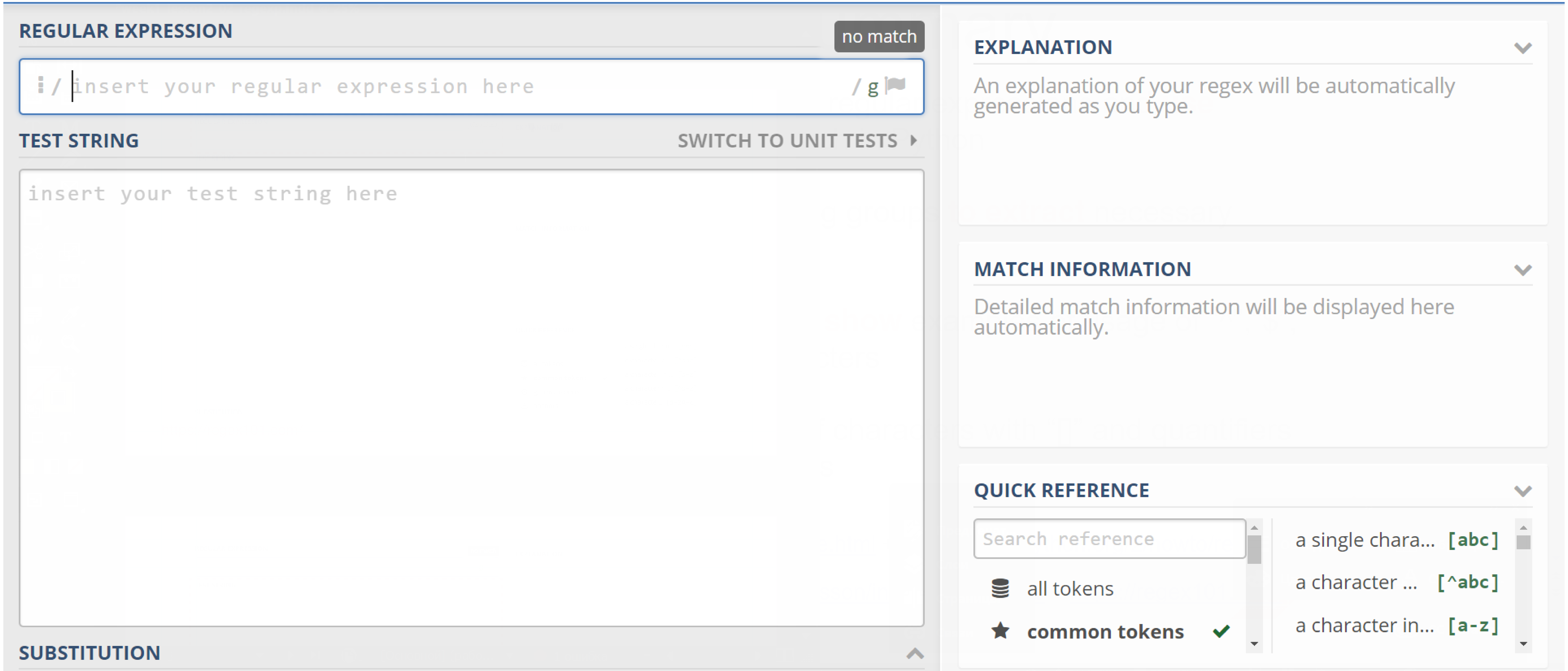
Regular Expressions

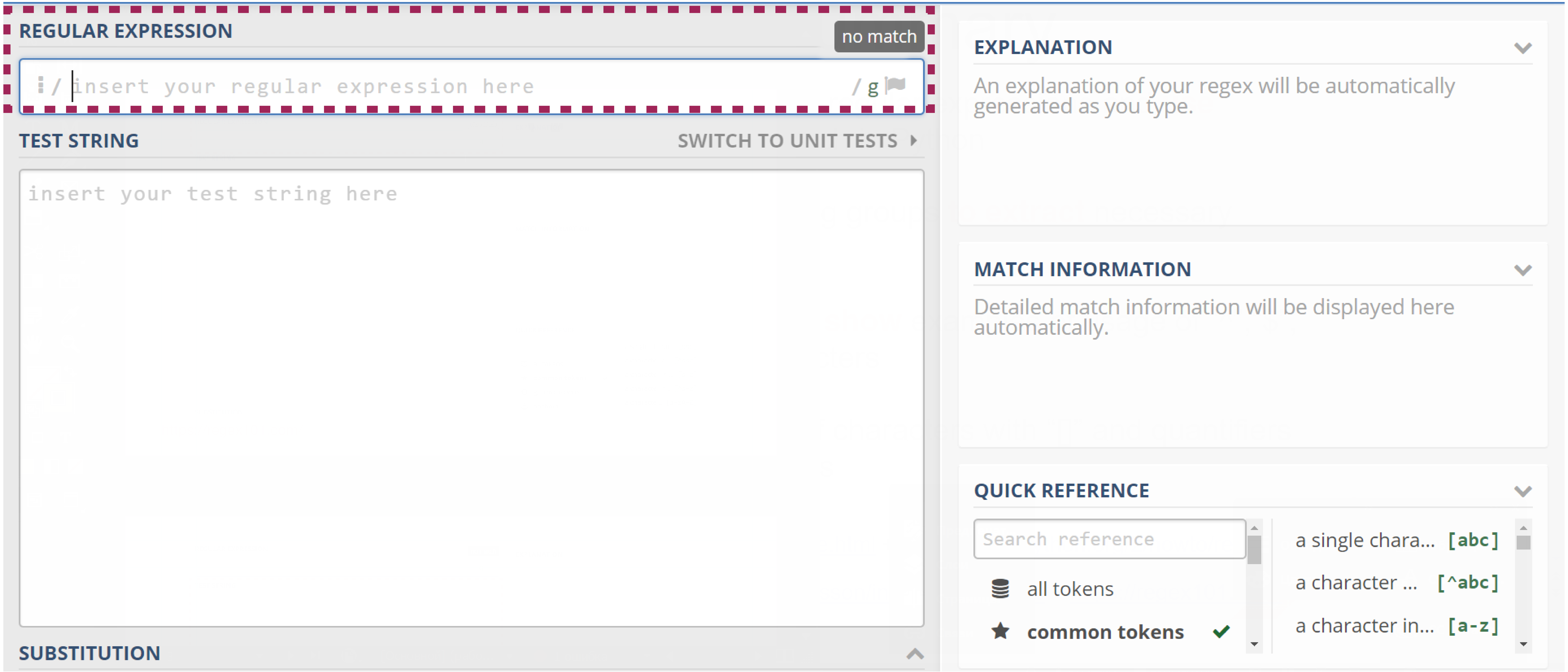
Likbez



An explanation of your regex will be automatically generated as you type.

Detailed match information will be displayed here automatically.

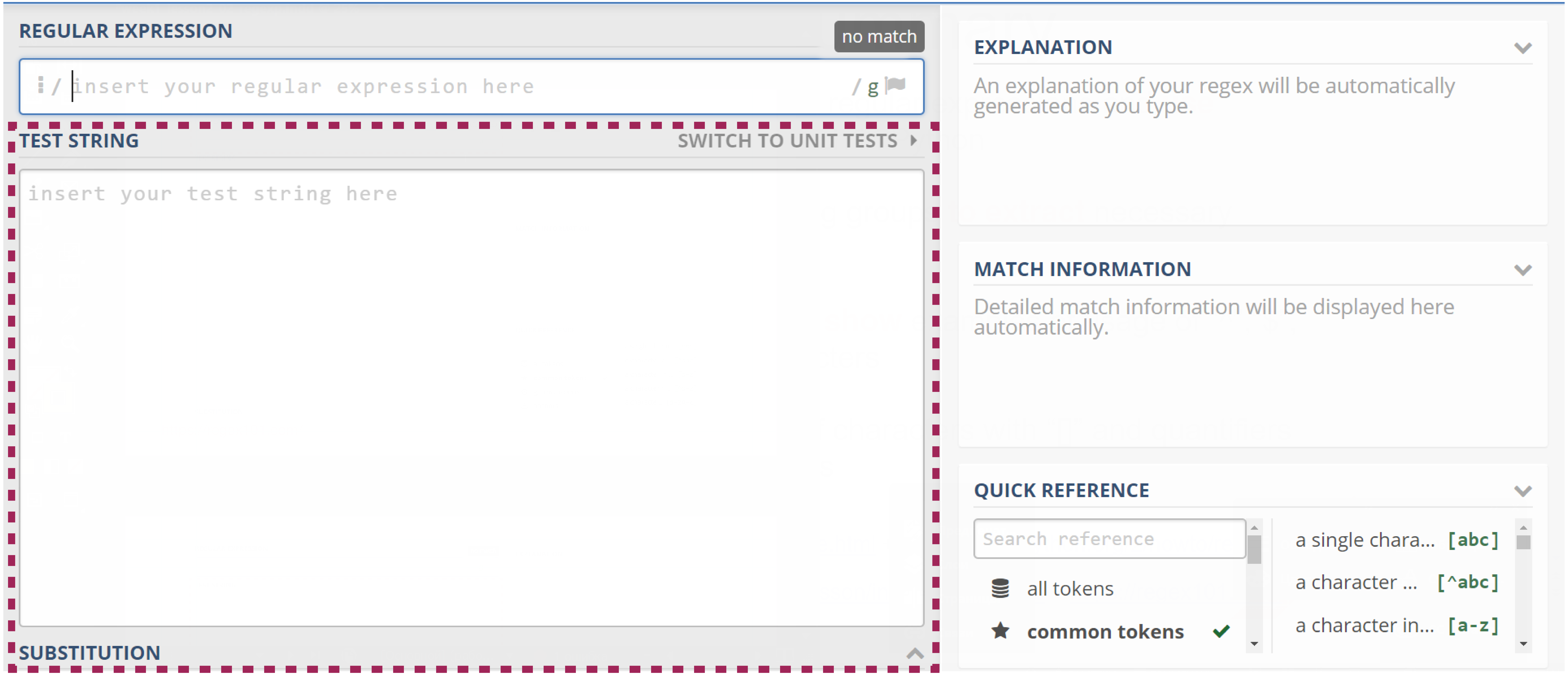
https://regex101.com/



An explanation of your regex will be automatically generated as you type.

Detailed match information will be displayed here automatically.

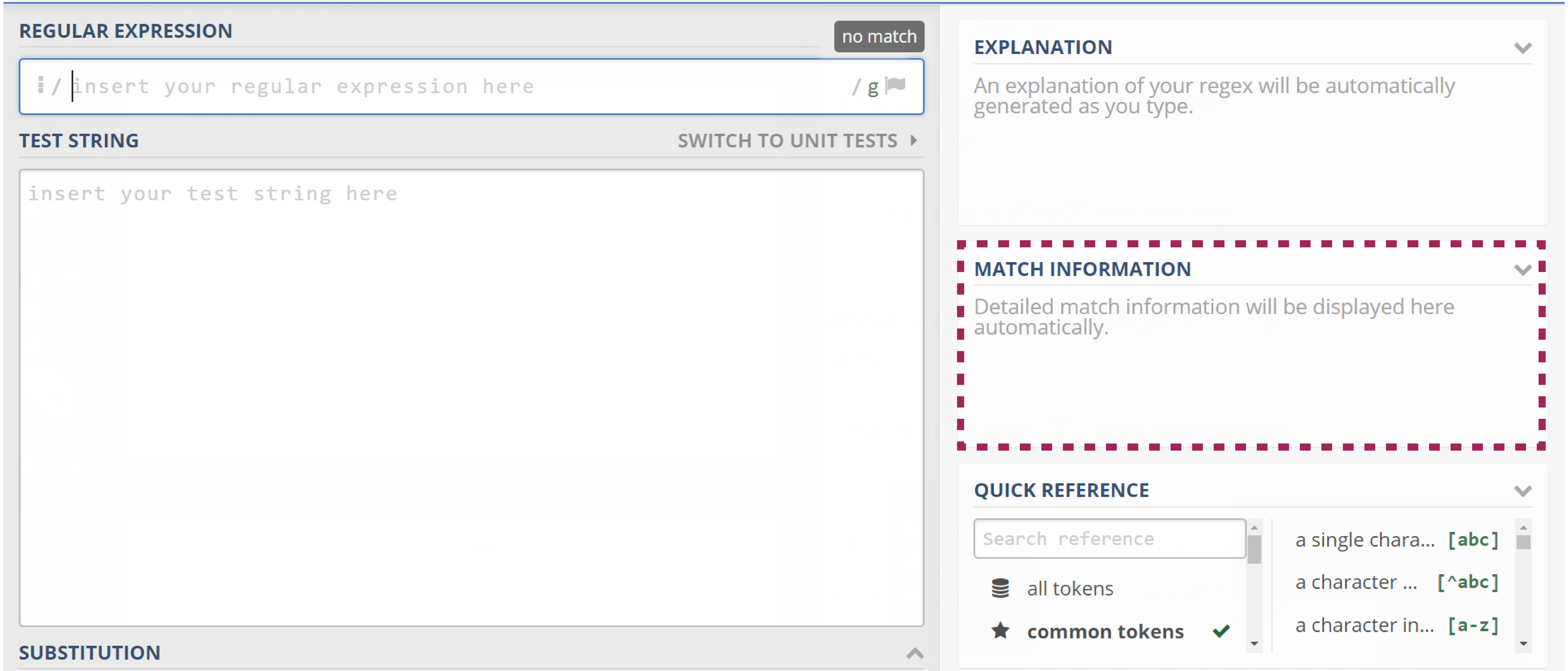
https://regex101.com/



An explanation of your regex will be automatically generated as you type.

Detailed match information will be displayed here automatically.

https://regex101.com/



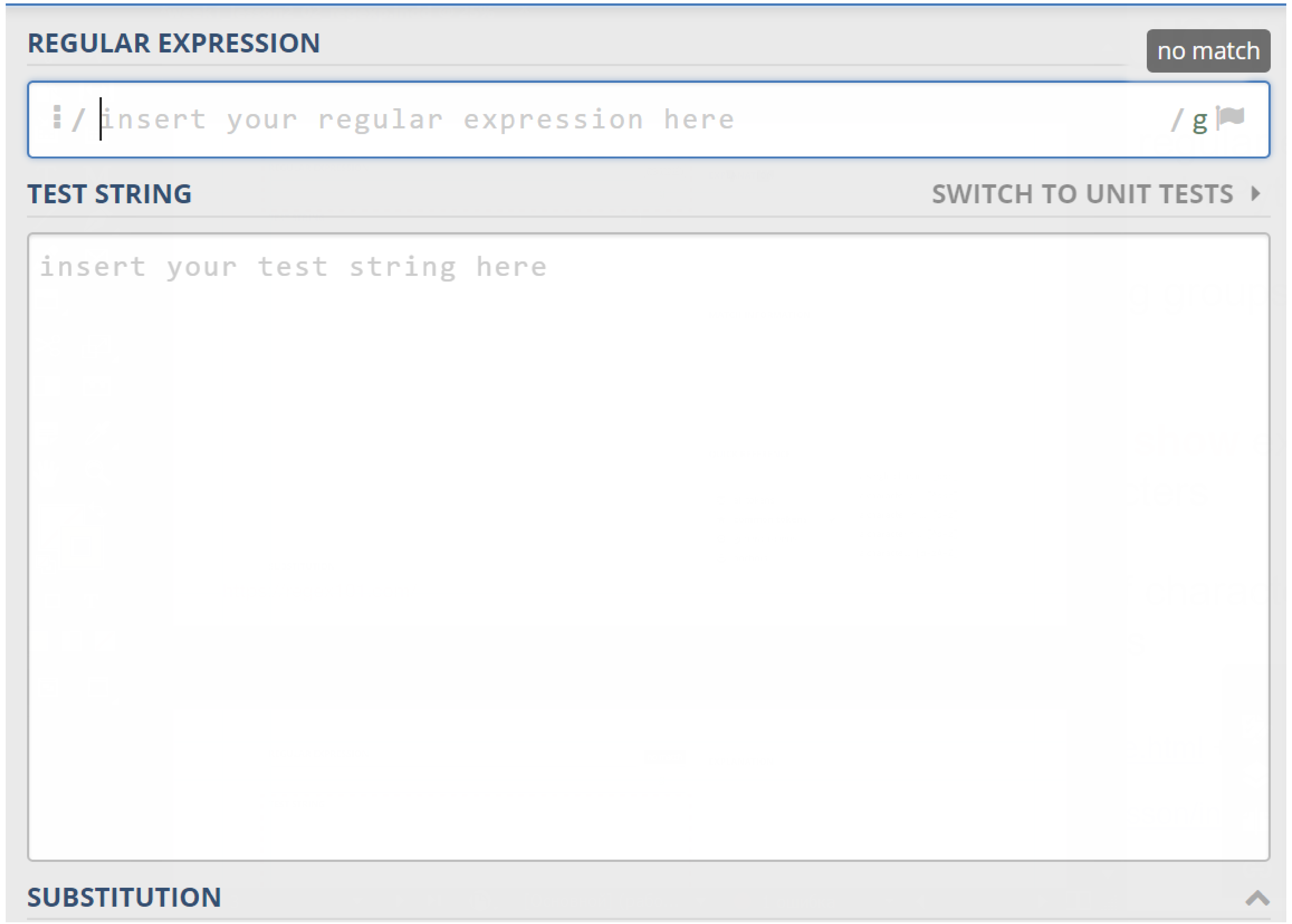
An explanation of your regex will be automatically generated as you type.

Detailed match information will be displayed here automatically.

https://regex101.com/

online

offline



REGULAR EXPRESSION

no match

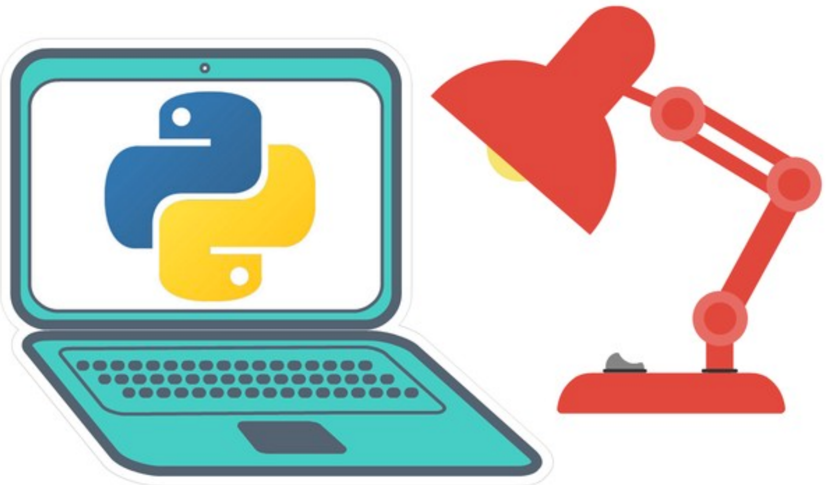
/ insert your regular expression here / g

TEST STRING

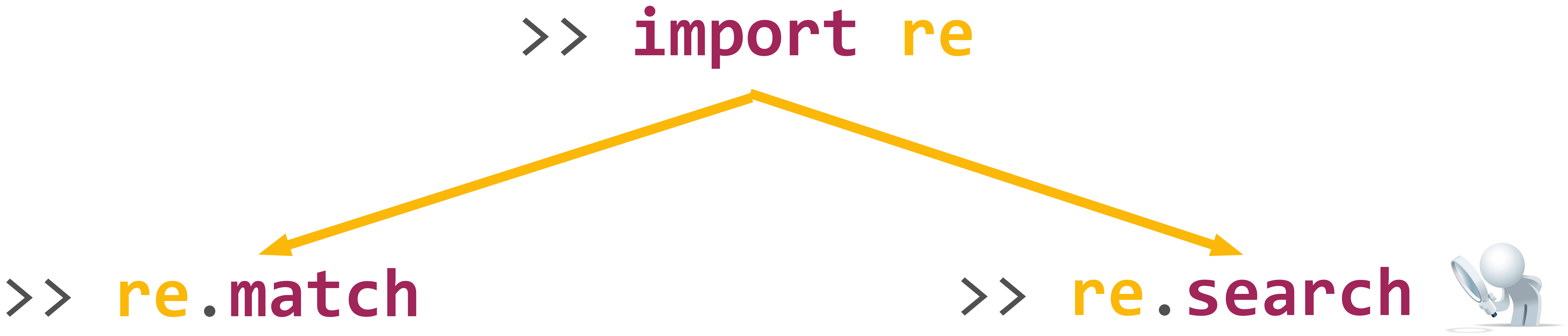
SWITCH TO UNIT TESTS

insert your test string here

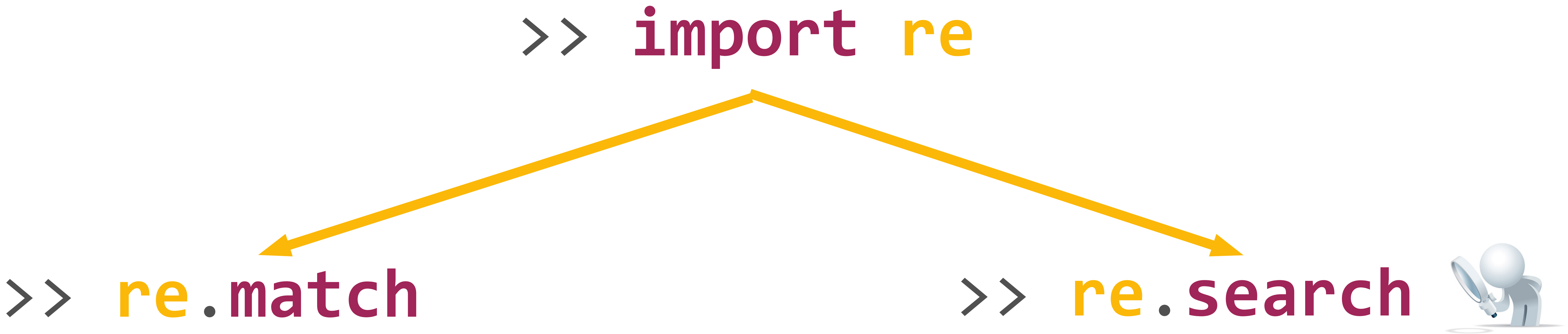
SUBSTITUTION



```
>> import re
```



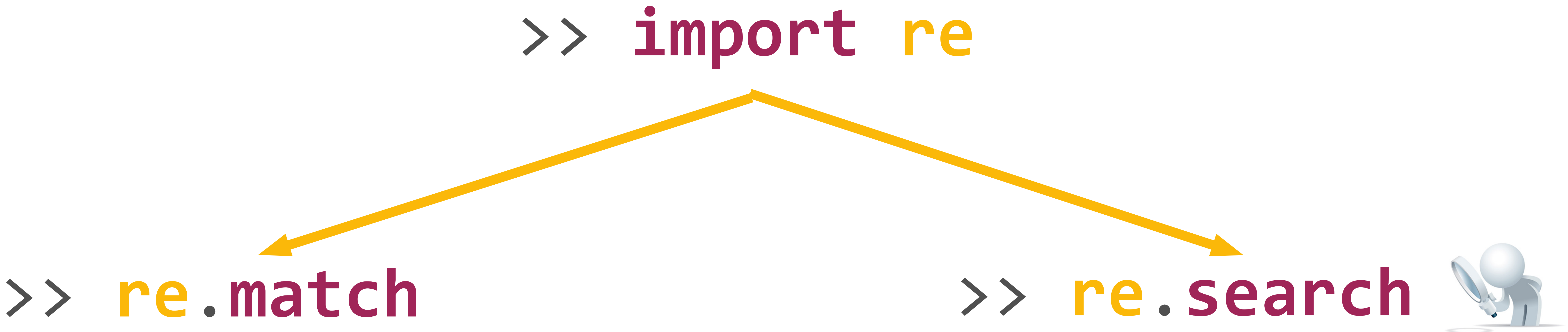
```
>> import re
>> re.match
>> re.search
```



```
>> import re
```

```
>> re.match
```

```
>> re.search
```



```
>> import re
```

```
>> re.match
```

```
>> re.search
```

really long item description … This item has pros and cons …

… **good** …

```
>> import re
```

```
>> re.match
```

```
>> re.search
>> import re
```

```
>> re.match
```

```
>> re.search
```

```
>> bool(re.match("good",
"good text!"))
True
```

```
>> bool(re.search("good",
"good text!"))
True
```

```
>> import re
```

```
>> re.match
```

```
>> re.search
```

```
>> bool(re.match("good",
"good text!"))
True
```

```
>> re.match("good", "It is a
good text")
None / False
```

```
>> bool(re.search("good",
"good text!"))
True
```

```
>> re.search("good",
"It is a good text"))
match / True
```

```
>> import re
```

```
>> re.match
```

```
>> re.search
```

```
>> bool(re.match("good",
"good text!"))
True

>> re.match("good", "It is a
good text")
None / False

>> re.match("good", "Good text
should start with capital G")
None / False
```

```
>> bool(re.search("good",
"good text!"))
True

>> re.search("good",
"It is a good text"))
match / True

>> re.search("good", "Good text
should start with capital G"))
None / False
```

```
>> import re
```

```
>> re.match
```

```
>> bool(re.match("good",
"good text!"))
True

>> re.match("good", "It is a
good text")
None / False

>> re.match("good", "Good text
should start with capital G")
None / False

>> re.match("good", "Good
text...", re.IGNORECASE)
match / True
```

```
>> re.search
```

```
>> bool(re.search("good",
"good text!"))
True

>> re.search("good",
"It is a good text"))
match / True

>> re.search("good", "Good text
should start with capital G"))
None / False

>> re.search("good", "Good text
...", re.IGNORECASE))
match / True
```

```
>> import re
```

```
>> re.search("wait", "text with
wait word")
match / True
```

```
>> re.search("w8", "text with
w8 word")
match / True
```

```
>> import re
```

```
>> re.search("wait", "text with
wait word")
match / True
```

```
>> re.search("w8", "text with
w8 word")
match / True
```

really long item description ... This item has pros and cons ...

... **good** ...

really long item description ... This item has pros and cons ...

... **nice** ...

```
>> import re
```

```
>> re.search("wait", "text with
wait word")
match / True
```

```
>> re.search("w8", "text with
w8 word")
match / True
```

really long item description
... This item has pros and
cons ...
... **good** ...

really long item description
... This item has pros and
cons ...
... **nice** ...

```
>> re.search("(good|nice)",
"text with good word")
match / True
```

```
>> re.search("(good|nice)", "text
with nice word")
match / True
```

```
>> re.search("(good|nice)",
"text with good word")
match / True
```

```
>> re.search("(good|nice)",
"text with nice word")
match / True
```

(...) - Capture Group

```
>> re.search("(good|nice)",
"text with good word")
match / True
```

```
>> re.search("(good|nice)",
"text with nice word")
match / True
```

(...) - Capture Group

```
>> re.search(pattern, text).groups()
```

```
>> re.search("(good|nice)",
"text with good word").groups()
('good',)
```

```
>> re.search("(good|nice)",
"text with nice word").groups()
('nice',)
```

```
>> re.search("(good|nice)",
"text with good word").group()
'good'
```

```
>> re.search("(good|nice)",
"text with nice word").group()
'nice'
```

(...) - Capture Group

```
>> re.search("(good|nice)",
"text with good word").groups()
('good',)
```

```
>> re.search("(good|nice)",
"text with nice word").groups()
('nice',)
```

```
>> re.search("(good|nice)",
"text with good word").group()
'good'
```

```
>> re.search("(good|nice)",
"text with nice word").group()
'nice'
```

~~(...) - Capture Group~~

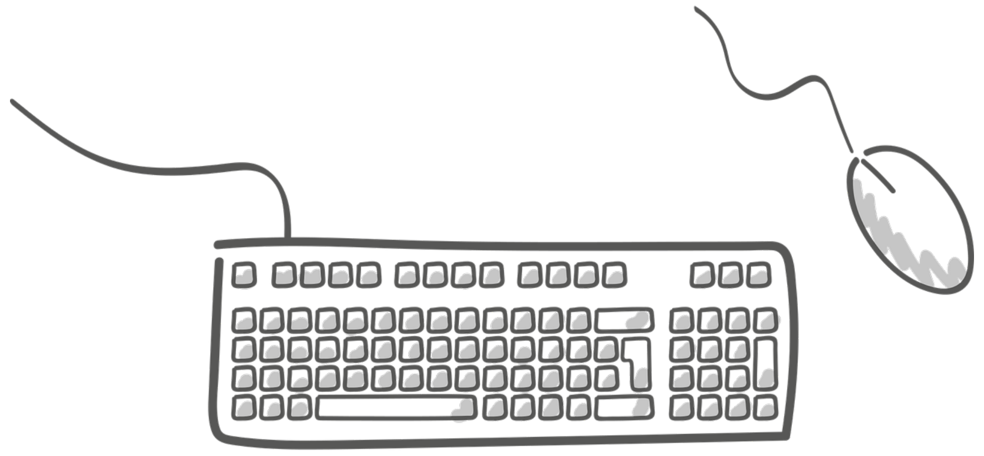
```
>> re.search("(good|nice)",
"text with good word").groups()
()
```

```
>> re.search("(good|nice)",
"text with nice word").groups()
()
```

```
>> re.search("(good|nice)",
"text with good word").group()
'good'
```

```
>> re.search("(good|nice)",
"text with nice word").group()
'nice'
```

optimize
products
for
xxx $$



Wired Keyboard and Wired Mouse Bundle

...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

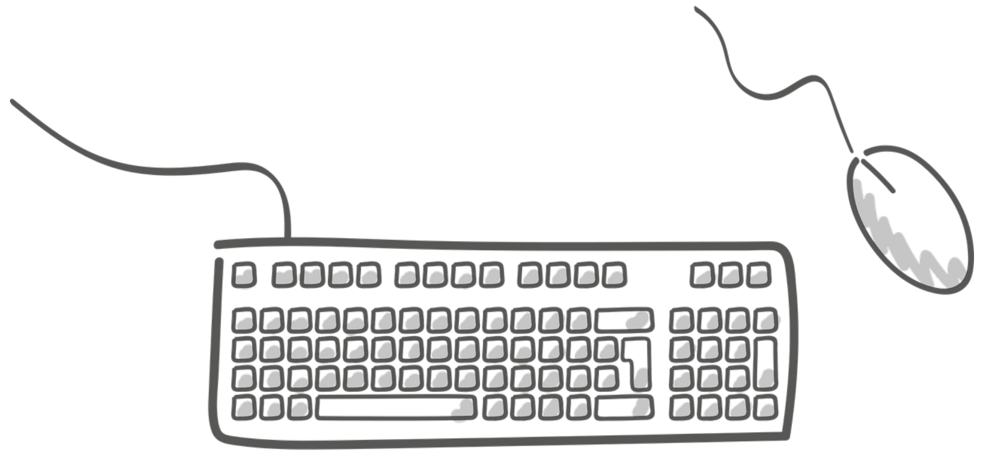
which is **good** for ...

Portable Wireless Bluetooth Speakers with Waterproof

... Rainfall or even accidental spils will no longer be a worrying factor ...

which is **nice** for ...

optimize
products
for
xxx $$



Wired Keyboard and Wired Mouse Bundle

...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

which is **good** for ...

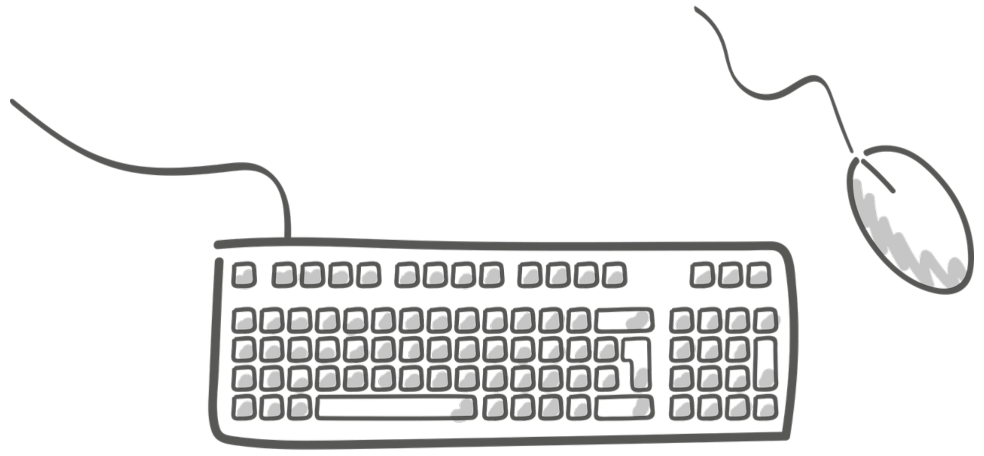
Portable Wireless Bluetooth Speakers with Waterproof

... Rainfall or even accidental spils will no longer be a worrying factor ...

which is **nice** for ...

```
>> re.search("really good|really nice",  ...)
```

optimize
products
for
xxx $$



Wired Keyboard and Wired Mouse Bundle

...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

which is **good** for ...

Portable Wireless Bluetooth Speakers with Waterproof

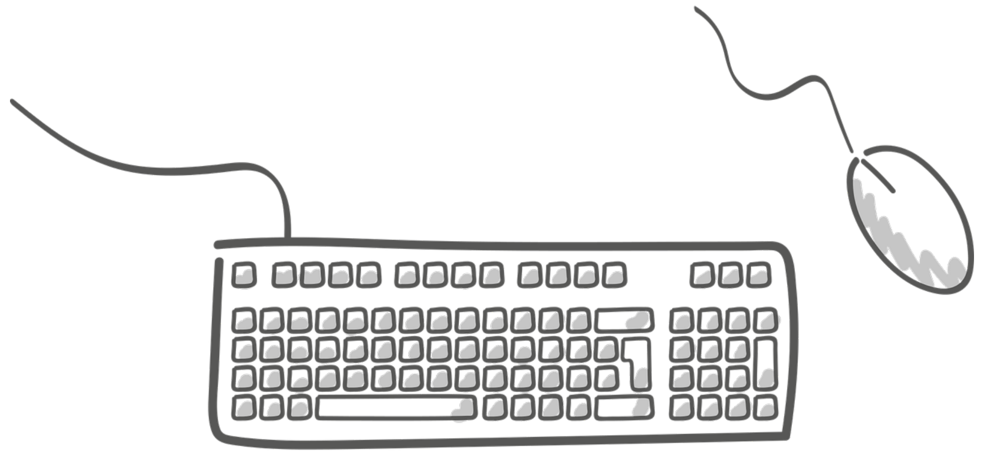
... Rainfall or even accidental spils will no longer be a worrying factor ...

which is **nice** for ...

```
>> re.search("really good|really nice",
...)
```

```
>> re.search("really (good|nice)",
...)
```

optimize products for xxx $$



Wired Keyboard and Wired Mouse Bundle

...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

which is **good** for ...

```
>> re.search("really good|really nice", ...)
```

group: really good

group**s**: really good

Portable Wireless Bluetooth Speakers with Waterproof

... Rainfall or even accidental spils will no longer be a worrying factor ...

which is **nice** for ...

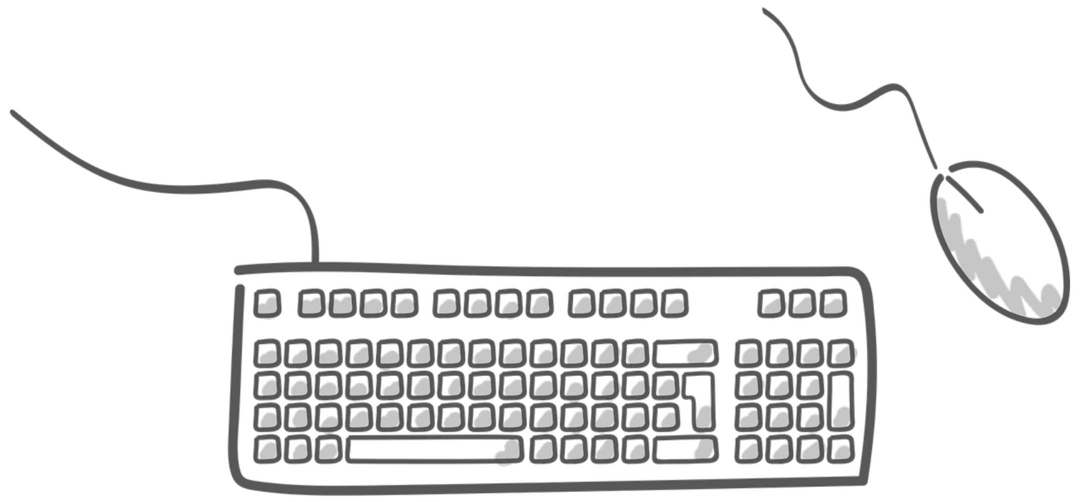
```
>> re.search("really (good|nice)", ...)
```

group: really nice

group**s**: nice

(?: ...) - Non-Capturing Group

```
>> re.search("really (?:good|nice)",  ...)
```



Wired Keyboard and Wired Mouse Bundle

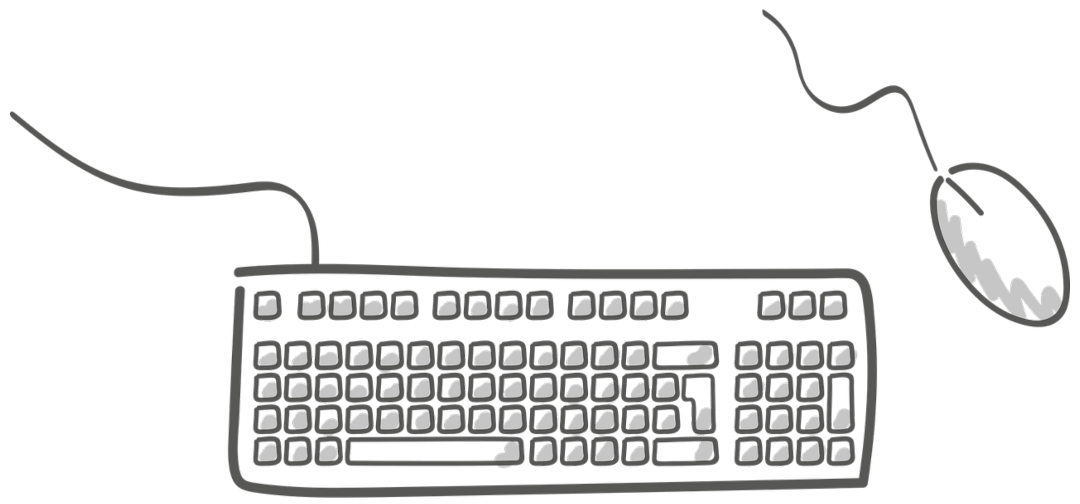
...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

... **good ???** ...

Portable Wireless Bluetooth Speakers with Waterproof

... Rainfall or even accidental spils will no longer be a worrying factor ...

... **nice ???** ...



Wired Keyboard and Wired Mouse Bundle

...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

... **good ???** ...

Portable Wireless Bluetooth Speakers with Waterproof

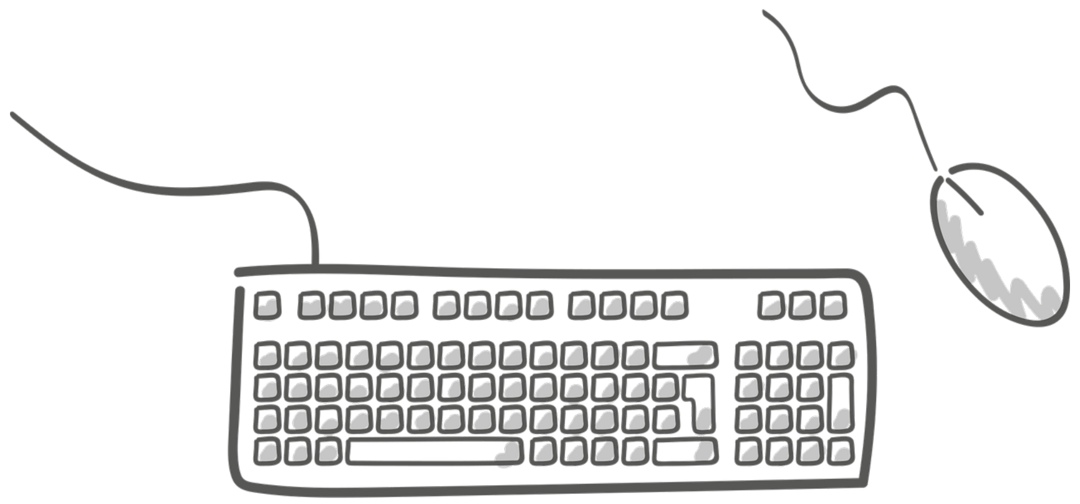
... Rainfall or even accidental spils will no longer be a worrying factor ...

... **nice ???** ...

[abcdefg] - set of characters a..g

[12345] - set of characters 1..5

[abd23] - example set of characters



Wired Keyboard and Wired Mouse Bundle

...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

... **good ???** ...

Portable Wireless Bluetooth Speakers with Waterproof

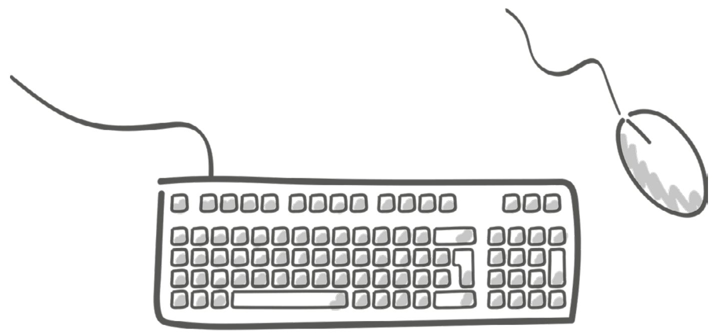
... Rainfall or even accidental spils will no longer be a worrying factor ...

... **nice ???** ...

[abcdefg] - set of characters a..g ← [a-g]

[12345] - set of characters 1..5 ← [1-5]

[abd23] - example set of characters



[abcdefg] - set of characters a..g ← [a-g]

[12345] - set of characters 1..5 ← [1-5]

[abd23] - example set of characters

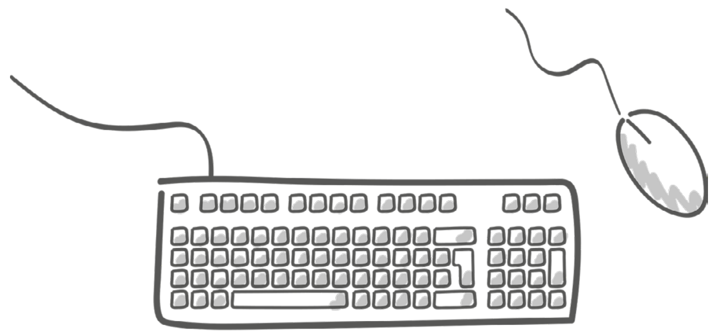
Wired Keyboard and Wired Mouse Bundle
...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

... good c??? ...

Portable Wireless Bluetooth Speakers with Waterproof
... Rainfall or even accidental spils will no longer be a worrying factor ...

... nice 3??? ...

```
>> re.search("(good|nice) [a-z]",
"text with good word").group()
'good w'
```



[abcdefg] - set of characters a..g ← [a-g]

[12345] - set of characters 1..5 ← [1-5]

[abd23] - example set of characters

Wired Keyboard and Wired Mouse Bundle
...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

... good c??? ...

Portable Wireless Bluetooth Speakers with Waterproof
... Rainfall or even accidental spils will no longer be a worrying factor ...

... nice 3??? ...

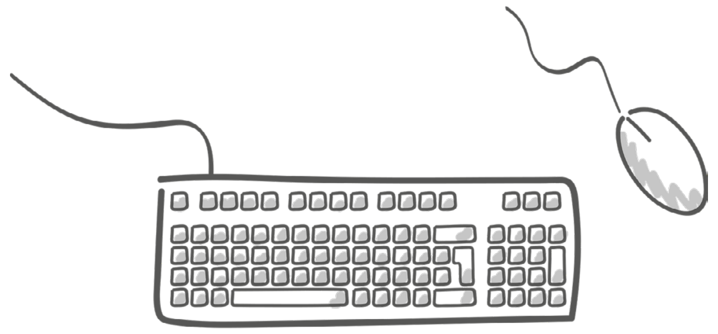
```
>> re.search("(good|nice) [a-z]",
"text with good word").group()
'good w'
```

```
>> re.search("(good|nice) [a-z]*",
"text with good word").group()
'good word'
```

[abcdefg] - set of characters a..g ← [a-g]

[12345] - set of characters 1..5 ← [1-5]

[abd23] - example set of characters



Wired Keyboard and Wired Mouse Bundle
...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

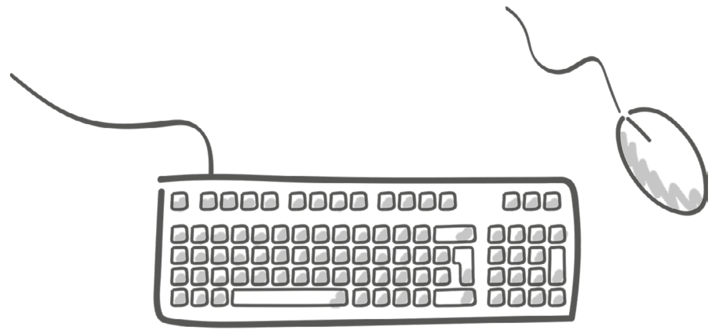
... **good c???** ...

```
>> re.search("(good|nice) [a-z]",
"text with good word").group()
'good w'

>> re.search("(good|nice) [a-z]*",
"text with good word").group()
'good word'
```

Portable Wireless Bluetooth Speakers
with Waterproof
... Rainfall or even accidental spils will no longer be a worrying factor ...

... **nice 3???** ...

```
>> re.search("go[a-z]*", "text with
good word").group()
'good'

>> re.search("go[a-z]*", "text with
go word").group()
'go'
```



[abcdefg] - set of characters a..g ← [a-g]

[12345] - set of characters 1..5 ← [1-5]

[abd23] - example set of characters

Wired Keyboard and Wired Mouse Bundle
...Bring sleek design and consistency to your desktop with the wired keyboard and mouse bundle from ...

... **good c???** ...

```
>> re.search("(good|nice) [a-z]",
"text with good word").group()
'good w'
```

```
>> re.search("(good|nice) [a-z]*",
"text with good word").group()
'good word'
```

Portable Wireless Bluetooth Speakers with Waterproof
... Rainfall or even accidental spils will no longer be a worrying factor ...

... **nice 3???** ...

```
>> re.search("go[a-z]*", "text with
good word").group()
'good'
```

```
>> re.search("go[a-z]*", "text with
go word").group()
'go'
```

```
>> re.search("go[a-z]+", "text with
go word")
no match / False
```

```
>> re.search("(really )?really (good|nice)", "long item description
of a really good product X").group()
'really good'

>> re.search("(really )?really (good|nice)", "long item description
of a really really good product X").group()
'really really good'
```

```
>> re.search("(really )?really (good|nice)", "long item description
of a really good product X").group()
'really good'

>> re.search("(really )?really (good|nice)", "long item description
of a really good product X").group()
'really really good'
```

way better vs. waaaaaay better

```
>> re.search("(really )?really (good|nice)", "long item description
of a really good product X").group()
'really good'

>> re.search("(really )?really (good|nice)", "long item description
of a really really good product X").group()
'really really good'
```

way better vs. waaaaaay better

waaaaaaaaaaaaaaaaaaaaa

aaaaaaaaaaaaaaaaaa

aaaaaaaaay better

way better vs. waaaaaay better

waaaaaaaaaaaaaaaaaaaaa

aaaaaaaaaaaaaaaaaa

aaaaaaaay better

```
>> re.search("wa{1,30}y", "waaay better").group()
'waaay'
```

way better vs. waaaaaay better

waaaaaaaaaaaaaaaaaaaaa

aaaaaaaaaaaaaaaaa

aaaaaaaay better

```
>> re.search("wa{1,30}y)", "waaay better").group()
'waaay'

>> re.search("wa{,30}y)", "wy better").group()
'wy'
```

way better vs. waaaaaay better

waaaaaaaaaaaaaaaaaaaaaa

ıaaaaaaaaaaaaaaaaa

ıaaaaaaaay better

```
>> re.search("wa{1,30}y)", "waaay better").group()
'waaay'

>> re.search("wa{,30}y)", "wy better").group()
'wy'

>> re.search("wa{1,}y)", "waaaaaaaay better").group()
'waaaaaaaay'
```

754-3010 Local

(**541**) 754-3010 Domestic

+1-**541**-754-3010 International

1-**541**-754-3010 Dialed in the US

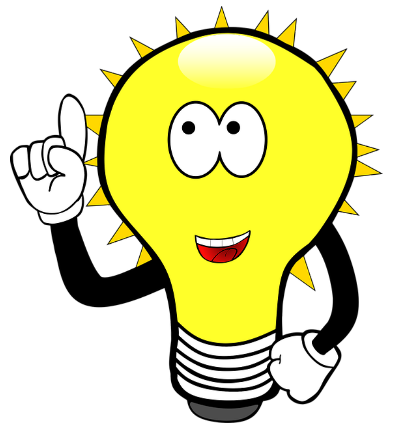
001-**541**-754-3010 Dialed from Germany

191 **541** 754 3010 Dialed from France

```
>> re.search("...[0-9]{3}...", ...)
```

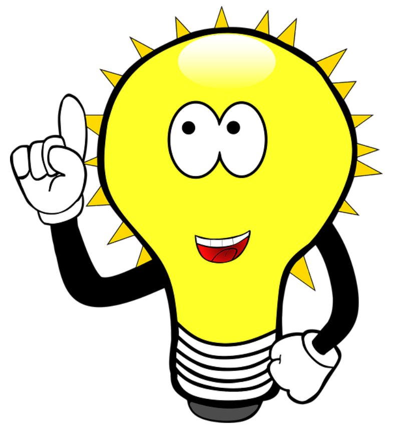
```
>> re.search("...[a-zA-Z0-9]...[0-9]+...[a-z]*", ...)
```

```
>> re.search("...[a-zA-Z0-9]...[0-9]+...[a-z]*", ...)
```



Metacharacters

```
>> re.search("...[a-zA-Z0-9]...[0-9]+...[a-z]*", ...)
```

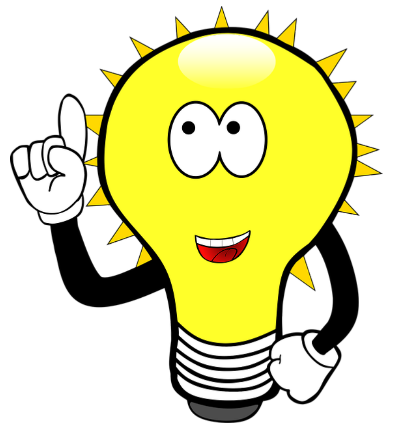


Metacharacters

\d - digits

```
>> re.match("\d+", "1230003")
match / True
```

```
>> re.search("...[a-zA-Z0-9]...[0-9]+...[a-z]*", ...)
```



Metacharacters

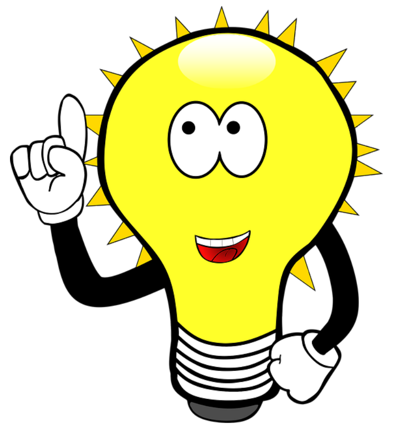
\d - digits

```
>> re.match("\d+", "1230003")
match / True
```

\w - alphanumeric characters

```
>> re.match("\w+", "0abc123def")
match / True
```

```
>> re.search("...[a-zA-Z0-9]...[0-9]+...[a-z]*", ...)
```



Metacharacters

\d - digits

```
>> re.match("\d+", "1230003")
match / True
```

\w - alphanumeric characters

```
>> re.match("\w+", "0abc123def")
match / True
```

\s - whitespace characters

```
>> re.match("\s+", "    ")
match / True
```

```
>> re.match("google[^\d]*.com", "google.com")
match / True

>> re.match("google[^\d]*.com", "google123.com")
no match / False

>> re.match("google[^\d]*.com", "google-ish.com")
match / True
```

```
>> re.match("google[^\d]*.com", "google.com")
match / True

>> re.match("google[^\d]*.com", "google123.com")
no match / False

>> re.match("google[^\d]*.com", "google-ish.com")
match / True
```

\D = [^\d] - no digit characters

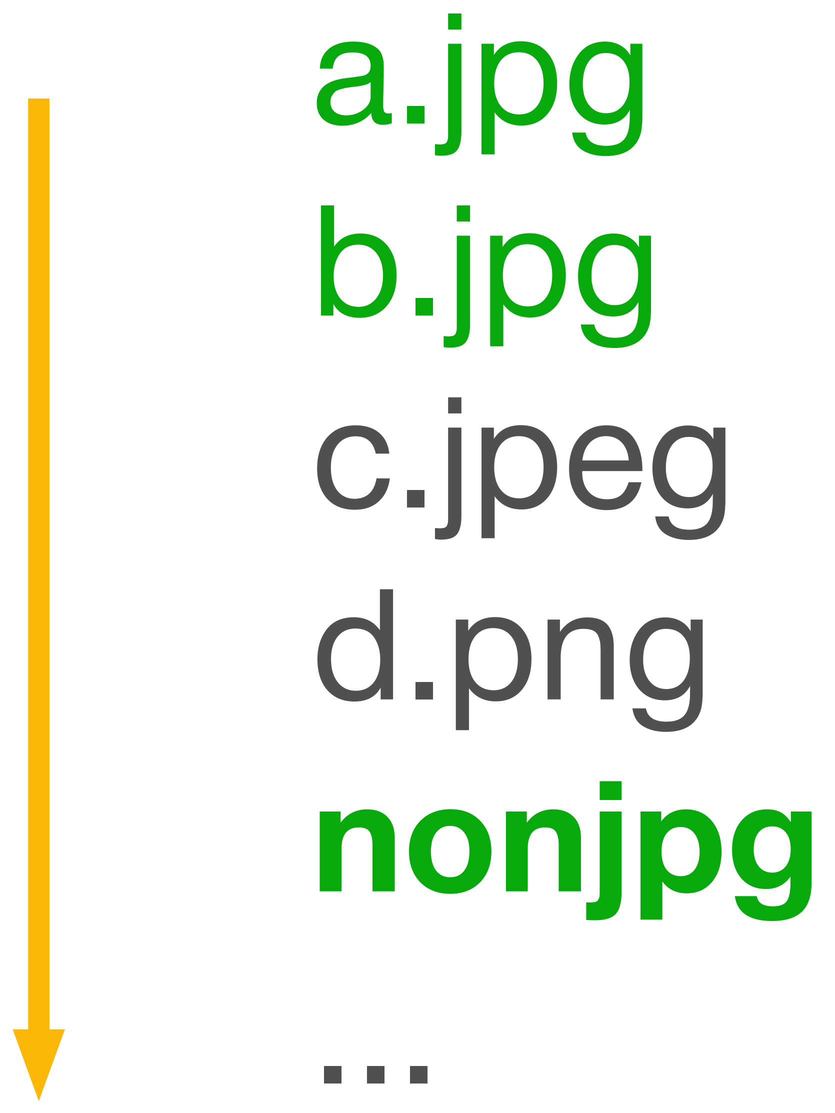
\W = [^\w] - no alphanumeric characters

\S = [^\s] - no whitespace characters

Regex Essentials

Regex Essentials

```
>> re.search(".jpg", filename)
```



a.jpg
b.jpg
c.jpeg
d.png
nonjpg
…

```
rm -rf <list of files>
```

Regex Essentials

```
>> re.search(".jpg", filename)
```

a.jpg
b.jpg
c.jpeg
d.png
nonjpg
...

```
rm -rf <list of files>
```

```
>> re.search("\.jpg", filename)
```

a.jpg
b.jpg
c.jpeg
d.png
nonjpg
...

```
rm -rf <list of files>
```

Regex Essentials

```
>> re.search("^good", "good description")
match / True
```

```
>> re.search("^good", "description is good")
no match / False
```

Regex Essentials

```
>> re.search("^good", "good description")
match / True

      >> re.search("good$", "good description")
      no match / False

>> re.search("^good", "description is good")
no match / False

      >> re.search("good$", "good description")
      match / True
```

Summary

Summary

- You can **write** simple regular expressions and **use** re.match and re.search in Python

Summary

- You can **write** simple regular expressions and **use** re.match and re.search in Python
- You can **use** capturing groups **to extract** necessary information from text

Summary

- You can **write** simple regular expressions and **use** re.match and re.search in Python
- You can **use** capturing groups **to extract** necessary information from text
- You can **explain** and **show** examples of usage of "^", "$", "." and "\" metacharacters

Summary

- You can **write** simple regular expressions and **use** re.match and re.search in Python
- You can **use** capturing groups **to extract** necessary information from text
- You can **explain** and **show** examples of usage of "^", "$", "." and "\" metacharacters
- You can **define** set of characters with "[]" and quantifiers in regular expressions

Summary

- You can **write** simple regular expressions and **use** re.match and re.search in Python
- You can **use** capturing groups **to extract** necessary information from text
- You can **explain** and **show** examples of usage of "^", "$", "." and "\" metacharacters
- You can **define** set of characters with "[]" and quantifiers in regular expressions

Python “re”: https://docs.python.org/2/library/re.html + https://docs.python.org/2/howto/regex.html#regex-howto
see: https://regexone.com/lesson/introduction_abcs + https://regex101.com/