int		examples
`5`	🟩	5
`042`	🟩	42
`400`	🟩	400

int		counterexamples

`54a`	🟩	**Parsing error:** The parser did not consume the entire input.
		`1 | 54a`
		`~~|^` Surplus characters
		`| Valid <int>`
		Hint: halted due to the following:
		`1 | 54a`
		`^` Regex does not match
		While trying to parse: `<int>`.

hexdigit		examples
`8`	🟩	8
`F`	🟩	F
`a`	🟩	a

hex		examples
`0x42`	🟩	0x42
`0xF`	🟩	0xF

identchar		examples
`a`	🟩	a
`F`	🟩	F

ident		examples
`x`	🟩	x
`foo`	🟩	foo

comma		examples
`,`	🟩	
`␣␣,␣`	🟩	

maybehex		examples
∅	🟩	`()`
`0x42`	🟩	`("0x42",)`

maybehex		counterexamples
`0x`	🟩	**Parsing error:** The input does not match the expected format. `1 | 0x` `^` Regex does not match While trying to parse: `<maybehex>` → `<hex>` → `<hexdigit>`.

value		examples
`5`	🟩	5
`x`	🟩	x

value		counterexamples

`5a`	🟩	**Parsing error:** The parser did not consume the entire input.
		`1 | 5a`
		   `~|^` Surplus characters
		     `|` Valid `<value>`
		Hint: halted due to the following:
		`1 | 5a`
		      `^` Regex does not match
		While trying to parse: `<value>` → `<int>`.

`x4`	🟩	**Parsing error:** The parser did not consume the entire input.
		`1 | x4`
		   `~|^` Surplus characters
		     `|` Valid `<value>`
		Hint: halted due to the following:
		`1 | x4`
		      `^` Regex does not match
		While trying to parse: `<value>` → `<ident>` → `<identchar>`.

`0xz`	🟩	**Parsing error:** The input does not match the expected format. `1 | 0xz` `^` Regex does not match While trying to parse: `<value>` → `<hex>` → `<hexdigit>`.

list		examples
`[1,␣2,␣3]`	🟩	`(1, 2, 3)`
`[42␣,␣54␣,␣␣␣␣]`	🟩	`(42, 54)`

list		examples
`[0x456,]`	🟩	`("0x456",)`

list		counterexamples
`42`	🟩	**Parsing error:** The input does not match the expected format. `1	42`    ^ Can't match string "[" While trying to parse: `<list>`.
`[42,␣54a]`	🟩	**Parsing error:** The input does not match the expected format. `1	[42,␣54a]`     ^ Can't match string "]" While trying to parse: `<list>`.
`[42]`	🟩	**Parsing error:** The input does not match the expected format. `1	[42]`    ^ Can't match string "," While trying to parse: `<list>` → `<comma>`.

parsing	🟩	🟥
29	29	0

`(true, "0x45")`