Tutorons
Generating Context-Relevant, On-Demand Explanations and Demonstrations of Online Code

Andrew Head, Codanda Appachu, Marti A. Hearst, Björn Hartmann
Computer Science Division, UC Berkeley
unix command to fetch all tarball files from website
Automatically download music

This last technique, suggested by Jeff Veen, is by far my favorite use of Wget. These days there are tons of directories, aggregators, filters and weblogs that point off to interesting types of media. Using Wget, you can create a text file list of your favorite sites that say, link to MP3 files, and schedule it to automatically download any newly-added MP3's from those sites each day or week.

First, create a text file called mp3_sites.txt, and list URLs of your favorite sources of music online one per line (like http://del.icio.us/tag/system:file... or stereogum.com). Be sure to check out my previous feature on how to find free music on the web for more ideas.

Then use the following Wget command to go out and fetch those MP3's:

```
wget -r -1L -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```

That Wget recipe recursively downloads only MP3 files linked from the sites listed in mp3_sites.txt that are newer than any you've already downloaded. There are a few other specifications in there - like to not create a new directory for every music file, to ignore robots.txt and to not crawl up to the parent directory of a link. Jeff breaks it all down in his original post.

The great thing about this technique is that once this command is scheduled, you get an ever-rotating jukebox of new music Wget fetches for you while you sleep. With a good set of trusted sources, you'll never have to go looking for new music again - Wget will do all the work for you.

Install Wget

Wanna give all this a try? Windows users, you can download Wget here; Mac users, go here. An alternative for Windows users interested in more Linuxy goodness is to download and install the Unix emulator Cygwin which includes Wget and a whole slew of other 'nixy utilities, too.
Then use the following `Wget` command to go out and fetch those MP3's:

```
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```
Table of Contents

1 Overview
2 Invoking
  2.1 URL Format
  2.2 Option Syntax
  2.3 Basic Startup Options
  2.4 Logging and Input File Options
  2.5 Download Options
  2.6 Directory Options
  2.7 HTTP Options
  2.8 HTTPS (SSL/TLS) Options
  2.9 FTP Options
  2.10 Recursive Retrieval Options
  2.11 Recursive Accept/Reject Options
  2.12 Exit Status
3 Recursive Download
4 Following Links
  4.1 Spanning Hosts
  4.2 Types of Files
  4.3 Directory-Based Limits
  4.4 Relative Links
  4.5 Following FTP Links
5 Time-Stamping
  5.1 Time-Stamping Usage
  5.2 HTTP Time-Stamping Internals
  5.3 FTP Time-Stamping Internals
6 Startup File
  6.1 Wgetrc Location
  6.2 Wgetrc Syntax
Then use the following Wget command to go out and fetch those MP3's:

```
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```
Then use the following Wget command to go out and fetch those MP3’s:

```
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```
Then use the following Wget command to go out and fetch those MP3’s:

```
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```

You found a `wget` command.

`wget` is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file `mp3_sites.txt`.

Recursively scrape web pages of type `.mp3` from URLs from the file `mp3_sites.txt`, following links 1 time.

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.
- `--tries (-t)`: set number of retries to 1 (0 unlimts).
- `--no-directories (-nd)`: don’t create directories.
- `--timestamping (-N)`: don’t re-retrieve files unless newer than local.
- `--no-parent (-np)`: don’t ascend to the parent directory.
- `--accept (-A)`.mp3 is a comma-separated list of accepted extensions.
- `--execute (-e)`: execute a `.wgetrc`-style command (COMMAND=robots=off).
- `--input-file (-i)`: download URLs found in local or external mp3_sites.txt.
Then use the following `wget` command to go out and fetch those MP3’s:

```
wget -r -l1 -H -tl -nd -N -np -A .mp3 -erobots=off -i mp3_sites.txt
```

**You found a `wget` command.**

`wget` is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file `mp3_sites.txt`.

Recursively scrape web pages of type `.mp3` from URLs from the file `mp3_sites.txt`, following links 1 time.

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.
- `--tries (-t)`: set number of retries to 1 (0 unlimits).
- `--no-directories (-nd)`: don’t create directories.
- `--timestamping (-N)`: don’t re-retrieve files unless newer than local.
- `--no-parent (-np)`: don’t ascend to the parent directory.
- `--accept (-A)`: .mp3 is a comma-separated list of accepted extensions.
- `--exec (-e)`: execute a `.wgetrc`-style command (COMMAND=robots=off).
- `--input-file (-i)`: download URLs found in local or external mp3_sites.txt.
An Explanation Built by a Tutoron

You found a *wget* command.

*wget* is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file 'mp3_sites.txt'.

Recursively scrape web pages of type '.mp3' from URLs from the file 'mp3_sites.txt', following links 1 time.

It uses these options:

- **--recursive** (-r): specify recursive download.
- **--level** (-l): 1 is a maximum recursion depth (inf or 0 for infinite).
- **--span-hosts** (-H): go to foreign hosts when recursive.
- **--tries** (-t): set number of retries to 1 (0 unlimits).
- **--no-directories** (-nd): don't create directories.
- **--timestamping** (-N): don't re-retrieve files unless newer than local.
- **--no-parent** (-np): don't ascend to the parent directory.
- **--accept** (-A): .mp3 is a comma-separated list of accepted extensions.
- **--execute** (-e): execute a `.wgetrc'-style command (COMMAND=robots=off).
- **--input-file** (-i): download URLs found in local or external mp3Sites.txt.
What Explanations Can a Tutoron Build?

VS.

Low Volume of Text to Sift Through

Recursively scrape web pages of type `.mp3` from URLs from the file 'mp3_sites.txt', following links 1 time.

It uses these options:
- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.
What Explanations Can a Tutoron Build?

Rich Explanations

Diagrams

Usage Examples

Synthesized Prose
What Explanations Can a Tutoron Build?

Diagrams

Usage Examples

Synthesized Prose

The selector '.content_container_7 button' chooses buttons from elements of class 'content_container_7'.
Then use the following Wget command to go out and fetch those MP3's:

```
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```

You found a `wget` command.

`wget` is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file `mp3_sites.txt`.

Recursively scrape web pages of type `.mp3` from URLs from the file `mp3_sites.txt`, following links 1 time.

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.
- `--tries (-t)`: set number of retries to 1 (0 unlimits).
- `--no-directories (-nd)`: don't create directories.
- `--timestamping (-N)`: don't re-retrieve files unless newer than local.
- `--no-parent (-np)`: don't ascend to the parent directory.
- `--accept (-A) .mp3 is a comma-separated list of accepted extensions.
- `--execute (-e)`: execute a `wgetrc`-style command (COMMAND=robots=off).
- `--input-file (-i)`: download URLs found in local or external mp3_sites.txt.
The Tutorons Project

Design Patterns
Implementation Patterns

for generating in-browser, context-relevant programming documentation
### Percentage of tasks where participants accessed external documentation (n = 9)

<table>
<thead>
<tr>
<th>Language</th>
<th>N</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>wget Unix command</td>
<td>203</td>
<td>95%</td>
<td>64%</td>
</tr>
<tr>
<td>CSS selectors</td>
<td>466</td>
<td>80%</td>
<td>41%</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>445</td>
<td>70%</td>
<td>14%</td>
</tr>
</tbody>
</table>

Without Tutorons: 64%
With Tutorons: 6%
This Talk

- Background
- Interacting with Tutorons
- Developing Tutorons
Over a two-hour programming session, participants spent an average 19% of their programming time on the web. 

(Brandt et al. 2009)

Programmers frequently ask questions like "What is the functionality of a given API type?"

(Duala-Ekoko & Robillard 2012)
Demonstrating Code

WebCrystal [Chang & Myers '12]

PythonTutor [Guo '13]

FluidEdt [Ou et al. '15]

AutomataTutor [D’Antoni et al. '15]
Demonstrating Code

WebCrystal [Chang & Myers '12]

PythonTutor [Guo '13]

FluidEdt [Ou et al. '15]

AutomataTutor [D'Antoni et al. '15]
Explaining Code

```
fileName = showFileDialog(...);
"show file dialog and get file name"
```

Java methods (Sridhara et al. 2010)

```
onlineBox.add(f);
onlineBox.add(s);
onlineBox.add(t);
"add given components to oneline box"
```

Action sequences (Sridhara et al. 2011)
Explaining Code

- Class diagrams (Burden & Heldal 2011)
- Parameters (Sridhara et al. 2011)
- Classes (Moreno et al. 2013)
- Unit test cases (Kamimura et al. 2014)
- Method context (McBurney & McMillan 2014)

...
This Talk

- Background
- Interacting with Tutorons
- Developing Tutorons
Tutoron. a routine on a web server with language-specific rules for detecting, parsing and explaining source code written on a web page.
You found a `wget` command.

`wget` is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file `/tmp/download.txt`.

It uses these options:
- `--continue (-c)`: resume getting a partially-downloaded file.
- `--background (-b)`: go to background after startup.
- `--output-file (-o)`: log messages to `/tmp/download.log`.
- `--input-file (-i)`: download URLs found in local or external `/tmp/download.txt`.
- `--quota (-Q)`: set retrieval quota to 100m.

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>(null)</td>
</tr>
<tr>
<td>b</td>
<td>(null)</td>
</tr>
<tr>
<td>o</td>
<td><code>/tmp/download.log</code></td>
</tr>
<tr>
<td>i</td>
<td><code>/tmp/download.txt</code></td>
</tr>
<tr>
<td>quota</td>
<td>100m</td>
</tr>
</tbody>
</table>

Use `wget` with the password protected sites:

You can supply the http username/password on server as follows:

```
$ wget --user=username --password=password http://cyberciti.biz/download.tar.gz
```

Another way to specify username and password is in the URL itself:

```
$ wget http://username:password@cyberciti.biz/file.tar.gz
```
wget Unix command
CSS selectors
Regular expressions
And perhaps in the future:
LaTeX, Apache configs, SQL, etc.
Interacting with Tutorons

Programmer

Web Browser
Tutorons Library or Addon

Navigate

In-Situ Help
I need files to be downloaded to /tmp/cron_test/. My wget code is

```
```

So is the

```
wget
```

You found a wget command.

wget is a Terminal command you run to download a page from the Internet. Here, it downloads content from http://math.stanford.edu/undergrad/.

Recursively scrape web pages linked from http://math.stanford.edu/undergrad/ of type "*.pdf".

It uses these options:

- `--random-wait`: wait from 0.5*WAIT...1.5*WAIT secs between retrievals.
- `--recursive (-r)`: specify recursive download.
- `--page-requisites (-p)`: get all images, etc. needed to display HTML page.
- `--no-directories (-nd)`: don’t create directories.
- `--execute (-e)`: execute a `wgetrc`-style command (COMMAND=robots=off).
- `--accept (-A)`: "*.pdf" is a comma-separated list of accepted extensions.
- `--user-agent (-U)`: identify as mozilla instead of Wget/VERSION.
I need files to be downloaded to /tmp/cron_test/. My wget code is

```
wget --recursive
```

So is the

```
wget
```

share.im

Recursively scrape web pages linked from http://math.stanford.edu/undergrad/ of type ".pdf".
It uses these options:

```
```

Context-Relevant Descriptions of Code's High-Level Intent

Recursively scrape web pages linked from http://math.stanford.edu/undergrad/ of type ".pdf"
I need files to be downloaded to /tmp/cron_test/. My wget code is

```
```

So I add: Recursively scrape web pages linked from http://math.stanford.edu/undergrad/ or type "pdf".

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--user-agent (-U)`: identify as mozilla instead of Wget/VERSION.
You found a CSS selector.

The selector '.content_container_7 button' chooses buttons from elements of class 'content_container_7'.

*If you haven’t seen them before*, selectors pick sections of HTML pages by their names or properties. Once you’ve ‘grabbed’ elements with a selector, you can manipulate them, like changing their appearance or text.

Here’s an example of what this selector will find:

```html
<div class="content_container_7">
  <button>
  
  </button>
</div>
```
The selector 'content_container_7 button' chooses buttons from elements of class 'content_container_7'.
The selector `'.content_container_7 button'` chooses buttons from elements of class `'content_container_7'`.

You found a CSS selector.

The selector `'.content_container_7 button'` chooses buttons from elements of class `'content_container_7'`.

*If you haven’t set any CSS properties, once the selector has been applied, the buttons would look exactly like any other button, with no additional styling.*

Here’s an example of what this selector will find:

```html
<div class='content_container_7'>
  <button>Button</button>
</div>
```

The selector `'.content_container_7 button'` chooses buttons from elements of class `'content_container_7'`.
Usage Examples

```html
<div class="content_container_7">
  <button>
    <button>
      </button>
  </button>
</div>
```
I tried using:

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^\[(www.)?domain.com$\]
RewriteRule ^//\%\$domain/ww_ \n```

You found a regular expression.

Regular expressions are patterns written to match classes of strings. By altering the pattern, you can match different types of strings, like URLs or e-mail addresses.

You can get a feeling for what this expression matches by reading the diagram from left to right:

The pattern `^\[(www.)?domain.com$\]` matches strings including:

```
wwwpdomain=com
domain\|com
```
You can get a feeling for what this expression matches by reading the diagram from left to right:

Start of line → “www” → any character → “domain” → any character → “com” → End of line

This question came from our site for professional and enthusiast programmers.
I tried using:

```
RewriteEngine On
RewriteCond %{HTTP_HOST} ^(www.)?domain.com$
RewriteRule ^//\S+domain/www/ [L]
```

Usage Examples

You found:

Regular expressions can match different types of strings.

You can get a feeling for what this expression matches by reading the diagram from left to right:

The pattern `^(www.)?domain.com$` matches strings including:

```
wwwpdomain=com  domain\|com
```

This question came from our site for professional and enthusiast programmers.
An Example of Synthesizing Help

```
div#summ a[href^=/video]
```

this selector chooses links with URLs starting with "/video" from a container with ID "summ"
1. Generate plaintext description of element

div #summ a[href^=/video]

div containers a links
2. Generate modifiers to describe IDs, attributes, etc.

- **a**
  - links
  - [href^=/video]
    - URLs starting with "/video"

- **div**
  - containers
  - #summ
    - ID "summ"
2. Generate modifiers to describe IDs, attributes, etc.

```html
<container with ID "summ">
  <a href="/video">
    links with URLs starting with "/video"
  </a>
</container>
```
3. Append descriptions of parents to children with "from" modifiers

   a[href^=/video]
   links with URLs starting with "/video"

   div#summ
   a container with ID "summ"
3. Append descriptions of parents to children with "from" modifiers.

```
div#summ a[href^=/video]
```

links with URLs starting with "/video" from a container with ID "summ"
3. Append descriptions of parents to children with "from" modifiers

```
div#summ a[href^=/video]
```

this selector chooses links with URLs starting with "/video" from a container with ID "summ"
Interacting with Tutorons

Web Browser
Tutorons Library or Addon

Programmer

Navigate
In-Situ Help
Then use the following `wget` command to go out and fetch those MP3's:

```
wget -r -1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```
form.cmxfm form fieldset {
  margin-bottom: 10px;
}

form.cmxfm legend {
  padding: 0 2px;
  font-weight: bold;
}

form.cmxfm label {
  display: inline-block;
  line-height: 1.8;
  vertical-align: top;
}
Try this code: $('my-par').text('');

The CSS selector

Try this code: $('my-par').text('');

The CSS selector with quotes

Try this code: $('my-par').text('');

The jQuery selection

Try this code: $('my-par').text('');

A sloppy selection
Recursively scrape web pages of type '.mp3' from URLs from the file 'mp3_sites.txt', following links 1 time.

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.

```bash
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```
You found a `wget` command.

`wget` is a Terminal command you run to download a page from the Internet. Here, it downloads content from URLs from the file 'mp3_sites.txt'.

Recursively scrape web pages linked from URLs from the file 'mp3_sites.txt' of type '.mp3', following links 1 times.

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.
- `--tries (-t)`: set number of retries to 1 (0 unlimits).
- `--no-directories (-nd)`: don't create directories.
- `--timestamping (-N)`: don't re-retrieve files unless newer than local.

```
wget -r -l=1 -H -nd -N -np -X.mp3 -erobots-url -U mp3_sites.txt
```
Experimental Setup

9 programmers

8 wget / CSS modification tasks

Controlled access to Tutorons

Measured access to external docs
Step 1: Read This Task

Write a CSS selector that selects only elements of class `myInput`.
Step 1: Read This Task

Write a CSS selector that selects only elements of class `myInput`.

Step 2: Look at the Snippet for Clues

Read the snippet below to find code you can modify to do the task. Small fragments of these snippets can be used to do your task. Ask the proctor if you should view accelerators. Afterwards, you can use any other strategies or information you want.

```javascript
Use:

```357
$(".myCheckbox").attr('checked', true); // Deprecated
```

You found a CSS selector.
The selector `.myCheckbox` chooses elements of class `myCheckbox`.

*If you haven’t seen them before, selectors pick sections of HTML pages by their names or properties. Once you’ve ‘grabbed’ elements with a selector, you can manipulate them, like changing their appearance or text.*

Here’s an example of what this selector will find:

```html
<div class="myCheckbox">
</div>
```
Percentage of tasks where participants accessed external documentation

Without Explanations: 64%

With Explanations: 6%

$p < .0001$ (Fisher's Exact Test, $n = 9$)
Limitations

• Small number of participants (n = 9)

• Tasks were designed such that the Tutorons would function properly

• Participants may have been more patient with explanation content than if they were on their own
Percentage of tasks where participants accessed external documentation

Without Explanations: 64%  
With Explanations: 6%

$p < .0001$ (Fisher's Exact Test, $n = 9$)
This Talk

- Background
- Interacting with Tutorons
- Developing Tutorons
Detecting Code Snippets

HTML

HTML Nodes

Code Regions

Filtered

mobileinit
page2
body

body
Parsing Code Examples

Parse Structures

CSS Selectors

wget

Table of arguments

<table>
<thead>
<tr>
<th>Short name</th>
<th>Long name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>recursive</td>
<td>(null)</td>
</tr>
<tr>
<td>l</td>
<td>level</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Generating Explanations and Examples

\[ \text{div#summ a[href^=/video]} \]

this selector chooses links with URLs starting with "/video" from a container with ID "summ"
Counting to Discover
Common Usage

wget --random-wait -r -p -nd -e robots=off
A".pdf" -U mozilla
http://math.stanford.edu/undergrad/

Recursively scrape web pages linked from
http://math.stanford.edu/undergrad/
of type ".pdf"
wget -r -nH --cut-dirs=2 --no-parent --reject="index.html*" http://mysite.com/dir1/dir2/data

wget -r --no-parent http://mysite.com/configs/.vim/

wget -r -nH --cut-dirs=2 --no-parent --reject="index.html*" http://mysite.com/dir1/dir2/data

Options: -r, -nH, --cut-dirs, --no-parent, --reject

wget -r --no-parent http://mysite.com/configs/.vim/

Options: -r, --no-parent


Options: -r, --no-parent, --reject
Options: -r, -nH, --cut-dirs, --no-parent, --reject

Options: -r, --no-parent

Options: -r, --no-parent, --reject
<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>--no-parent</td>
<td>3</td>
</tr>
<tr>
<td>-r</td>
<td>--reject</td>
<td>2</td>
</tr>
<tr>
<td>-r</td>
<td>-nH</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
# Counting to Discover Common Usage

<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>-A</td>
<td>28</td>
</tr>
<tr>
<td>--user</td>
<td>--password</td>
<td>23</td>
</tr>
<tr>
<td>-r</td>
<td>-l</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Counting to Discover Common Usage

<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>-A</td>
<td>28</td>
</tr>
<tr>
<td>--user</td>
<td>--password</td>
<td>23</td>
</tr>
<tr>
<td>-r</td>
<td>-l</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Testing Detection Quality

50 tutorials / language

```
wget -r -nH --cut-dirs=2 --no-parent --reject="index.html*" http://mysite.com/dir1/dir2/data
wget -r --no-parent http://mysite.com/configs/.vim/
```

200-500 snippets / language
# Detection Accuracy

<table>
<thead>
<tr>
<th>Language</th>
<th>N</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>wget Unix command</td>
<td>203</td>
<td>95%</td>
<td>64%</td>
</tr>
<tr>
<td>CSS selectors</td>
<td>466</td>
<td>80%</td>
<td>41%</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>445</td>
<td>70%</td>
<td>14%</td>
</tr>
</tbody>
</table>
## Detection Accuracy

<table>
<thead>
<tr>
<th>Language</th>
<th>N</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>wget Unix command</td>
<td>203</td>
<td>95%</td>
<td>64%</td>
</tr>
<tr>
<td>CSS selectors</td>
<td>466</td>
<td>80%</td>
<td>41%</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>445</td>
<td>70%</td>
<td>14%</td>
</tr>
</tbody>
</table>
It's Hard to Find Code Within Text
Regular expressions must be found in many languages
Percentage of tasks where participants accessed external documentation

Without Explanations: 64%
With Explanations: 6%

\( p < .0001 \) (Fisher's Exact Test, \( n = 9 \))

Detection Accuracy

<table>
<thead>
<tr>
<th>Language</th>
<th>N</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>wget Unix command</td>
<td>203</td>
<td>95%</td>
<td>64%</td>
</tr>
<tr>
<td>CSS selectors</td>
<td>466</td>
<td>80%</td>
<td>41%</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>445</td>
<td>70%</td>
<td>14%</td>
</tr>
</tbody>
</table>
Recursively scrape web pages of type '.mp3' from URLs from the file 'mp3_sites.txt', following links 1 time.

It uses these options:
- --recursive (-r): specify recursive download.
- --level (-l): 1 is a maximum recursion depth (inf or 0 for infinite).
- --span-hosts (-H): go to foreign hosts when recursive.

```
wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt
```
Recursively scrape web pages of type '.mp3' from URLs from the file 'mp3_sites.txt', following links 1 time.

It uses these options:

- `--recursive (-r)`: specify recursive download.
- `--level (-l)`: 1 is a maximum recursion depth (inf or 0 for infinite).
- `--span-hosts (-H)`: go to foreign hosts when recursive.

VS.

`wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off -i mp3_sites.txt`
Tutorons

Design Patterns
Implementation Patterns

for generating in-browser, context-relevant programming documentation
www.tutorons.com
Tutorons

var conclusion = /Thanks, (VL|HCC) 2015!/g;

Andrew Head, Codanda Appachu, Marti A. Hearst, Björn Hartmann
Computer Science Division, UC Berkeley