

# Getting InDesign to do Python Syntax Highlighting

**Amy Mather - @MiniGirlGeek**



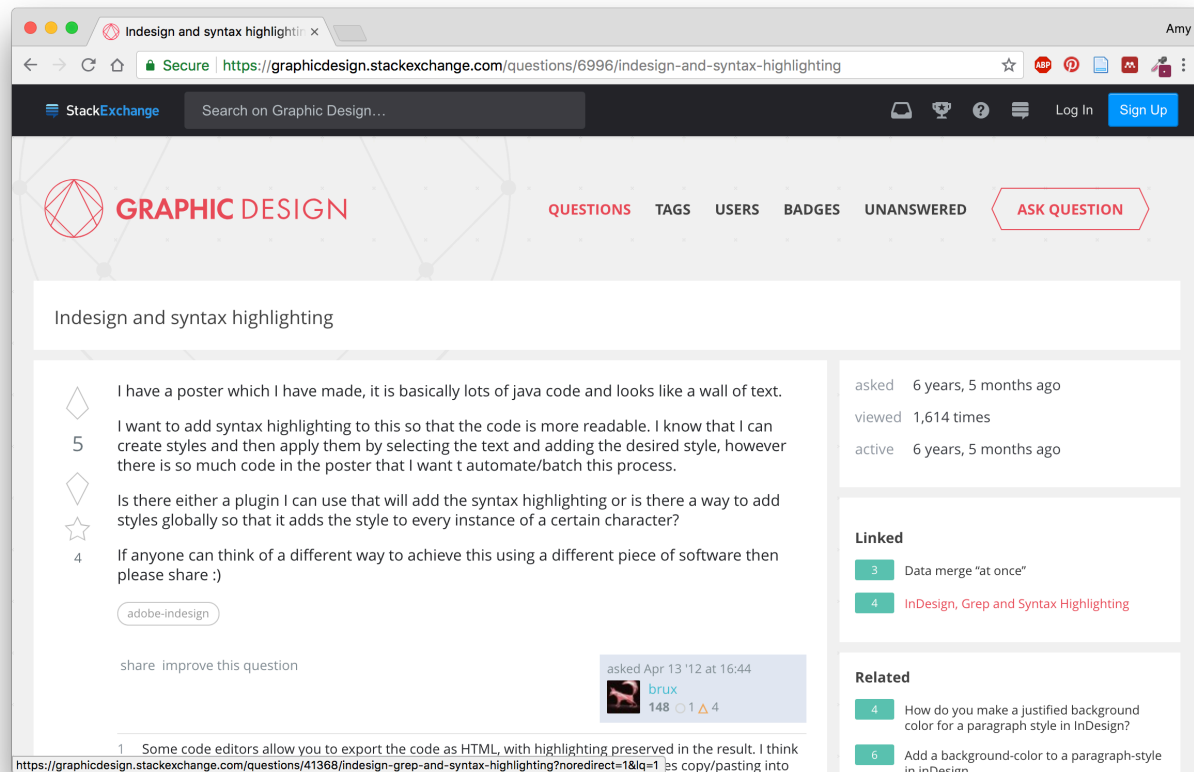
indesign code syntax highlighting



Google Search

I'm Feeling Lucky

**Amy Mather - @MiniGirlGeek**



**Amy Mather - @MiniGirlGeek**

This is the kind of task where GREP Styles shine. Since you're setting javascript, I assume you're not unfamiliar with grep, but the GREP (for some reason, Adobe uses all caps) Styles dialog has plenty of help if you're rusty.

The first step is to create a set of colors you'll want to use to highlight different code elements, so they're ready in the Swatches panel.

The next step is to assign a Paragraph Style to all the code, if you haven't already. Once that's done, edit the Paragraph Style and go to the "GREP Style" section, where the magic starts.

```
// function created using the Scripting Listener
// to capture the command for Layer>New Layer Based Slice
function nameLayerBasedSlice(){
var id2488 = charIDToTypeID( "setd" );
var desc298 = new ActionDescriptor();
var id2489 = charIDToTypeID( "setd" );
var ref168 = new ActionReference();
var id2490 = stringIDToTypeID( "Paragraph Styles" );
var id2491 = charIDToTypeID( "setd" );
var id2492 = charIDToTypeID( "setd" );
ref168.putEnumerated( id2490, id2491, id2492 );
desc298.putReference( id2489, ref168 );
var id2493 = charIDToTypeID( "setd" );
var desc299 = new ActionDescriptor();
var id2494 = charIDToTypeID( "setd" );
desc299.putString( id2494, "code" );
var id2495 = stringIDToTypeID( "Paragraph Styles" );
var id2496 = stringIDToTypeID( "Paragraph Styles" );
var id2497 = charIDToTypeID( "setd" );
desc299.putEnumerated( id2495, id2496, id2497 );
var id2498 = stringIDToTypeID( "Paragraph Styles" );
desc298.putObject( id2493, id2498, desc299 );
}
```

**Paragraph Style Options**

Style Name: Code

GREP Style

Click New GREP Style button to create a GREP style.

New GREP Style Delete

**Hot Network Questions**

- How would a ship defend against a sea creature, and possibly win?
- Changing the distance considered on a metric space changes open sets inside?
- Filament isn't going into the bowden tube. Instead it goes "into the room"
- Are suites still danced to today?
- How to defuse or prevent an adult temper tantrum
- Why does work depend on distance?
- 20% monthly mining vs 5% monthly trading
- How to handle a paper by a reviewer who wants to be paid?
- How to pass a bicycle while driving a car
- Axis of rotation of the body
- Why would a turboprop rev down just before going full throttle during takeoff?

Amy Mather - @MiniGirlGeek

<http://www.ericagamet.com/wp-content/uploads/2016/04/Erica-Gamets-GREP-Cheat-Sheet.pdf>

A screenshot of a web browser displaying the first page of a PDF titled "Erica-Gamets-GREP-Cheat-Sheet.pdf". The browser's address bar shows "www.ericagamet.co...". The PDF content features a blue header with the title "GREP QUICK REFERENCE CHART". Below this, there are three main sections: "WILDCARDS", "LOCATIONS", and "REPEAT". Each section contains a table of GREP symbols and their functions, accompanied by example sentences. For instance, under "WILDCARDS", it lists symbols for digits, letters, characters, white space, word characters, lowercase letters, and uppercase letters. The "LOCATIONS" section includes symbols for beginning/end of word, word boundaries, and paragraph/story boundaries. The "REPEAT" section covers quantifiers like "?", "\*", "+", and "??". The bottom of the page has a blue footer with the text "See more GREP examples and other Tips and Tricks at www.ericagamet.com".

### GREP QUICK REFERENCE CHART

WILDCARDS	
Any Digit	\d
Any Letter	[A-Za-z]
Any Character	.
Any White Space	\s
Any Word Character	\w
Any Lowercase Letter	[a-z]
Any Uppercase Letter	[A-Z]

**Any Digit finds each single digit:**  
Mary had 3 little lambs. Her whole flock was made up of these three, and 15 fully-grown sheep.

**Any Letter finds each single letter (uppercase or lowercase):**  
Mary had 3 little lambs. Her whole flock was made up of these three, and 15 fully-grown sheep.

**Any Character finds each single character (except line break):**  
Mary had 3 little lambs. Her whole flock was made up of these three, and 15 fully-grown sheep.

LOCATIONS	
Beginning of Word	\b
End of Word	\b
Word Boundary	\b
Beginning of Paragraph	\P
End of Paragraph	\P
Beginning of Story	\S
End of Story	\S

**Beginning of Paragraph:**  
Mary had 3 little lambs. Her whole flock was made up of these three, and 15 fully-grown sheep.

**End of Paragraph:**  
Mary had 3 little lambs. Her whole flock was made up of these three, and 15 fully-grown sheep.

**End of Story:**  
Mary had 3 little lambs. Her whole flock was made up of these three, and 15 fully-grown sheep.  
Her sister, Shari, didn't have any sheep at all!

REPEAT	
Zero or One Time	?
Zero or More Times	*
One or More Times	+
Zero or One Time (Shortest Match)	?*
One or More Times (Shortest Match)	?+

**Zero or One Time:**  
The item may appear one time in sequence, or it may not appear at all.

**Zero or More Times:**  
The item may appear any number of times in sequence, or it may not appear at all.

**One or More Times:**  
The item appears one time or more in sequence.

**Shortest Match:**  
Keeps the search to the first complete sequence. Without "shortest match," InDesign looks to the whole paragraph for the sequence.

See more GREP examples and other Tips and Tricks at [www.ericagamet.com](http://www.ericagamet.com)

A screenshot of a web browser displaying the second page of the "Erica-Gamets-GREP-Cheat-Sheet.pdf". The browser's address bar shows "www.ericagamet.co...". The PDF content continues the "GREP QUICK REFERENCE CHART" with a "MATCH" section and several explanatory paragraphs. The "MATCH" section is a table listing various GREP match symbols and their uses. Following this, there are sections titled "Offset text as expressions:", "Look for something OR something else:", and "Look before or after to find a string:". Each section provides detailed instructions and examples for using the corresponding GREP features. The bottom of the page has a blue footer with the text "See more GREP examples and other Tips and Tricks at www.ericagamet.com".

MATCH	
Marking Subexpression	() (parentheses)
Non-Marking Subexpression	(?:)
Character Set	[] (square brackets)
Or	(pipe character)
Positive Lookbehind	(?<=)
Negative Lookbehind	(?<!=)
Positive Lookahead	(?=)
Negative Lookahead	(?!)

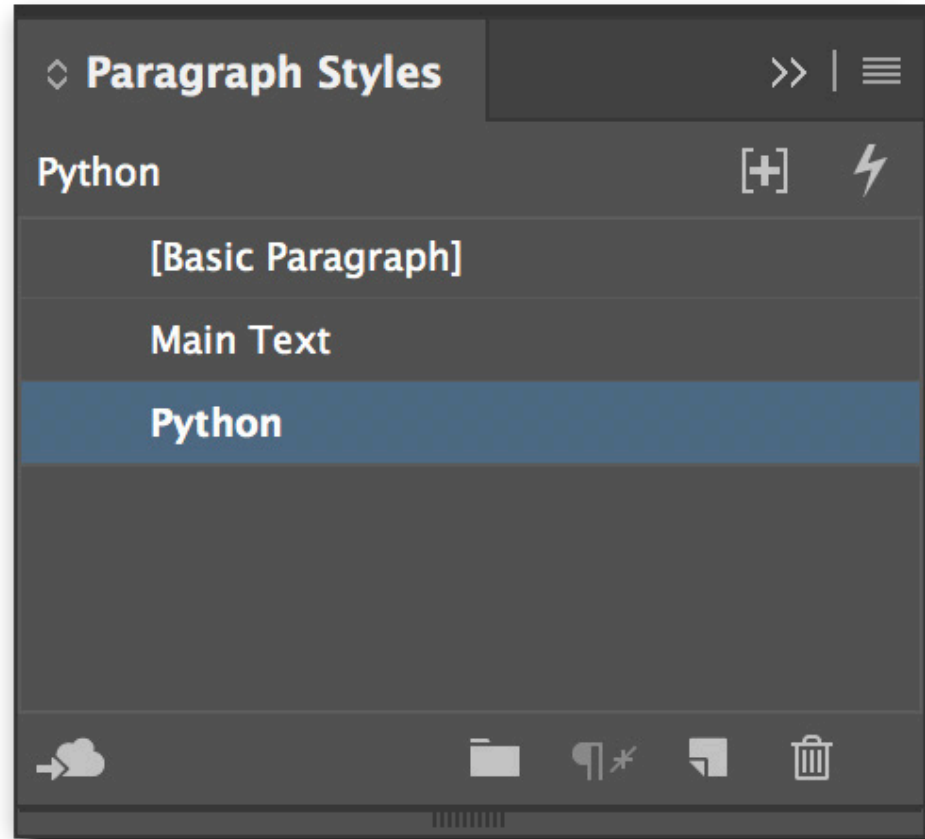
**Offset text as expressions:**  
Use parentheses to offset an entire expression or to group items together. For instance, in Find/Change, if you are looking for a string of 10 digits, but you want to return them in a different order, or insert additional information between the groups, use parentheses.  
**Example** - format a phone number from a 10-digit string:  
3035551212  
In Find/Change: (\d\d\d)(\d\d\d)(\d\d\d)(\d\d\d)  
This will see each group separately. You could then return only the 2nd and 3rd groups, if wanted. (See "Found" expressions)

**Look for something OR something else:**  
Use the straight slash (or pipe character, located over the backslash) to indicate either this item OR that item. You can have multiple items, each divided by the pipe character, but you need to enclose this OR search in parentheses.  
**Example** - look for the word "grey" or "gray." You could easily look for the whole word spelled both ways, but the only difference is the letter "e" or "a." The search will always look to the entire string on either side of the pipe character.  
Look for: gr(e|a)y - Will find: grey AND gray  
Look for: (Re|Gr|Bl|Wh) - Will find: Red, Green, AND Blue  
Look for: (color|fill|stroke|color|color) - Will find each of these phrases

**Look before or after to find a string:**  
Use Lookahead and Lookbehind to locate a string and affect just that string.  
**Example:** Find digits after a decimal point by using a Positive Lookbehind.  
Looking behind the text (what appears before), if the string exists (the decimal point in this case), then do something with the searched text (the digits).  
Look for: (?<=\d)\d+  
To find this: 123.45  
Use Positive Lookbehind (?<=) and add in the decimal point. However, since the decimal means something specific in GREP, you need to "escape" the decimal to indicate you're looking for an actual decimal. That is the backslash before the decimal. Put that inside the parentheses as well so it finds an entire string. The next part, the \d+, is looking for any digit, one or more times. You can then do something to the found portion, which is the 45 in the example above.

See more GREP examples and other Tips and Tricks at [www.ericagamet.com](http://www.ericagamet.com)

Amy Mather - @MiniGirlGeek



**Amy Mather - @MiniGirlGeek**

`(\w*)(?=\=)`

Keyword Argument



**Amy Mather - @MiniGirlGeek**

```
(def \w*\()\K(. | )(?=\))
```

Parameter

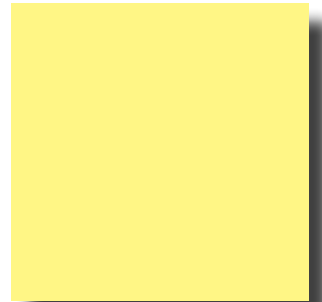


**Amy Mather - @MiniGirlGeek**



("1").+("1")

String



**Amy Mather - @MiniGirlGeek**

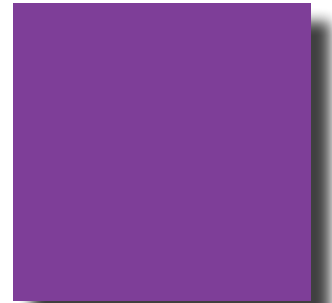


# Commas

**Amy Mather - @MiniGirlGeek**

`(?<!\w)\d+`

Numbers



**Amy Mather - @MiniGirlGeek**

(?<!w)(try|continue|for...

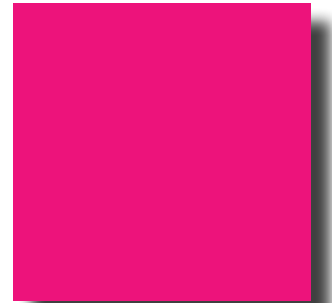
Keywords



**Amy Mather - @MiniGirlGeek**

$\backslash = \backslash + \backslash - \backslash \vee \backslash *$

# Operators



**Amy Mather - @MiniGirlGeek**

# True|False|None

## Boolean



**Amy Mather - @MiniGirlGeek**

```
(#. *|("'"|'"). *("'"|''))
```

Comment



**Amy Mather - @MiniGirlGeek**

# `(def|class|lambda)(?!\\w)`

## Defining



**Amy Mather - @MiniGirlGeek**



`\w+?(?=\s)`

Defined name



**Amy Mather - @MiniGirlGeek**

**(def|class) \K\w\***

Definition name



**Amy Mather - @MiniGirlGeek**

```
>>> import library

>>> def a_function(that, has, parameters=True):
    #I am a comment
    some_text = "a string"
    some_maths = 99 * 2018

>>> a_function(some, values, parameters=False)
```

**Amy Mather - @MiniGirlGeek**