1) function call in expression: 100%

2) x = 12;

if (x >= (y / 4)) { <<= detect y not defined

x = foobar();

}

3) const x = 12;

x = 100

# General Issues

1. Add ***print(...a)*** support (**100**%)

2. better variable output (for all defined variables) (90%: maybe better formatting)

3. add ***for of*** support (**can delay**)

4. distinguish “warning” from “error”, the former shall not stop parsing activity (**can delay**)

5. console output the latest should scroll up when printing (**100**%)

6. formatter error should be also in dialog box (**100**%)

# Parsing Issues

---------------

principal: a. resolve asap: a

A: when op2 .priority > op1

B: when ")" is met, resolve back till "("

C: if unary back is met, resolve backward immediately

D: IF UNARY FRONT IS MET, push var, and wait to see the next

E: Skip comments (done)

10: parsing forEach =>

11: while () {} => endless loop detection

12: switch () { case: default}

13: do {} while()

14: lambd () => expre, a => { block }, (a, b, c) => {}

# Code formatter issues:

1: remove parser added comments: /\* #$%^&\*: blabla \*/ (100%)

2: maintain quote type (100%)

3: the following statement complains error:

x = {

x: 1,

y: 2

}

print (x)

------------->

/\* noneedtokeet Error: Undefined valrable found at line: 5, position: 29, found token <<< x >>> \*/

x = {

x: 1,

y: 2

}

print (x)

4: x = ++a + + b // should have no space between + and b

# BUGS:

Current:

## Bug 1

Const x = 1; ++x should cause error in formatter (Fixed)

## Bug 2

Multiple error messages printed (Fixed: in method: showDiagnoticMessage)

## Bug 3

showDiagnoticMessage error for the following code (Fixed: in where the error was thrown)

const x; // error for const no-assignment

let y;

x = 10;

print(x+y)

## Bug 4

const xxx = 100;

let yyy = "abc";

const zzz = [1, 2, 3];

t= 100000;

const www = {

x: xxx,

y: yyy

} => got error

## Bug 4

x = 12;

const www = {

x: x,

y: 12,

z: {

z: 1

}

};

print ("www is =", www) => cannot print z property

1) (Fixed) IF STATEMENT does not complain about undefined variables

- cause: variable check is only in binary resolver

- fix: make sure assignment still works after this

2) (Fixed) const x = 1;

let y = 2;

x = 10; // error for const assignment

print(x+y)

got weir error message: expecting "("

3) (Fixed) INORE Grammar check inside /\* \*/

4) (Fixed) ! operator does not work for parser: x = a ! b => should be invalid

5) save exception

6) (Fixed) student -> teacher code sync does not work

7) object syntax, by pass: performance.now()

8) (Fixed) (x != 0) and (x !== 0) does not work

9) (Fixed) ++2 should not be working

10) // conment should have a space in front: x = 100;// 100 should be => x = 100; // 100

12) support multiple accessory syntax: a.b.c or a.b.foo()

13) performance .now(); should be performance.now();

14) (Completed) support true false as a known value

15) "," should have a space after it: a,b,c -> a, b, c

16)(Fixed for var scoping added in expressState) x = 3;

while (x-- > 0) {

for (let i = 0; i < x; i++) {

let y = i \* 10;

print (y);

}

print (x);

print (i); => i should be undefined

}

17) (Fixed) add endless loop detection for "for" loop

- for loop analys til it's loop body ends "}". reference "while" loop

17.1) no-body loop endless detection (delayed)

x = 1;

while (x++ < 10) ;

for(let i = 0; i < 100; i--);

17.2) nested endless detection (Delayed)

while (x > 0) {

while (y > 0) {

}

}

18) (Delayed) do {} while() loop analysis

19) switch (case) analysis

20)(Fixed) print all the const / let / global in the top scope

21) add nested loop analysis and endless loop prevention

22) (FIXED) expression does not end if no ";" or the last expression is an assignment, such as x = 100; y = 12; => y is not parsed

23) (Fixed) array subscription assignment does not work:

arr[pos1] = arr[pos2];

arr[pos2] = x;

Needed features:

---------------------------

1) teacher notes per class

2) teacher set invisible coe (for testing purpose)

3) load my code into running context

4) view running context and manage it (by add and / removing code by functions)

Teaching content:

------------------------

Varaible, value, and data type

value: "abc" - string type

'abc' - string type

`abc` - string type

100 - integer number type

1.245 - float number type

[1, 2, 3] - array type

{name: 'Kevin Yang', age: 12} - object type

comment:

/\* \*/ and //

seperator:

CR, , ; :

CODE BLOCK: () -> EXPRESSION

{ } -> BLOCK SCOPE

Teach this after students can werite simple code, before function is introduced