

5		-	const consumptionChars = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#%^&*()_+1234567890-=[]{};\\':"../>?,<\\ ';
	5	+	const consumptionCharacters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#%^&*()_+1234567890-=[]{};\\':"../>?,<\\ ';
6	6		
7	7		// Testing against codes found at: <a href="http://ascii-table.com/ansi-escape-sequences-vt-100.php">http://ascii-table.com/ansi-escape-sequences-vt-100.php</a>
8	8		test('match ansi code in a string', t => {
⌵ ⌶			@@ -67,7 +67,7 @@ for (const codeSet of Object.keys(ansiCodes)) {
67	67		const skipText = skip ? '[SKIP] ' : '';
68	68		const ecode = `\\u001B\${code}`;
69	69		
70		-	test(`\${skipText}\${code} → \${codeInfo[0]}`, t => {
	70	+	test(`\${codeSet} - \${skipText}\${code} → \${codeInfo[0]}`, t => {
71	71		if (skip) {
72	72		t.pass();
73	73		return;
⌵ ⌶			@@ -79,13 +79,13 @@ for (const codeSet of Object.keys(ansiCodes)) {
79	79		t.is(string.replace(ansiRegex(), ''), 'hello');
80	80		});
81	81		
82		-	test(`\${skipText}\${code} should not overconsume`, t => {
	82	+	test(`\${codeSet} - \${skipText}\${code} should not overconsume`, t => {
83	83		if (skip) {
84	84		t.pass();
85	85		return;
86	86		}
87	87		
88		-	for (const c of consumptionChars) {
	88	+	for (const c of consumptionCharacters) {