# regex no character

Ask Question

Create value in a global mar
mercari    anyone can buy & sell

In this text:

▲

3

▼

★

```
warning here
there are several types of warnings
in this string warning.gif
at the end warning
end of line warning
```

I want to match every `warning` except `warning.gif` .

I cannot seem to get it to include the last one that has no character (control or otherwise) after it.

using `warnings?[^\.]` gets what I want except for the last warning (on the last line). I think because there is no character at all after. How can I get it to include that one?

regex

edited Jun 1 '10 at 21:52
Mark Byers
**600k**   129   1371
1349

asked Jun 1 '10 at 21:46
iamnotmad
**170**   1   2   8

1    What language are you using? Not all regular expression engines are equally powerful, and the syntax differs between engines. – Mark Byers Jun 1 '10 at 21:50

## 2 Answers

▲      If you have negative lookaheads you can use that:

This should work in Javascript.

edited Jun 1 '10 at 22:24

answered Jun 1 '10 at 21:49

**Mark Byers**

**600k**    129    1371
1349

I'm working in javascript regex. –
iamnotmad    Jun 1 '10 at 21:50

Good news: Javascript has negative
lookaheads. – Mark Byers Jun 1 '10
at 22:23

This is the winner! Thanks a lot! I
changed the ".gif" to just "." –
iamnotmad    Jun 2 '10 at 12:42

Hi Mark, I'd like to add something to
this. I want to NOT include warning
where it is preceded by "0 " (zero
space), but yes if any other number
>0. Lastly do not include if there is
any other character after warning or
warnings except a space/end of line
(or I think in my case nothing (no eol
control chars). Can you help with that
one? –    iamnotmad    Jun 2 '10 at
13:26 ✏

I got this to work with other chars
besides . after. now it looks like:
warnings?(?!\w) Now I just need to
make it not include warning where
there is a "0 " <zero space> in front,
but any other number ok, or nothing.
–    iamnotmad    Jun 2 '10 at 14:23

▲

**EDITED**

1

I completely misread the question. My
apologies.

Here's a corrected solution:

Alternatives can bind more tightly
than the eof marker.

▼

Example file:

```
warning here
 there are several types of warning
```

Example script:

```
awk '/warnings?$|warnings?[[:space
```

Original (wrong) answer:

*Use:* `/warnings?$/m` *- note that the "?" means that the "s" is optional*

[edited Jun 2 '10 at 13:24](#)

answered Jun 1 '10 at 21:51

[geocar](#)
**7,997**   1   24   36

---

for some reason this did not catch the warning on the first line. –
[iamnotmad](#)   Jun 1 '10 at 21:54

---

@iamnotmad: You have something else wrong with your program. I've edited my post to include an example which you can try yourself. – [geocar](#) Jun 1 '10 at 22:08

---

This regex only matches if the word is at the end of the line, which is not always the case. – [Alan Moore](#) Jun 2 '10 at 2:14

---

the new one works. I'd just like to exclude warning where there is a "0 " <zero space> in front of it. –
[iamnotmad](#)   Jun 2 '10 at 14:16

---

@iamnotmad: Then use:

```
/^warnings?$|
[[:space:]]warnings?
$|^warnings?[[:space:]]|
[[:space:]]warnings?
[[:space:]]/
```
 – [geocar](#) Jun 2 '10 at 14:26