

Side By Side

It's often handy to print text files side-by-side, making their contents easier to compare. In fact, your team works regularly with text comparison and would love a utility written in your favorite language to help automate the task. The script should be expandable to handle a variety of custom parameters in the future (the team plans to use it to build their own `diff`- and `pr`-like utility, expanding on this project).

Your task for this challenge is to write a command line utility which can print a series of text files side-by-side according to the following specifications:

Input

- The script file to code is `bin/sidebyside.js`.
- `sidebyside.js` may be passed two optional flags:
 - `-s`: the size (width) in characters of each horizontal text column. Defaults each column to the longest row length for its file contents if not provided. Will be > 0 if provided.
 - `-c`: the number of space characters between text columns. Defaults to 2 if not provided. If provided, will be ≥ 0 .
- Beyond these two optional flag arguments, `sidebyside.js` will be passed a series of unflagged filenames, whose contents should be read from file and displayed side-by-side in columns in the order provided in the arguments array. If there are no files in the arguments array, simply exit without printing anything.
- Arguments may appear in any order, although the flagged arguments will always be followed by a positive number if present. There is no need to check these flagged arguments for errors, but if you wish to go the extra mile, you may.

Output

- Your script should print the result of the columnization operation determined by the argument file's contents and `-s` and `-c` flags (described above) to stdout.
- All characters in the text files should be left intact, with two exceptions: tab characters (`\t`) and newline characters (`\n`), which interfere with the column output.
 - Tabs should be converted to 4 spaces and treated as normal space characters in the output.
 - Newlines should be converted to enough spaces to fill the remainder of the current line, ensuring the next line of text begins at position 0 of the file's column in the next row.
- When one file is shorter than the others, any padding rows needed to make it the same size as the longest file should be entirely empty spaces. In other words, the output should be a rectangle of characters, each row with a width equal to the following formula, not including the newline character:

```
total_row_width = column_size * num_files + column_spacing * (num_files - 1)
```

Tests

You've been provided a set of plain text files and tests to help guide you to the appropriate output. All tests will take place on the plain text files included in the `./content` directory.

The example and submission tests run the `sidebyside.js` script and examine `stdout/stderr`, but you may prefer writing your own tests in `candidate_tests.js` to validate your module and function components piecemeal to make development and debugging easier. You can create any number of supporting source files or modules using any names you wish as long as the `sidebyside.js` file is ultimately able to take the command line arguments and produce the correct output.

Examples

Consider the following set of three text files.

Here are the contents of `./content/poetry/basho_rexroth.txt`:

Matsuo Bashô: Frog Haiku
Translated by Kenneth Rexroth

An old pond –
The sound
Of a diving frog.

Here are the contents of `./content/poetry/basho_chilcott.txt`:

Matsuo Bashô: Frog Haiku
Translated by Tim Chilcott

ancient is the pond –
 suddenly a frog leaps – now!
 the water echoes

Here are the contents of `./content/poetry/basho_marks.txt`:

Matsuo Bashô: Frog Haiku
Translated by Alfred H. Marks

There once was a curious frog
Who sat by a pond on a log
And, to see what resulted,
In the pond catapulted
With a water-noise heard round the bog.

Example 1

The command (which will be run by the test code)

```
$ node bin/sidebyside.js content/poetry/basho_rexroth.txt
content/poetry/basho_chilcott.txt
```

should write the following string to stdout:

```
'Matsuo Bashô: Frog Haiku      Matsuo Bashô: Frog Haiku      \nTranslated by
Kenneth Rexroth Translated by Tim Chilcott      \n'
```

When printed on the terminal, this string is rendered as:

```
Matsuo Bashô: Frog Haiku      Matsuo Bashô: Frog Haiku
Translated by Kenneth Rexroth  Translated by Tim Chilcott

An old pond –                  ancient is the pond –
The sound                      suddenly a frog leaps – now!
Of a diving frog.             the water echoes
```

Here's a version that has been modified to illustrate the locations of added space characters (denoted by ~) not present in the original source (your version should show the above output; below is a demonstration only):

```
Matsuo Bashô: Frog Haiku~~~~~ Matsuo Bashô: Frog Haiku~~~~~
Translated by Kenneth Rexroth  Translated by Tim Chilcott~~~~~
~~~~~
An old pond ~~~~~~              ancient is the pond ~~~~~~
The sound~~~~~                suddenly a frog leaps – now!
Of a diving frog.~~~~~         the water echoes~~~~~
```

Example 2

This example illustrates output when the left file is longer than the right. The command

```
$ node bin/sidebyside.js content/poetry/basho_marks.txt
content/poetry/basho_chilcott.txt
```

prints the string

```
'Matsuo Bashô: Frog Haiku      Matsuo Bashô: Frog Haiku      \n
\nTranslated by Alfred H. Marks  Translated by Tim Chilcott      \n
\nThere once was a curious frog  ancient is the pond –      \n
\nWho sat by a pond on a log      suddenly a frog leaps –
now!\nAnd, to see what resulted,   the water
echoes      \nIn the pond catapulted
\nWith a water-noise heard round the bog.      \n'
```

which renders as

```
Matsuo Bashô: Frog Haiku      Matsuo Bashô: Frog Haiku
Translated by Alfred H. Marks  Translated by Tim Chilcott

There once was a curious frog  ancient is the pond –
Who sat by a pond on a log      suddenly a frog leaps – now!
And, to see what resulted,      the water echoes
In the pond catapulted
With a water-noise heard round the bog.
```

Once again, please take note of the space locations. Below is the example but with added padding spaces converted to ~ for demonstration purposes. Additionally, the default column padding of 2 characters was emitted as # characters.

```
Matsuo Bashô: Frog Haiku~~~~~##Matsuo Bashô: Frog Haiku~~~~~
Translated by Alfred H. Marks~~~~~##Translated by Tim Chilcott~~~~~
~~~~~##~~~~~
There once was a curious frog~~~~~##ancient is the pond ~~~~~~
Who sat by a pond on a log~~~~~##      suddenly a frog leaps – now!
And, to see what resulted,~~~~~##      the water echoes~~~~~
In the pond catapulted~~~~~##~~~~~
With a water-noise heard round the bog.##~~~~~
```

Example 3

This example illustrates output when the right file is longer than the left. The command

```
$ node bin/sidebyside.js content/poetry/basho_chilcott.txt
content/poetry/basho_marks.txt
```

prints the string

```
'Matsuo Bashô: Frog Haiku      Matsuo Bashô: Frog Haiku
\nTranslated by Tim Chilcott    Translated by Alfred H. Marks      \n
\nancient is the pond –        There once was a curious frog      \n
suddenly a frog leaps – now!   Who sat by a pond on a log        \n
the water echoes              And, to see what resulted,          \n
In the pond catapulted        \n                                With
a water-noise heard round the bog.\n'
```

which renders as

Matsuo Bashô: Frog Haiku	Matsuo Bashô: Frog Haiku
Translated by Tim Chilcott	Translated by Alfred H. Marks
ancient is the pond –	There once was a curious frog
suddenly a frog leaps – now!	Who sat by a pond on a log
the water echoes	And, to see what resulted,
	In the pond catapulted
	With a water-noise heard round the bog.

Example 4

This example shows the behavior of the -S (column size) flag, which has a value of 19 in this example. The command

```
$ node bin/sidebyside.js -s 19 content/poetry/basho_rexroth.txt
content/poetry/basho_marks.txt
```

prints the string

'Matsuo Bashô: Frog	Matsuo Bashô: Frog \nHaiku	Haiku
\nTranslated by Kenne	Translated by Alfre\nnth Rexroth	d H. Marks
\n	\nAn old pond –	There once was

```

a cu\nThe sound          rious frog          \nOf a diving frog.    Who sat by
a pond o\n                n a log              \n                And, to
see what re\n            sulted,              \n                In
the pond catapult\n            ted              \n
With a water-noise \n        heard round the bog\n
.                          \n'

```

which renders as

Matsuo Bashô: Frog	Matsuo Bashô: Frog
Haiku	Haiku
Translated by Kenne	Translated by Alfre
th Rexroth	d H. Marks

An old pond –	There once was a cu
The sound	rious frog
Of a diving frog.	Who sat by a pond o
	n a log
	And, to see what re
	sulted,
	In the pond catapult
	ted
	With a water-noise
	heard round the bog
	.

The two output columns are exactly 19 characters wide and content wraps, preserving the feel of newlines. No other content was adjusted (i.e., words were not hyphenated or preserved in any way).

Example 5

This example shows the behavior of the -s flag, the -c flag (padding between columns) and three text files. The command

```
$ node bin/sidebyside.js -s 14 -c 3 content/poetry/basho_rexroth.txt
content/poetry/basho_marks.txt content/poetry/basho_chilcott.txt
```

prints the string

```

'Matsuo Bashô: Frog    Matsuo Bashô: Frog \nHaiku                Haiku
\nTranslated by Kenne  Translated by Alfre\nth Rexroth            d H. Marks
\n                                \nAn old pond –          There once was
a cu\nThe sound          rious frog          \nOf a diving frog.    Who sat by
a pond o\n                n a log              \n                And, to
see what re\n            sulted,              \n                In
the pond catapult\n            ted              \n
With a water-noise \n        heard round the bog\n
.                          \n'

```

which renders as

Matsuo Bashô:	Matsuo Bashô:	Matsuo Bashô:
Frog Haiku	Frog Haiku	Frog Haiku
Translated by	Translated by	Translated by
Kenneth Rexrot	Alfred H. Mark	Tim Chilcott
h	s	
		ancient is the
An old pond –	There once was	pond –

The sound	a curious fro	suddenly
Of a diving fr	g	a frog leaps –
og.	Who sat by a p	now!
	ond on a log	the
	And, to see wh	water echoes
	at resulted,	
	In the pond ca	
	tapulted	
	With a water-n	
	oise heard rou	
	nd the bog.	

The three output columns are 14 characters wide (due to the -S 14 flag) and padded three spaces apart (due to the the -C 3 flag).