Like a Puzzle: Assembling RegEx through Blockly

Hello! Work with your group to discuss and solve and explore the following list of tasks. If you have questions, please don't hesitate to ask!

Help Documentation

1. Please navigate to the Help Documentation and read/skim the **Quick Start Guide** page. Are there any errors in the wording? Any points of confusion that arise for you?

2. Considering the Blocky language outlined in section **Block Reference Guide** of the Help Documentation, would you consider the language complete? Is there anything missing?

3. Go to the **Examples** section. Are there any examples that stand out to you as incorrect or confusing?

First Blocky Regex

Follow the steps below to assemble your first blocky regular expression. If you are experienced in regular expressions, you are welcome to skip these steps.

4. Use the **Text** block to capture any instance of the word “Hello”
5. Add the **Start** block in front of the Text block.
6. Add a comma to the end of the first **Text** block and add a **Include Whitespace** block after the Text block. Ensure only the s is checked.
   Add an Or block. Add a Text block to both of the green starter expression blocks. Edit the Text blocks to say “world” and your name, respectively. Continue to add an additional two or more Text options to the Or block by dragging more short green blocks to the empty section(s) and more Text blocks.
7. Next, add a Repeat block to the top/end of the Or block. Put a Text block inside the Repeat block that contains a single “!”. Change the Infinity to be 4.
8. Add an End block after the Repeat
9. Write the following test cases:
   a. Hello, world!
   b. Hello, Laura!!!
   c. Hello, you!!!!!
10. Write some more of your own!

Pass the tests

The following tasks will have tables showing test cases that pass or don't pass. Craft a blocky regular expression to pass all the cases for each table of tests. Careful to note the “does match” check mark!

11. Test Cases

Text	**Does Match?**	**Test passed?**
I LOVE PLOP	X	✓
I love PLoP!	✓	✓
Welcome to PLoP 2024!	✓	✓
I LOVE PLOP 2024!	X	✓
Are we at the same PLoP?	✓	✓
PLoP is a great acronym and onomatopoeia.	✓	✓
PLoP!!	✓	✓

12. Test Cases

Text	**Does Match?**	**Test passed?**
I think Jar Jar Binks is great!	✓	✓
Jar Jar Binks may be the greatest.	✓	✓
Jarjarbinks is silly	X	✓
jar jar binks	X	✓
Jar Jar Binks is good	X	✓
Jar Jar Binks is GREAT	X	✓

13. Test Cases

Text	**Does Match?**	**Test passed?**
I have a joke about variables but sadly I can’t remember it.	✓	✓
I have a CS joke but it doesn’t compute.	✓	✓
You have a CS joke?	X	✓
I have a joke.	✓	✓
I don’t have a joke—	X	✓
I have a joke about my work as a software engineer, but it only works for me.	✓	✓

14. Test Cases

Text	**Does Match?**	**Test passed?**
We_Dont_Need_No_Stinking_Space_Bar	✓	✓
Aint_A_problem_To_Avoid_A_spAcebAr	✓	✓
Underscores_Are_For_Cool_Kids	✓	✓
Spaces are lame	X	✓

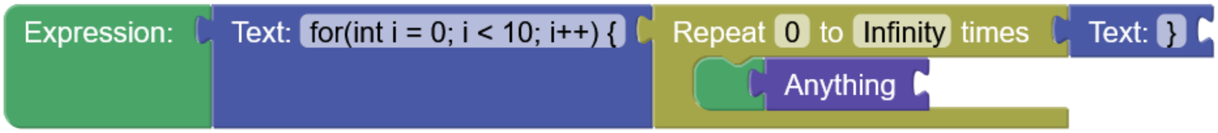
15. Test Cases

Text	**Does Match?**	**Test passed?**
10 S paces apart	✓	✓
S paces are fun	✓	✓
S - - - - paces are weird	X	✓
Hi bye	✓	✓

Testing

For each blocky regex below, please write & implement your own test cases.

16. Blocky Regex

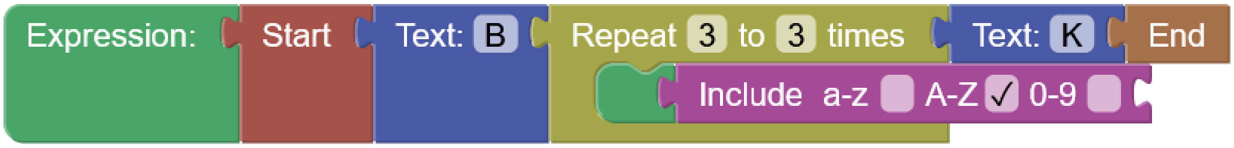


Expression: Text: for(int i = 0; i < 10; i++) { Repeat 0 to Infinity times Anything Text: }

Tests Cases

Text	Does Match?	Test passed?

17. Blocky Regex

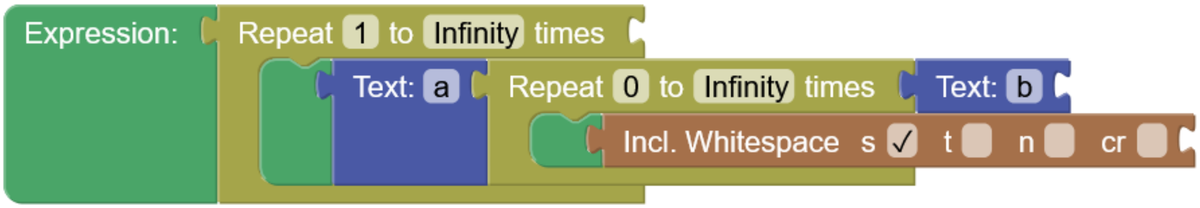


Expression: Start Text: B Repeat 3 to 3 times Include a-z □ A-Z ☑ 0-9 □ Text: K End

Test Cases

Text	Does Match?	Test passed?

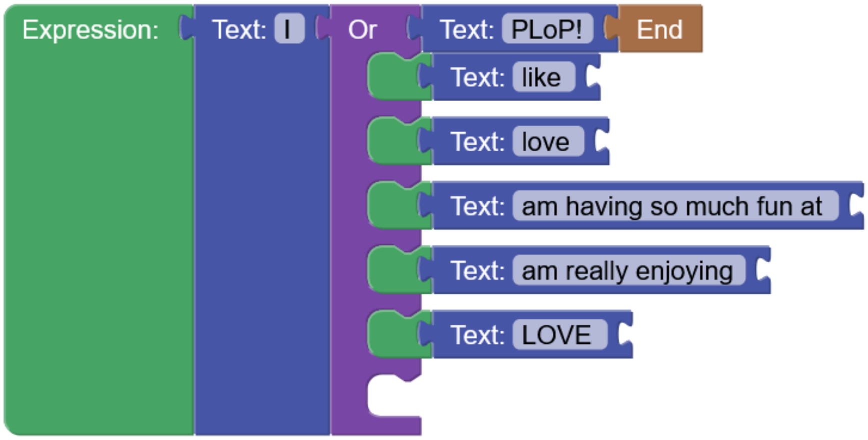
18. Blocky Regex


Expression:
Repeat 1 to Infinity times
Text: a
Repeat 0 to Infinity times
Text: b
Incl. Whitespace s ✓ t n cr

Test Cases

Text	Does Match?	Test passed?

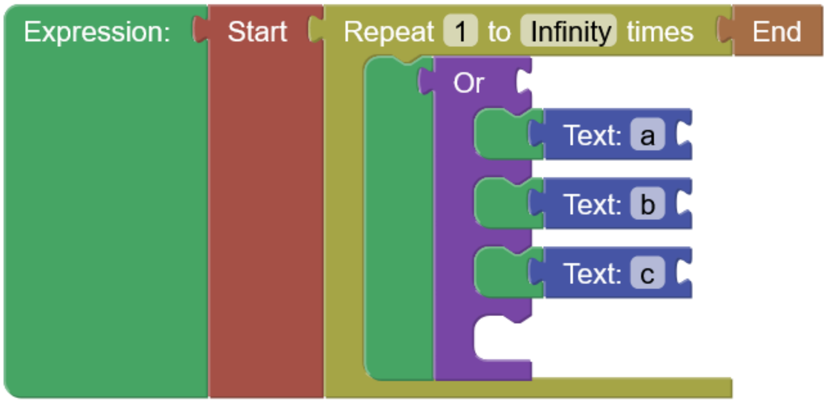
19. Blocky Regex



Test Cases

Text	Does Match?	Test passed?

20. Blocky Regex



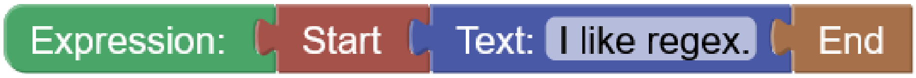
Test Cases

Text	Does Match?	Test passed?

Blocky Regex + Test Cases

Please look at the given blocky regex and its corresponding tests.

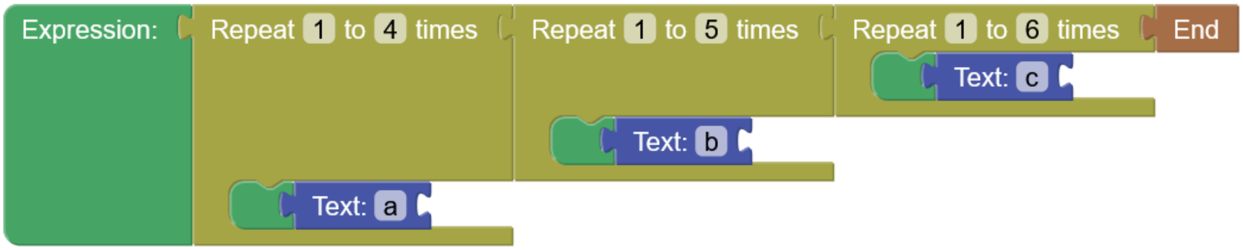
21. Blocky Regex + Tests



Text	Does Match?	Test passed?
I like regex.	✓	✓
I don’t like regex.	X	✓

Do you think the tests cover all edge cases? If not, what test is missing?

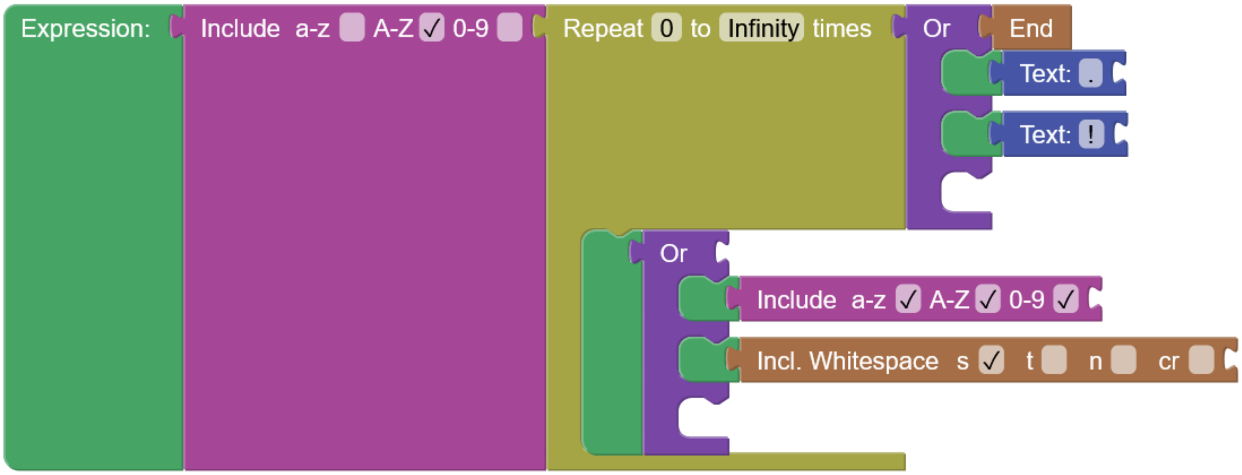
22. Blocky Regex + Tests



Text	Does Match?	Test passed?
abc	✓	✓
aaaaaabbbbbbbcccc	✓	✓

Do you think the tests cover all edge cases? If not, what test is missing?

23. Blocky Regex + Tests



Expression: Include a-z □ A-Z ☑ 0-9 □ Repeat 0 to Infinity times Or End

Text: .

Text: !

Or

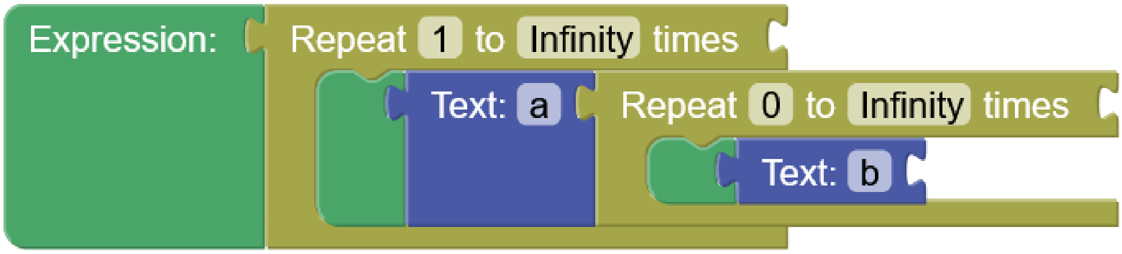
Include a-z ☑ A-Z ☑ 0-9 ☑

Incl. Whitespace s ☑ t □ n □ cr □

Text	Does Match?	Test passed?
This is a sentence.	✓	✓
This is an EXTREME sentence.	✓	✓
this isnt a sentence.	X	✓
Is this a sentence?	X	✓

Do you think the tests cover all edge cases? If not, what test is missing?

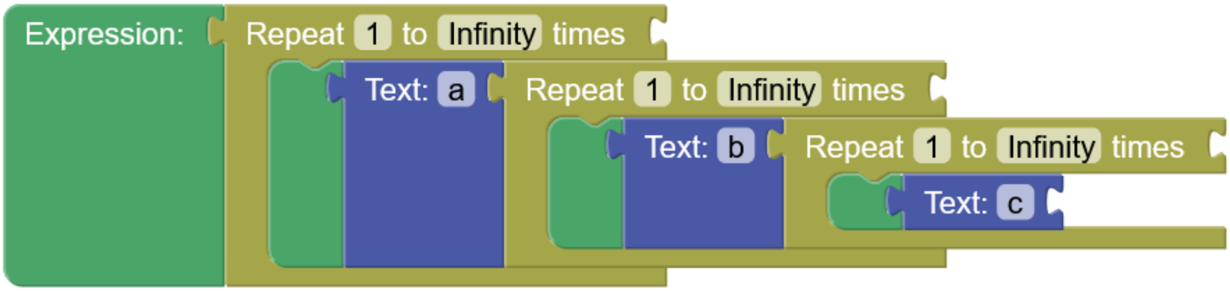
24. Blocky Regex + Tests



Text	**Does Match?**	**Test passed?**
ab	✓	✓
aaaaaaaaaaaaaaaaaaaaaaa	✓	✓
abbbbbbbbbbbbbbbbbbbb	✓	✓

Do you think the tests cover all edge cases? If not, what test is missing?

25. Blocky Regex + Tests


Expression:
Repeat 1 to Infinity times
Text: a
Repeat 1 to Infinity times
Text: b
Repeat 1 to Infinity times
Text: c

Text	Does Match?	Test passed?
abc	✓	✓
abcabc	✓	✓
aaabbbccc	X	✓

Do you think the tests cover all edge cases? If not, what test is missing?

Regex to Blocky

Translate the following text regular expressions (both in Java regex and the app's regex), to Blocky Regex.The first three provide test cases to aid. The last two do not have this aid as a challenge.

26. Regex (in app): `(?:.){1,}`
    Regex (Java): `.+`

Tests:

Text	Does Match?	Test passed?
gibberish	✓	✓
Literally anything should pass	✓	✓
ilshadbfipwqrjbgwr	✓	✓
8071256415761986509	✓	✓

27. Regex (in app): `^a(?:-){1,}b$`
    Regex (Java): `^a-+b$`

Tests:

Text	Does Match?	Test passed?
ab	X	✓
a-b	✓	✓
a------b	✓	✓

28. Regex (in app): `^(?:(?:a|b|c|d)){1,}e$`
    Regex (Java):

Tests:

Text	Does Match?	Test passed?
abcde	✓	✓
acdc	X	✓
ababababae	✓	✓
age	X	✓

29. Regex (Challenge): `#(?:(?:A|B|C|D|E|F|[0-9])){6,6}$`
    Regex (Java): `#([A-F]|[0-9]){6}$`

30. Regex (Challenge):
    `rgb\((?:[0-9]){1,3},(?:[0-9]){1,3},(?:[0-9]){1,3}\)`
    Regex (Java): `rgb\([0-9]{1,3},[0-9]{1,3},[0-9]{1,3}\)`