

candour

Robots.txt introduction

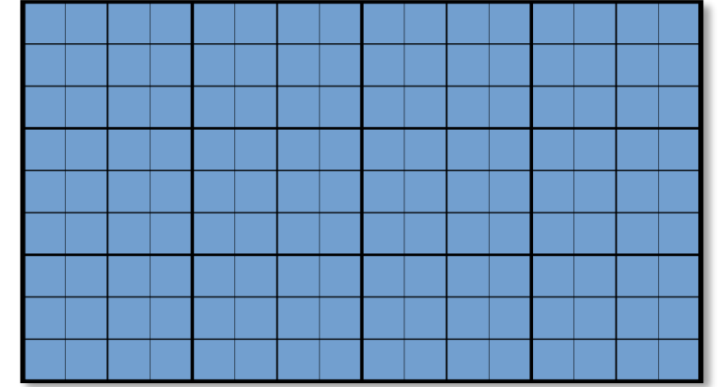


Trainer:

Mark Williams-Cook
Candour

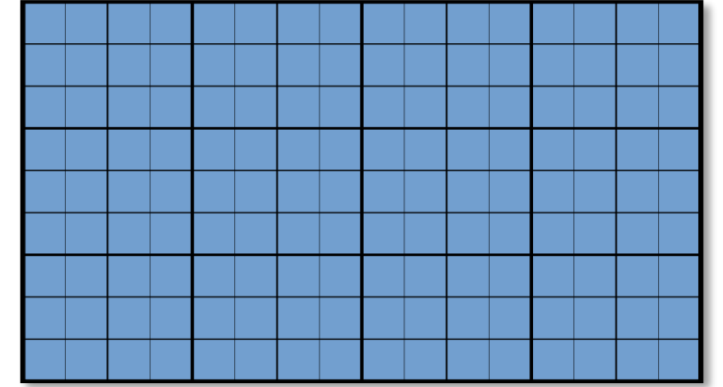


What is it?

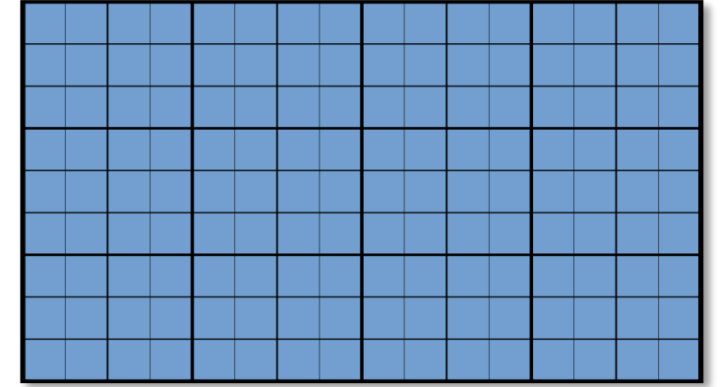


What is it?

Robots.txt is a text file that instructs robots which pages or files they are allowed to **crawl** on your website.

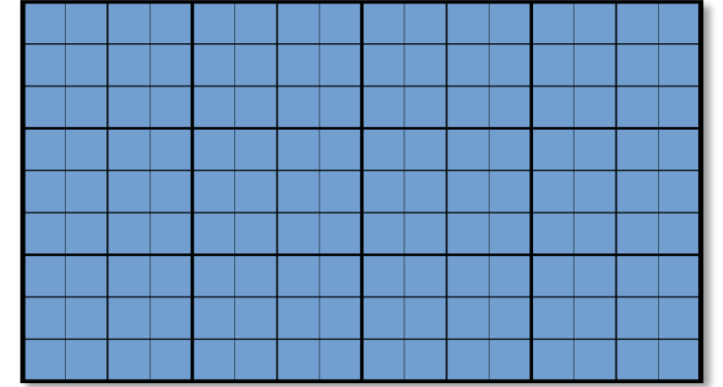


What it is not!



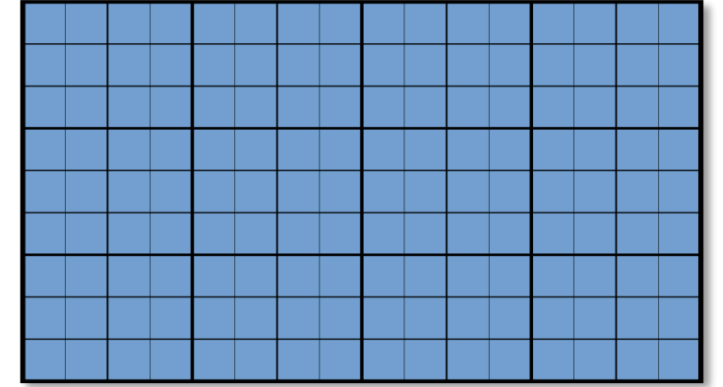
What it is not!

It is not a reliable mechanism to stop **pages** becoming **indexed** with Google or other search engines!

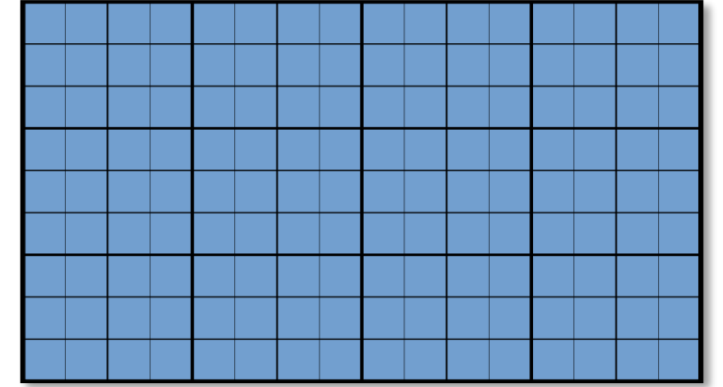


Key concept

Robots.txt should not be used to try and stop web pages from appearing within Google search results

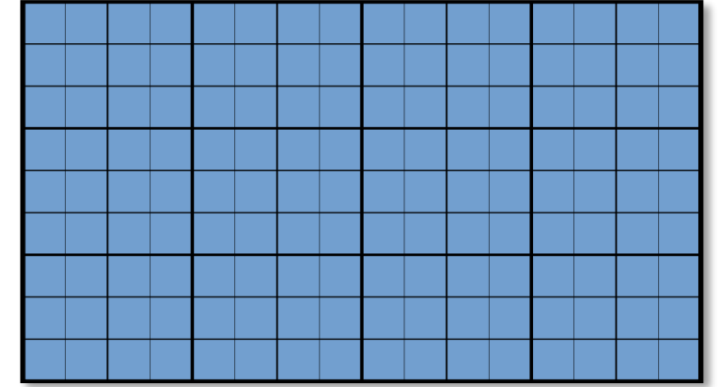


What is it for?



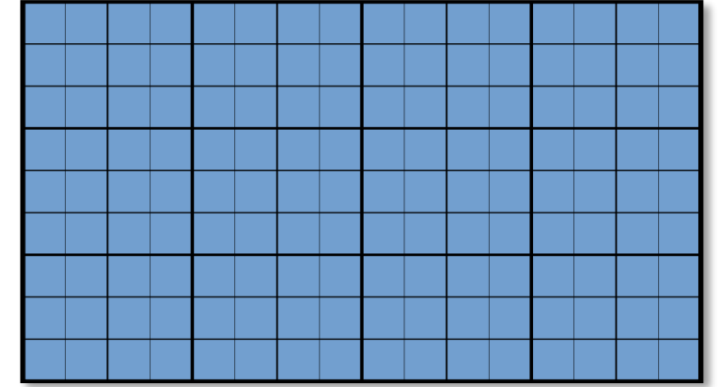
What is it for?

- 1) It can be used to manage how robots crawl your site for traffic management and on larger sites, 'crawl budget' management



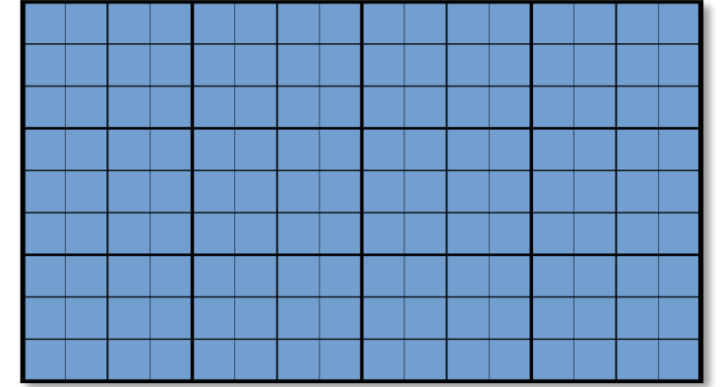
What is it for?

- 1) It can be used to manage how robots crawl your site for traffic management and on larger sites, 'crawl budget' management
- 2) It can stop search engines accessing media and resources files such as images.



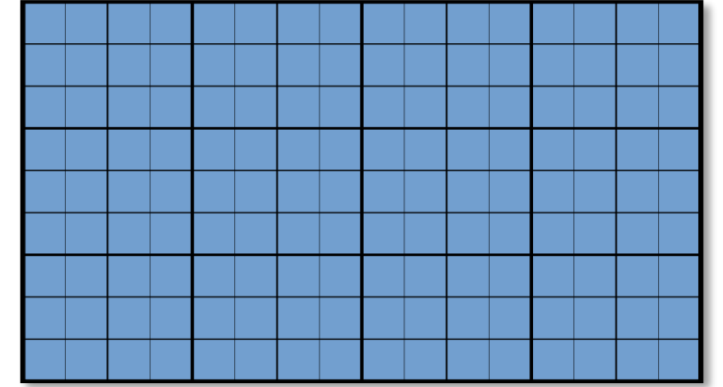
What is it for?

- 1) It can be used to manage how robots crawl your site for traffic management and on larger sites, 'crawl budget' management
- 2) It can stop search engines accessing media and resources files such as images.
- 3) You can define where your sitemap(s) are



Where is it?

www.yourwebsite.com/robots.txt



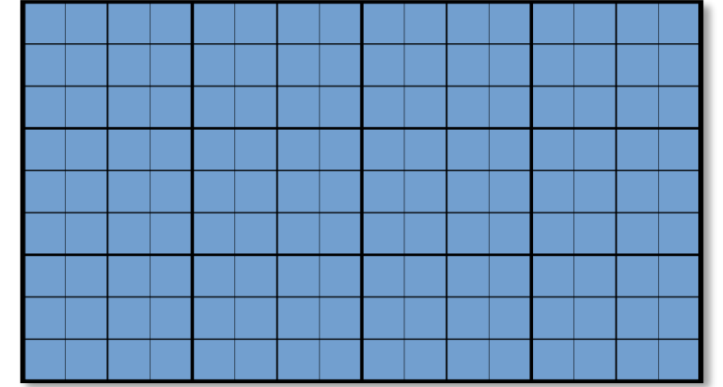
Challenge



See if you can locate your robots.txt file by visiting your website with /robots.txt on the end.

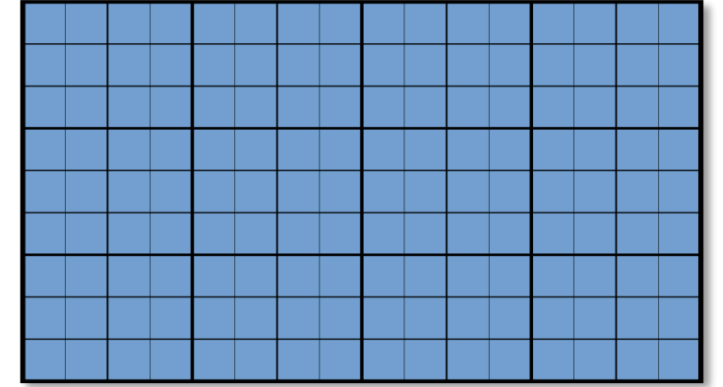
Don't worry if you can't find it – not all websites have one!

What's in a robots.txt?



What's in a robots.txt?

Robots.txt specifies which files and folders can be accessed by usage of the **Allow** and **Disallow** commands.

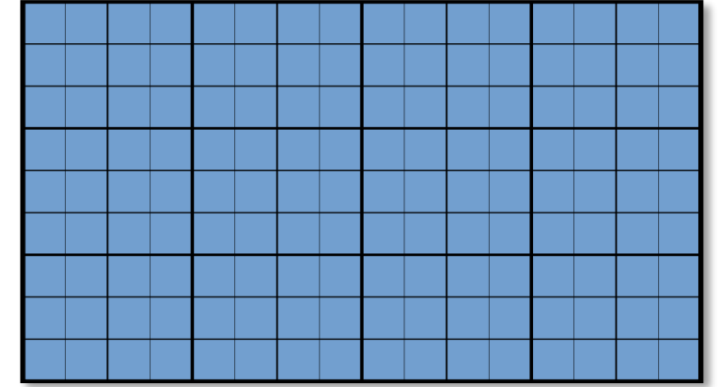


What's in a robots.txt?

Robots.txt specifies which files and folders can be accessed by usage of the **Allow** and **Disallow** commands.

Allow: /yesplease/

Disallow: /nothankyou/



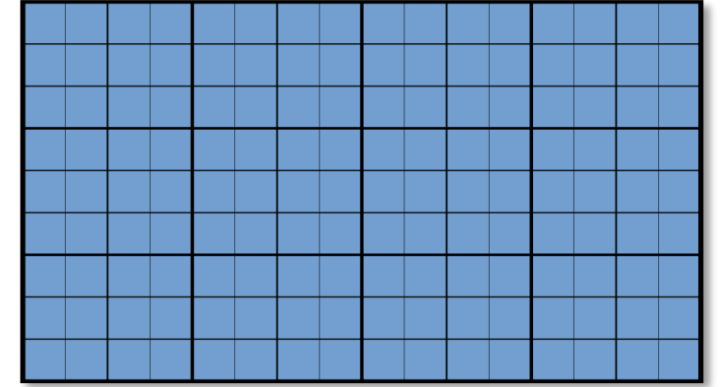
What's in a robots.txt?

You can specify which robots this rules apply to by using the **User-agent** command.

User-agent: Googlebot

Allow: /yesplease/

Disallow: /nothankyou/



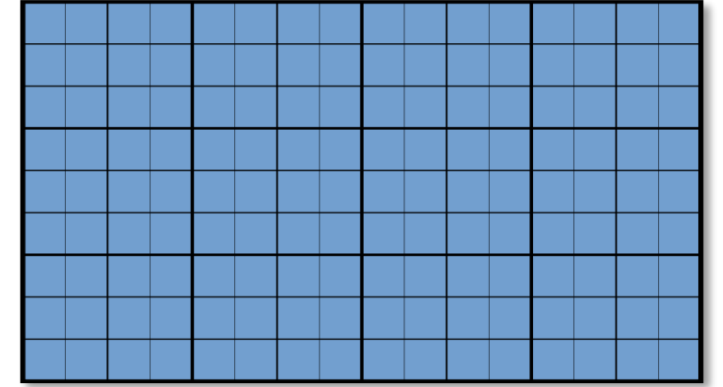
What's in a robots.txt?

You can specify which robots this rules apply to by using the **User-agent** command.

User-agent: *

Allow: /yesplease/

Disallow: /nothankyou/



What's in a robots.txt?

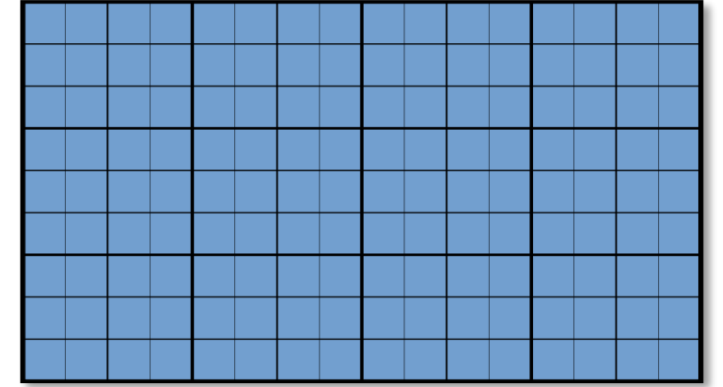
You can specify which robots this rules apply to by using the **User-agent** command.

User-agent: *

Disallow: /googleonly/

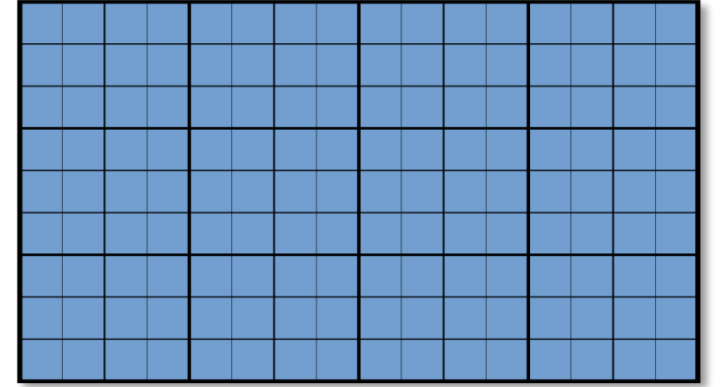
User-agent: Googlebot

Allow: /googleonly/



Some examples

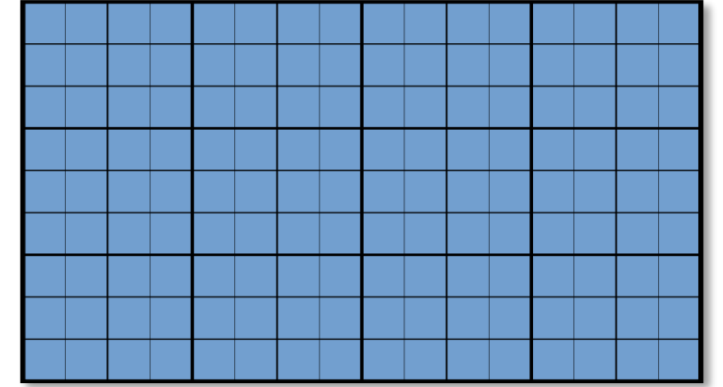
Disallow crawling by all robots to all of your website



Some examples

Disallow crawling by all robots to all of your website

User-agent: *

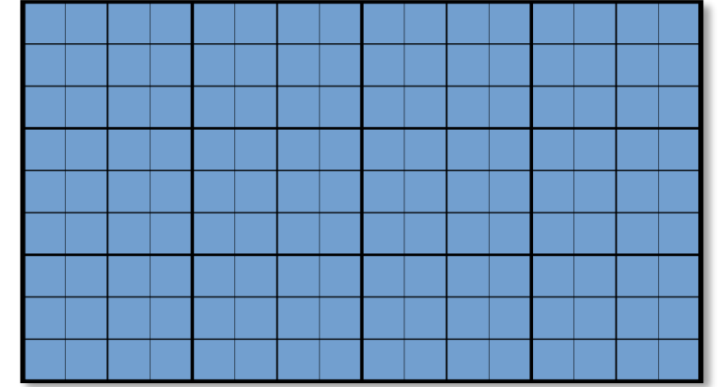


Some examples

Disallow crawling by all robots to all of your website

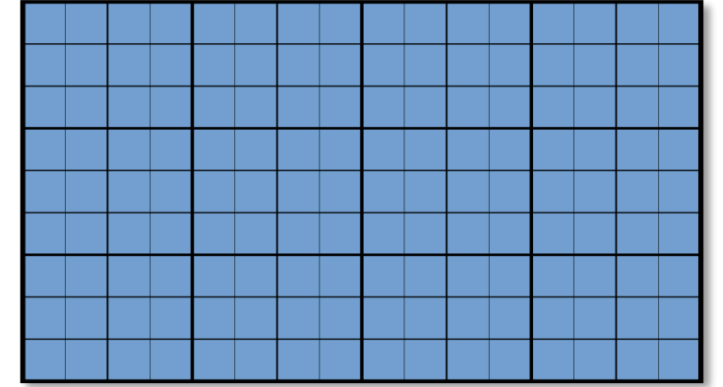
User-agent: *

Disallow: /



Key concept

Always check your robots.txt file as soon as you “go live” with a new website!



Some examples

/fish

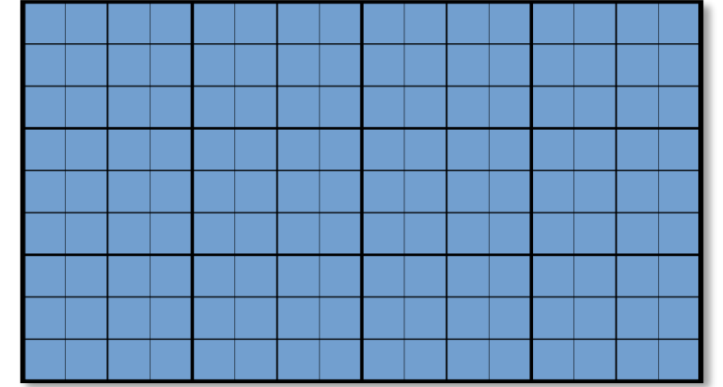
👍 Matches:

- /fish
- /fish.html
- /fish/salmon.html
- /fishheads
- /fishheads/yummy.html
- /fish.php?id=anything

👎 Does not match:

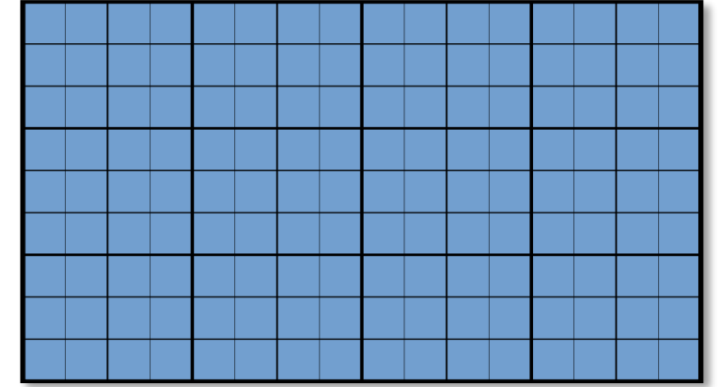
- /Fish.asp
- /catfish
- /?id=fish

★ Note the case-sensitive matching.



Wildcards

- * Matches 'any' occurrence of a character
- \$ Designates the end of the URL



Some examples

/fish

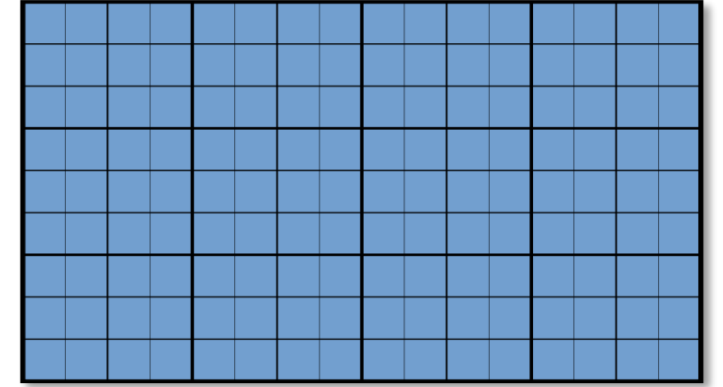
👍 Matches:

- /fish
- /fish.html
- /fish/salmon.html
- /fishheads
- /fishheads/yummy.html
- /fish.php?id=anything

👎 Does not match:

- /Fish.asp
- /catfish
- /?id=fish

★ Note the case-sensitive matching.



Some examples

/fish*

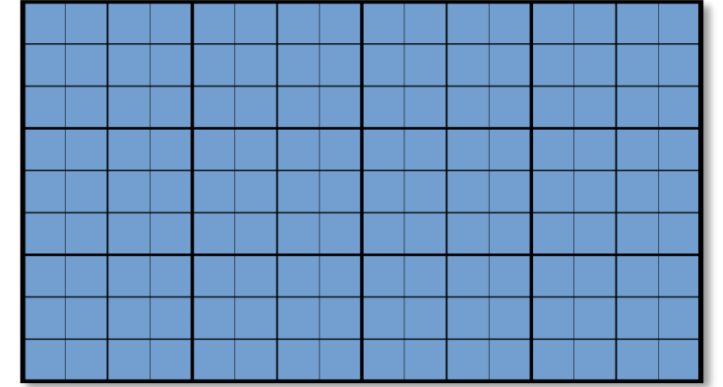
👍 Matches:

- /fish
- /fish.html
- /fish/salmon.html
- /fishheads
- /fishheads/yummy.html
- /fish.php?id=anything

👎 Does not match:

- /Fish.asp
- /catfish
- /?id=fish

★ Note the case-sensitive matching.




Some examples

`/fish/`

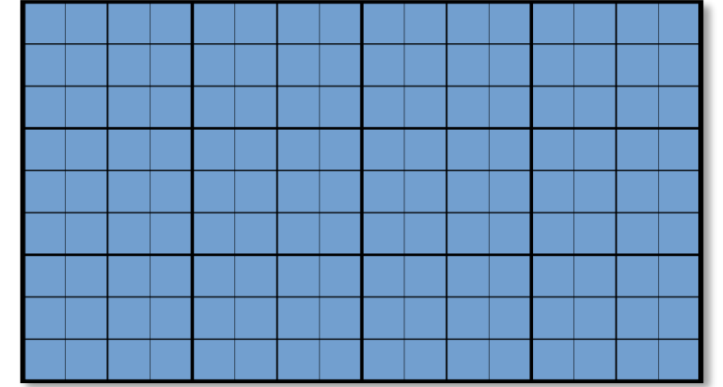
The trailing slash means this matches anything in this folder.

 Matches:

- `/fish/`
- `/fish/?id=anything`
- `/fish/salmon.htm`

 Does not match:

- `/fish`
- `/fish.html`
- `/Fish/Salmon.asp`




Some examples

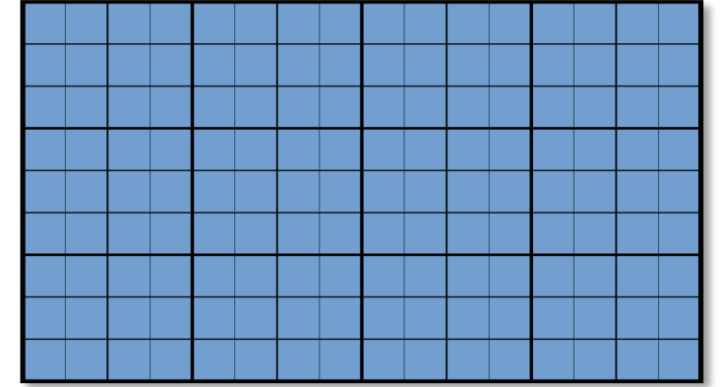
`/*.php`

 Matches:

- `/filename.php`
- `/folder/filename.php`
- `/folder/filename.php?parameters`
- `/folder/any.php.file.html`
- `/filename.php/`

 Does not match:

- `/` (even if it maps to `/index.php`)
- `/windows.PHP`



Some examples

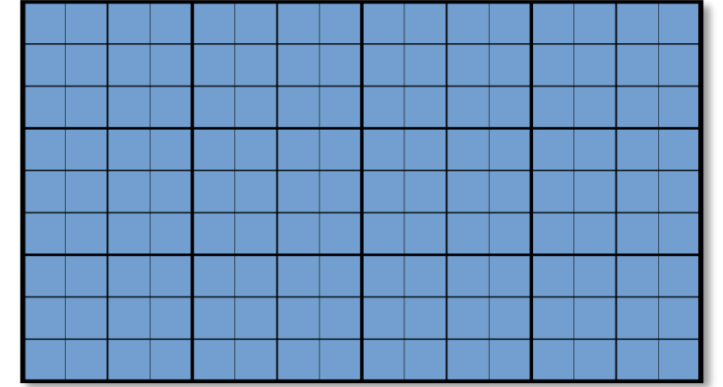
```
/*.php$
```

👍 Matches:

- /filename.php
- /folder/filename.php

👎 Does not match:

- /filename.php?parameters
- /filename.php/
- /filename.php5
- /windows.PHP



Challenge



Use the link in the lecture resources to look at Google's Robots.txt Specifications and take the time to go through the tables of examples they have there.

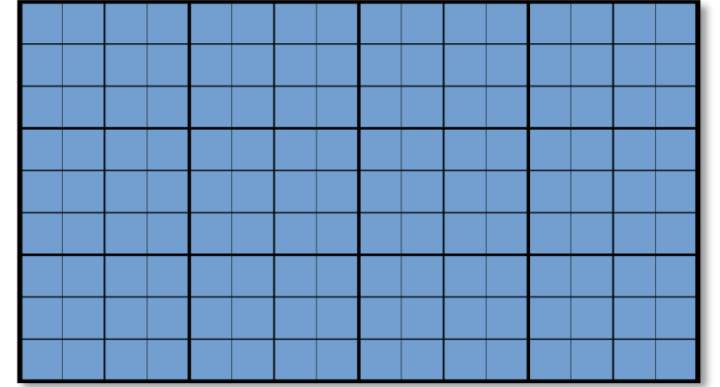
Some examples

What do we think would happen here?

User-agent: *

Disallow: /

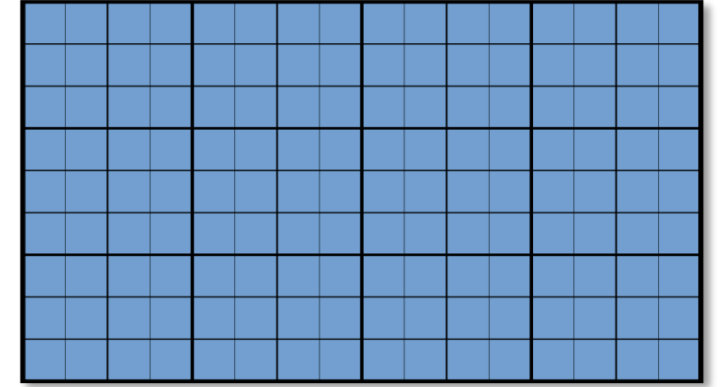
Allow: /public



Key concept

The most specific instruction takes precedence.

In the case of conflicting rules, the least restrictive rule is used.



Challenge



What would happen with this robots.txt file?

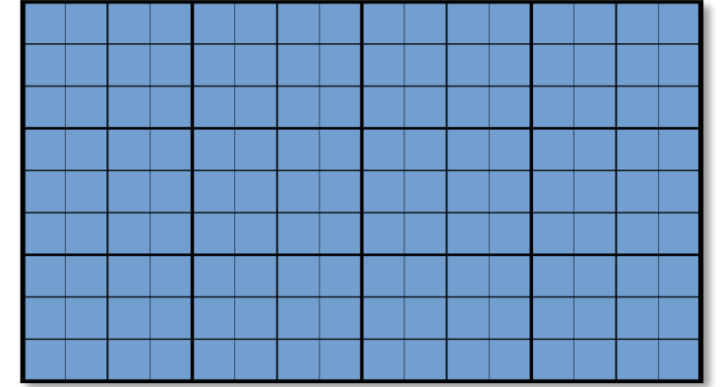
User-agent: *

Disallow: /folder

Allow: /folder

Answer

/folder would be crawled because the “Allow” is the least restrictive rule.



Key concept

Some “off the shelf” platforms such as Shopify and Wix do not allow you to control or edit robots.txt file entries.

