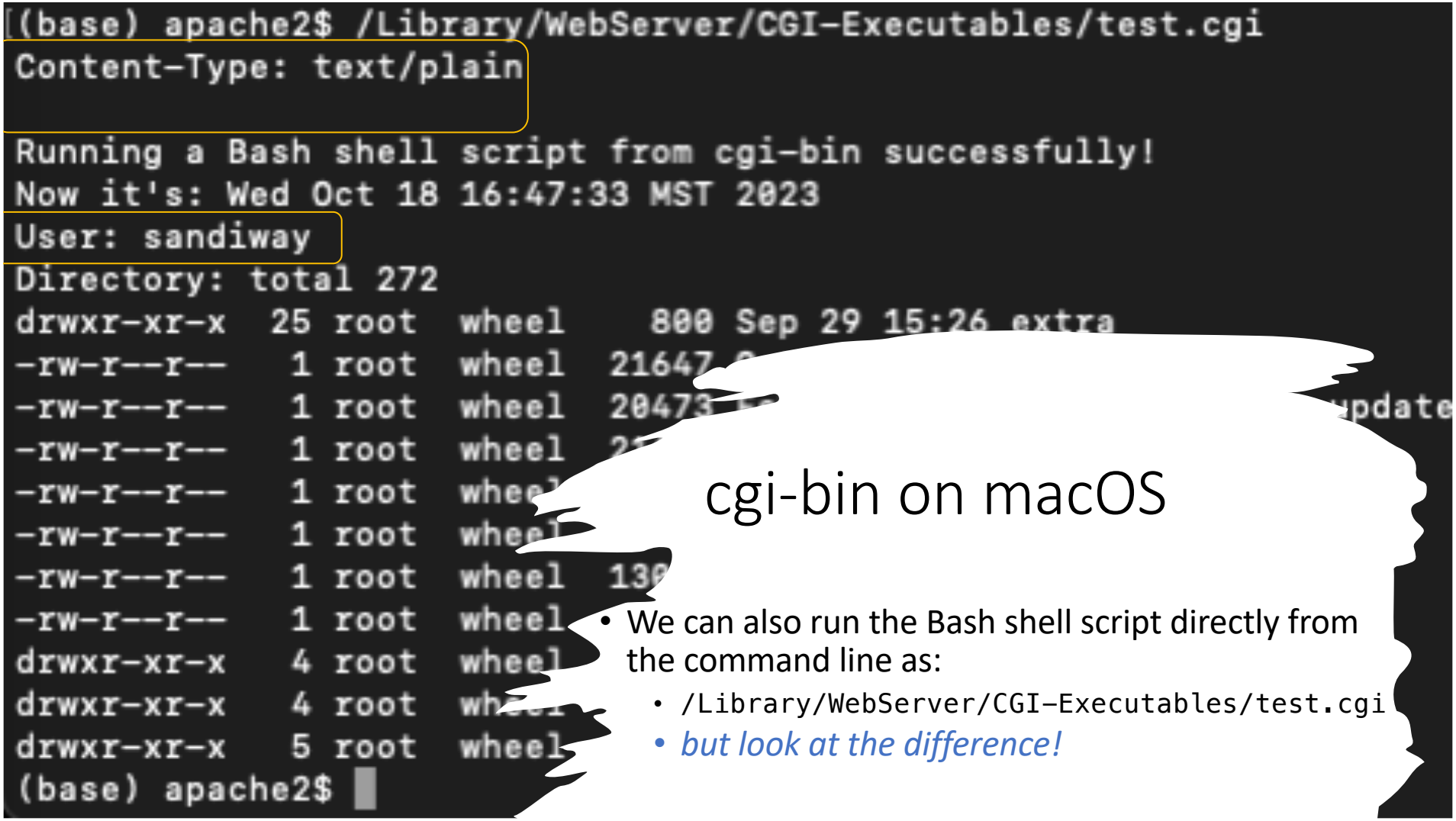
Boilerplate browser expects:

```
Content-Type: text/plain
<blank line>
```

Content

```
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Running a Bash shell script from cgi-bin successfully!"
echo -n "Now it's: "
date
echo -n "User: "
whoami
echo -n "Directory: "
ls -l
exit 0
```





cgi-bin on macOS

- We can also run the Bash shell script directly from the command line as:
  - `/Library/WebServer/CGI-Executables/test.cgi`
  - *but look at the difference!*

`cgi-bin` on on Ubuntu

- http://localhost/cgi-bin/
- CGI binaries directory: **`/usr/lib/cgi-bin/`**
  - `test.cgi` *must be made executable!*

```
sandiway@sandiway-VirtualBox:/usr/lib$ ls cgi-bin
sandiway@sandiway-VirtualBox:/usr/lib$ cd cgi-bin
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ls -l
total 0
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ sudo nano test.cgi
[sudo] password for sandiway:
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ more test.cgi
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo "Run Bash script from /usr/lib/cgi-bin successfully!"
echo -n "Now: "
date
echo -n "User: "
whoami
exit 0

sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ./test.cgi
bash: ./test.cgi: Permission denied
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ chmod 755 test.cgi
chmod: changing permissions of 'test.cgi': Operation not permitted
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ sudo chmod 755 test.cgi
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ./test.cgi
Content-Type: text/plain

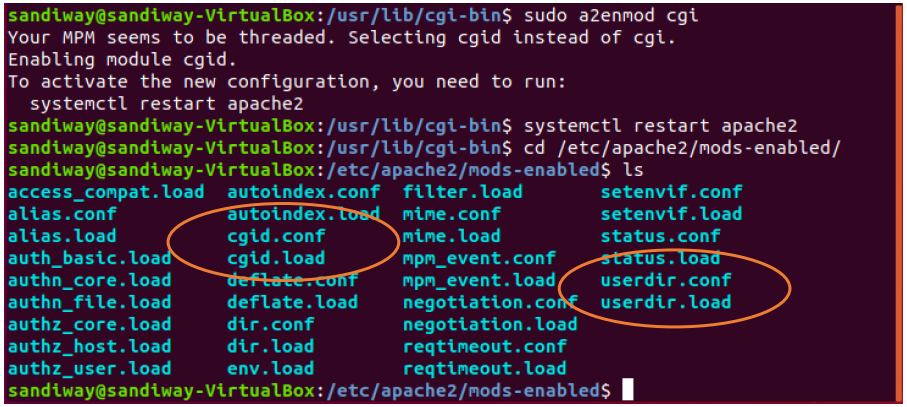
Run Bash script from /usr/lib/cgi-bin successfully!
Now: Tue Oct 30 11:24:21 MST 2018
User: sandiway
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$
```

create same test script

executable permissions needed

`cgi-bin` on Ubuntu

- Enabling `cgi-bin` on VirtualBox:
  - **`sudo a2enmod cgi`** (*enables cgid instead of cgi*)
  - directory **`/etc/apache2/mods-enabled/`**



in WSL, systemctl doesn't seem to work, use
`sudo service apache2 restart`

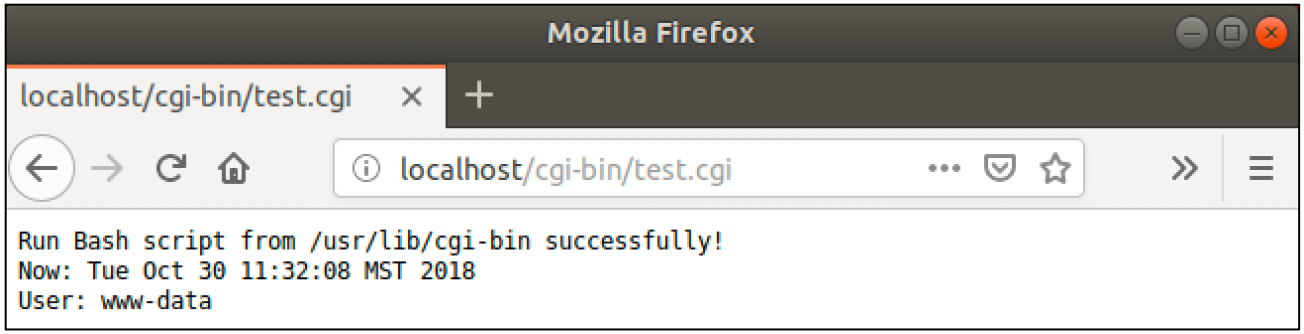
Apache Webserver on Ubuntu

- Compare running `test.cgi` directly:

```
sandiway@sandiway-VirtualBox:/usr/lib/cgi-bin$ ./test.cgi
Content-Type: text/plain

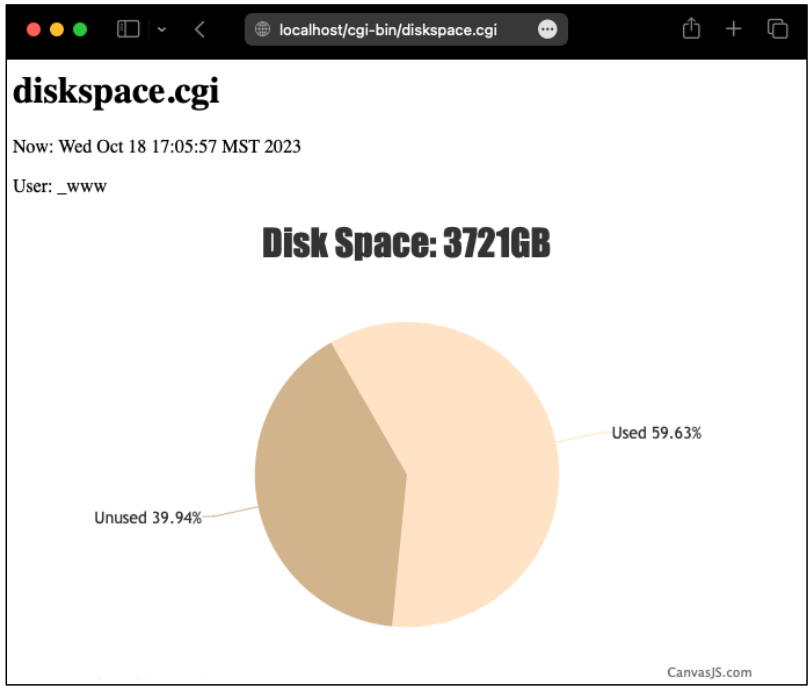
Run Bash script from /usr/lib/cgi-bin successfully!
Now: Tue Oct 30 11:24:21 MST 2018
User: sandiway
```

- `http//localhost/cgi-bin/test.cgi:`



User: **`www-data`**
On macOS: **`_www`**

An Example using HTML/Javascript



- Course website:
  - `diskspace.cgi`
  - `canvasjs.min.js`
- Instead of

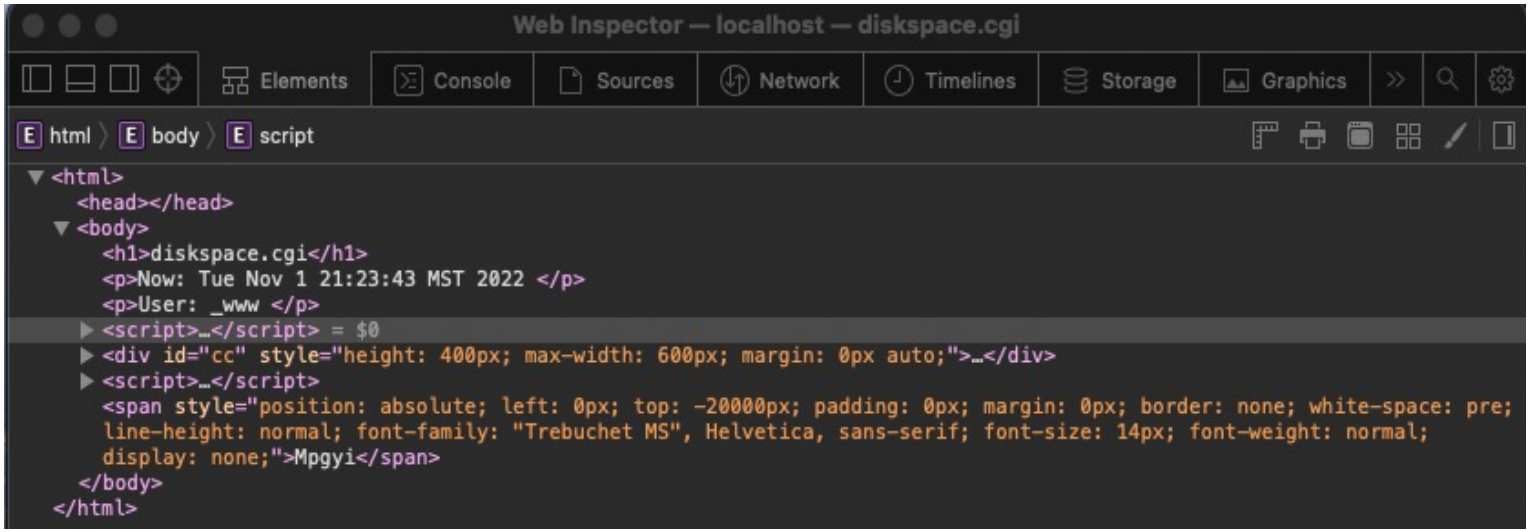
```
Content-Type: text/plain
(blank line)
```

- we use:

```
Content-Type: text/html; charset=utf-8
(blank line)
```

An Example using HTML/Javascript

- Let's look inside the source code!



An Example using HTML/Javascript

diskspace.cgi:

```
#!/bin/bash
echo "Content-Type: text/html; charset=utf-8"
echo
echo "<html><head></head>"
echo "<body><h1>diskspace.cgi</h1>"
echo -n "<p>Now: "
date
echo "</p>"
echo -n "<p>User: "
whoami
echo "</p>"
capacity=$(df -g | awk 'NR==10 {print $2}')
used=$(df -g | awk 'NR==10 {print $3}')
unused=$(df -g | awk 'NR==10 {print $4}')
```

An Example using HTML/Javascript

```
echo "<script>"
cat canvasjs.min.js
echo "</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
```

An Example using HTML/Javascript

- `$ df –g` (*on my M1 Apple laptop*), or `df` (`df –g` doesn't work on Ubuntu)

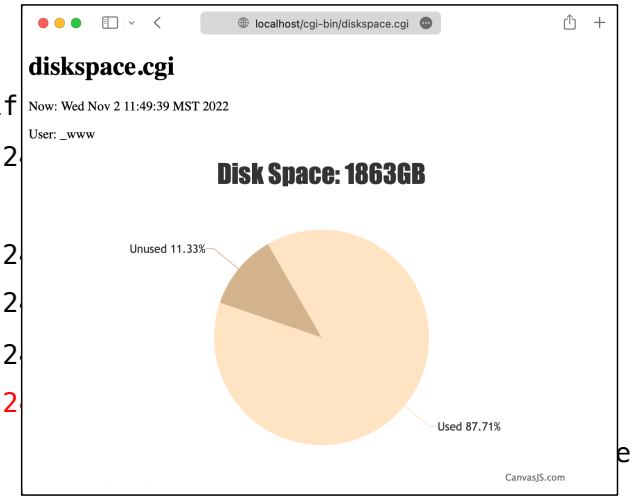
```
1. Filesystem                  1G-blocks Used Available Capacity  iused       ifree %iused  Mounted on
2. /dev/disk3s1s1                   3721    9      1486     1%   387452  4292442203    0%   /
3. devfs                               0    0         0   100%      744           0  100%   /dev
4. /dev/disk3s6                     3721    0      1486     1%        0 15591697480    0%   /System/Volumes/VM
5. /dev/disk3s2                     3721    5      1486     1%     1040 15591697480    0%   /System/Volumes/Preboot
6. /dev/disk3s4                     3721    0      1486     1%       45 15591697480    0%   /System/Volumes/Update
7. /dev/disk1s2                        0    0         0     2%        1     4916120    0%   /System/Volumes/xarts
8. /dev/disk1s1                        0    0         0     2%       28     4916120    0%   /System/Volumes/iSCPreboot
9. /dev/disk1s3                        0    0         0     1%       52     4916120    0%   /System/Volumes/Hardware
10./dev/disk3s5                     3721 2219      1486    60%  4025155 15591697480    0%   /System/Volumes/Data
11.map auto_home                       0    0         0   100%        0           0     -   /System/Volumes/Data/home
```

```
• capacity=$(df –g | awk 'NR==10 {print $2}')
• used=$(df –g | awk 'NR==10 {print $3}')
• unused=$(df –g | awk 'NR==10 {print $4}')
```

An Example using HTML/Javascript

- `$ df –g` (*on my Intel Apple laptop*)

```
1.Filesystem     1G-blocks Used Available Capacity iused      if
2./dev/disk1s5s1      1863   14       211     7%  502068 2217512
3.devfs                  0    0         0   100%     666
4./dev/disk1s4        1863    1       211     1%       1 2217512
5./dev/disk1s2        1863    0       211     1%    4571 2217512
6./dev/disk1s6        1863    0       211     1%      18 2217512
7./dev/disk1s1        1863 1634       211    89% 3813194 2217512
8.map auto_home          0    0         0   100%       0
```




```
capacity=$(df -g | awk 'NR==7 {print $2}')
used=$(df -g | awk 'NR==7 {print $3}')
unused=$(df -g | awk 'NR==7 {print $4}')
```

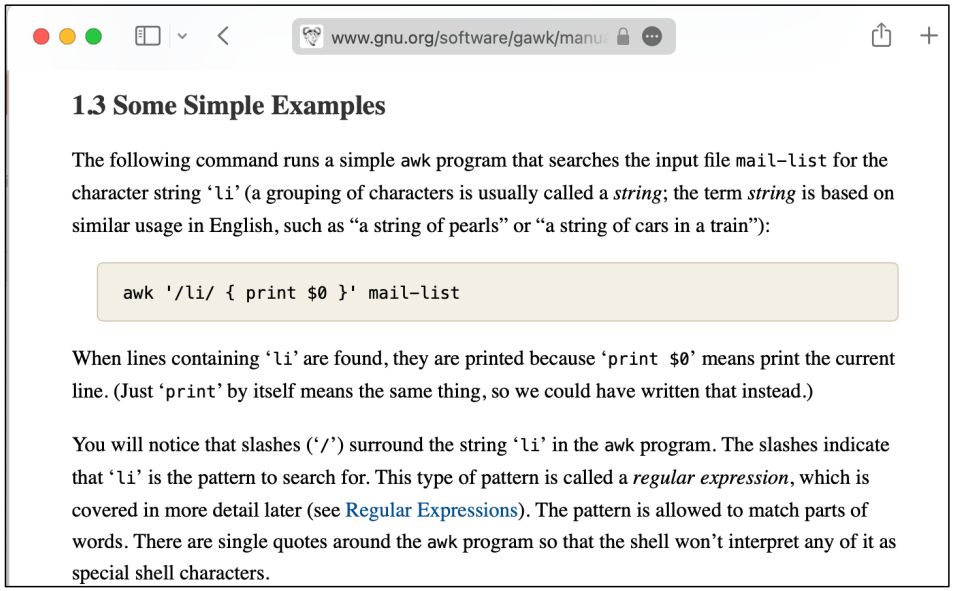
What is awk?

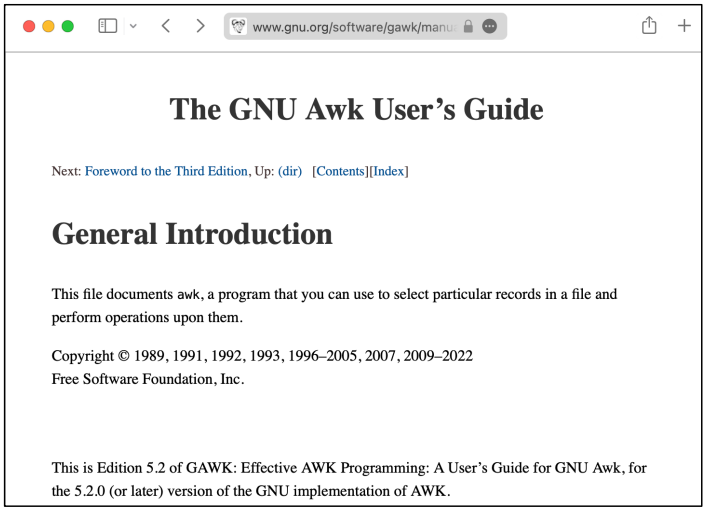
- record:
  - basically, a line
- NR:
  - NR is the number of records read so far
- $1:
  - You use a dollar sign (‘$’) to refer to a field in an awk program, followed by the number of the field you want.
  - Thus, $1 refers to the first field, $2 to the second, and so on.

- Idea:
  - When awk reads an input record, the record is automatically *parsed* or separated by the awk utility into chunks called *fields*.
  - By default, fields are separated by *whitespace*, like words in a line.

What is awk?

https://www.gnu.org/software/gawk/manual/gawk.html





Documentation

http://httpd.apache.org/docs/current/

