

A lightweight JavaScript CAS for comparing expressions and equations. It is used throughout [Khan Academy](#)'s interactive exercises.

What can it do?

It can parse plain text math, LaTeX, or a mix of both:

```
var expr = KAS.parse("3x \\frac{42}{42} sin^2y").expr;
expr.print();
// "3*x*42/42*sin(y)^(2)"
```

It can evaluate expressions:

```
var expr = KAS.parse("(x^2+y^2)^.5").expr;
expr.eval({x: 3, y: 4});
// 5
```

It can compare expressions and equations:

```
var expr1 = KAS.parse("(1-x)(-1-6x)").expr;
var expr2 = KAS.parse("(6x+1)(x-1)").expr;
KAS.compare(expr1, expr2).equal;
// true

var eq1 = KAS.parse("2w+50/w=25").expr;
var eq2 = KAS.parse("w(12.5-w)=25").expr;
KAS.compare(eq1, eq2).equal;
// true
```

It can perform basic transforms that always simplify an expression:

```
var expr = KAS.parse("1+1+x+x+x+y").expr;
expr.collect().print();
// "2+3*x+y"

var expr = KAS.parse("b^(2*y*log_b x)").expr;
expr.collect().print();
// "x^(2*y)"
```

It can perform non-simplifying transforms on an expression:

```
var expr = KAS.parse("ab(c+d)e^f").expr;
expr.print();
```

```
// "a*b*(c+d)*e^(f)"
expr.expand().print();
// "a*b*e^(f)*c+a*b*e^(f)*d"
expr.expand().factor().print();
// "a*b*e^(f)*(c+d)"
```

It can combine the above abilities to perform powerful simplification:

```
var expr = KAS.parse("((nx^5)^5)/(n^2x^2)^-3").expr;
expr.print();
// "(n*x^(5))^5*(n^(-2)*x^(2))^(-1*-3)"
expr.simplify().print();
// "n^(-1)*x^(31)"

var expr = KAS.parse("(15np-25mp)/(15p^2-5p)+(20mp+10p^2)/(15p^2-5p)").expr;
expr.print();
// "(15*n*p+-25*m*p)*(15*p^(2)+-5*p)^(-1)+(20*m*p+10*p^(2))*(15*p^(2)+-5*p)^(-1)"
expr.simplify().print();
// "(-1+3*p)^(-1)*(3*n+-1*m+2*p)"
```

How to build the library

```
npm install
npm run build
```

How to build the parser

First, make any changes in `src/parser-generator.js`

```
npm install
npm run build:parser
```

License

[MIT License](#)