



Минифицируем приватные поля в TypeScript

Алексей Гусев

Старший разработчик интерфейсов

 alex-gusev@yandex-team.ru
 [@alexey_gusev](https://twitter.com/alexey_gusev)

Яндекс.Эфир: фильмы, сериалы +

Ведите поисковый запрос в Google или укажите URL.

Яндекс Эфир Передача, канал Найти плюс mad.gooze

Что посмотреть Телеканалы Фильмы Сериалы Мультфильмы Мульсериалы Спорт Блогеры Музыка Игры и Киберспорт КиноПоиск

Список каналов Наша Сибирь | Телепрограмма

РЕКОМЕНДАЦИИ Мой эфир Наша Сибирь Москва 24 Развлекательный earthTV Звезда Семейные комедии Мультфильмы Старт КВН Успех МОЙ РЕГИОН Москва 24

Чат трансляции | прекрасный телеканал с... Спасибо ❤ Сообщение удалено модератором в связи с нарушением правил сообщества.

Артём В. Доброго Славяне! Спаси Бог Славяне! 21 августа

Хмель Дмитрий Вчера

Екатерина Н. Какая красота, господи!

Напишите сообщение...

Смотрите также Мыс Кекурский 22 августа Мыс Кекурский 22 августа

расскажу вам лично. Оно включает в себя как изменения во внешнем виде Браузера, так и новую полезную функцию...

БЛОГ ЯНДЕКС.БРАУЗЕРА

Лучшие киборги из кинофантастики. Часть II Кинофантастика последних десятилетий очень богата на тему киберпанка, научной фантастики и экранизаций комиксов. Так что...

КИНОТЕРАПИЯ

По уши в работе КРАСОТА СПАСАЕТ МИР

Ученые показали возможную будущую Землю Будет ли компьютер на Родине Андрей Судальцов: в гибели МИ-17 штурмовой транспортного вертолета

https://zen.yandex.ru/media/id/5bb63eeccc626f000b84808cb/5d5e5fd39e5aa0ba5e9e29a7?from=feed&utm_referrer=https%3A%2F%2Fzen.yandex.com&id=665573680.438.1566549674816.36413

Яндекс Найти

Поиск Картинки Видео Карты Маркет Новости Эфир Коллекции Знатки Услуги Ещё

Главное Санкт-Петербург Рекомендуем Политика Общество Экономика В мире Спорт Происшествия Культура Технологии Наука Авто

Известия С Джеки Чана и Шварценеггера попросили взыскать 216,6 млн рублей

Российская компания подала в суд на Джеки Чану и Шварценеггера

0:03 из 0:56

44 фото Ещё 1 видео ...

Конкурсный управляющий «Русской фильм группы» (РФГ) обратился с иском в Арбитражный суд Москвы о взыскании с китайской компании China Film Group, а также актеров Джеки Чана и Арнольда Шварценеггера 216,6 млн рублей, сообщает в пятницу, 23 августа, газета «Ведомости».

Яндекс Дзен Лента Подписки Каналы Скачать приложение

расскажу вам лично. Оно включает в себя как изменения во внешнем виде Браузера, так и новую полезную функцию...

БЛОГ ЯНДЕКС.БРАУЗЕРА

Лучшие киборги из кинофантастики. Часть II Кинофантастика последних десятилетий очень богата на тему киберпанка, научной фантастики и экранизаций комиксов. Так что...

КИНОТЕРАПИЯ

По уши в работе КРАСОТА СПАСАЕТ МИР

Ученые показали возможную будущую Землю Будет ли компьютер на Родине Андрей Судальцов: в гибели МИ-17 штурмовой транспортного вертолета

https://zen.yandex.ru/media/id/5bb63eeccc626f000b84808cb/5d5e5fd39e5aa0ba5e9e29a7?from=feed&utm_referrer=https%3A%2F%2Fzen.yandex.com&id=665573680.438.1566549674816.36413

КиноПоиск HD Подписка Покупка и аренда Мои покупки

Аккумулятор Xiaomi Mi Power market.yandex.ru

Вы уже слышали об Устройстве IQOS? iqostestdrive.ru

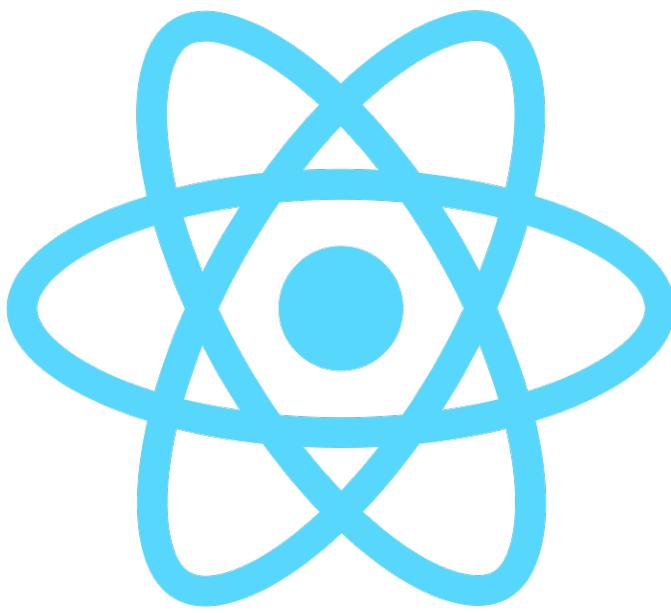
0:01 из 2:11:50

Аудиодорожки Русский, Английский

Про видеоплеер



webpack





github.com/google/shaka-player



Google
Closure Compiler

Add a URL: 

Add

Example: http://www.example.com/bigfile.js

Optimization: Whitespace only Simple Advanced[Which optimization is right for my code?](#)Formatting: Pretty print Print input delimiter[Compile](#)[Reset](#)

```
// ==ClosureCompiler==  
// @compilation_level SIMPLE_OPTIMIZATIONS  
// @output_file_name default.js  
// ==/ClosureCompiler==  
  
// ADD YOUR CODE HERE  
function hello(name) {  
  alert('Hello, ' + name);  
}  
hello('New user');
```

Compilation was a success!

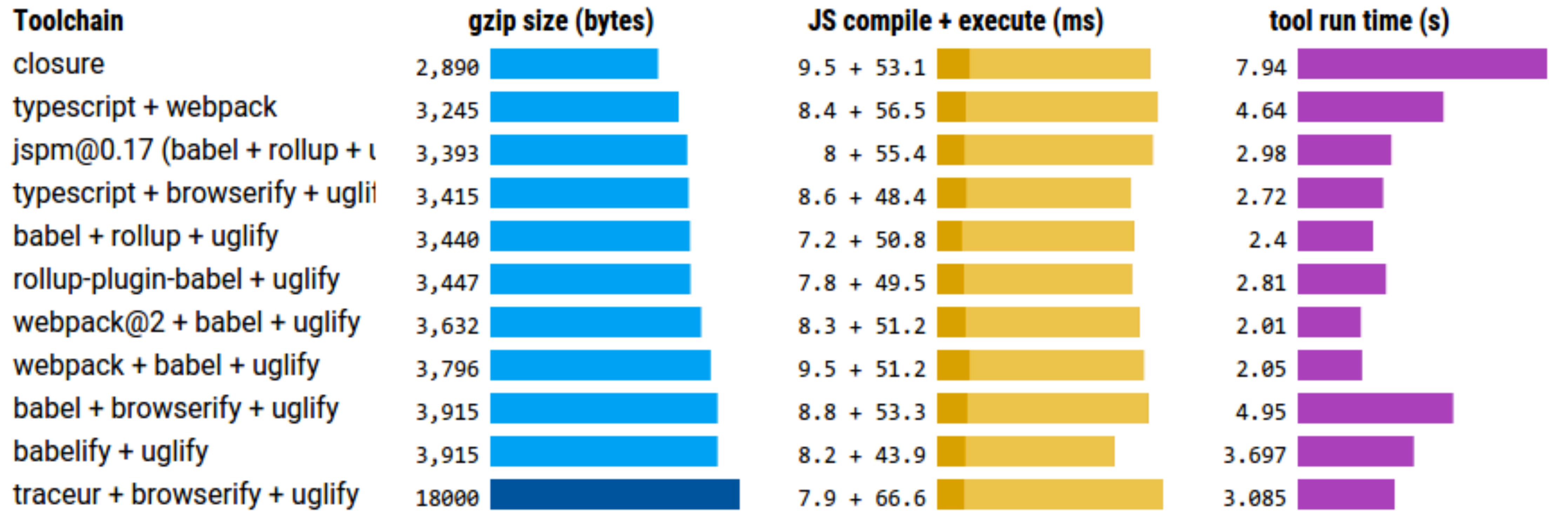
Original Size: 102 bytes gzipped (92 bytes uncompressed)

Compiled Size: 68 bytes gzipped (55 bytes uncompressed)

Saved 33.33% off the gzipped size (40.22% without gzip)

The code may also be accessed at [default.js](#).[Compiled Code](#)[Warnings](#)[Errors](#)[POST data](#)

```
function hello(a){alert("Hello, "+a)}hello("New user");
```



npmjs.com/package/webpack-closure-compiler

Add a URL: ▼

Example: <http://www.example.com/bigfile.js>

Optimization: Whitespace only Simple Advanced
[Which optimization is right for my code?](#)

Formatting: Pretty print Print input delimiter

[Reset](#)

```
// ==ClosureCompiler==  
// @compilation_level SIMPLE_OPTIMIZATIONS  
// @output_file_name default.js  
// ==/ClosureCompiler==  
  
// ADD YOUR CODE HERE  
function hello(name) {  
  alert('Hello, ' + name);  
}  
hello('New user');
```

Compilation was a success!

Original Size: 102 bytes gzipped (92 bytes uncompressed)
Compiled Size: 68 bytes gzipped (55 bytes uncompressed)
Saved 33.33% off the gzipped size (40.22% without gzip)

The code may also be accessed at [default.js](#).

Compiled Code		Warnings
Errors	POST data	
function hello(a){alert("Hello, "+a)}hello("New user");		

Инлайн функций

```
function f() {  
    console.log("f");  
}  
f();
```

```
console.log("f");
```

Удаление dead code

```
class A {  
    method() {  
        console.log("A.method");  
    }  
}  
  
class B {  
    deadCode() {  
        console.log("this is dead code!");  
    }  
}  
  
new A().method();
```

```
console.log("A.method");
```

Инлайн переменных

```
function f(x) {  
    console.log("got: " + x);  
}  
  
var someVar = 9001;  
f(someVar);
```

```
console.log("got: 9001");
```

Минификация полей объектов

```
class A {  
    constructor() {  
        this.prop = 100;  
    }  
}  
  
let a = new A();  
a.prop++;  
console.log(a.prop);
```

```
var a = new function (){  
    this.a = 100  
};  
  
a.a++;  
console.log(a.a);
```

И это — далеко не всё

medium.com/@thomasdeegan/10-optimizations-closure-compiler-can-do-and-uglify-friends-cant-6dca4232299f

SHUT UP AND



TAKE MY MONEY!

TypeScript: private

```
class Foo {  
    private bar(): void {  
        // ...  
    }  
}
```

```
class Foo {  
    bar() {  
        // ...  
    }  
}
```

TypeScript: private

```
class Foo {  
    private bar(): void {  
        // ...  
    }  
}
```

```
const foo = new Foo();  
// Property 'bar' is private  
// and only accessible  
// within class 'Foo'.  
foo.bar();
```

```
class Foo {  
    bar(): void {  
        // ...  
    }  
}
```

Минификация приватных полей

```
type MyButtonProps = { text: string };

class MyButton extends React.PureComponent<MyButtonProps> {
    private clickCount = 0;

    public render() {
        return <button
            onClick={() => this.clickCount++}>
            {this.props.text}
        </button>;
    }
}
```

Минификация приватных полей

```
class MyButton extends React.PureComponent {  
  clickCount = 0;  
  
  render() {  
    return <button  
      onClick={() => this.clickCount++}>  
      {this.props.text}  
    </button>;  
  }  
}
```

```
class MyButton extends React.PureComponent {  
  a = 0;  
  
  render() {  
    return <button  
      onClick={() => this.a++}>  
      {this.props.text}  
    </button>;  
  }  
}
```

Используем Closure Compiler
в Advanced-режиме как
минификатор?



Advanced-оптимизации

```
let x = { foobar: 1 };  
let y = x.foobar;
```

Advanced-оптимизации

```
let x = { foobar: 1 };
```

```
let y = x.foobar;
```

// *foobar* будет переименовано

// тут будет ошибка!

```
if ('foobar' in x) { ... }
```

// тут тоже!

```
x['foobar'];
```

Advanced-оптимизации

```
let foo = {};
foo.bar = function (a) {
  this._a = a;
};
foo.bar('hello');
```

```
var foo$bar = function (a) {
  // this теперь глобальный!
  this._a = a;
};
foo$bar('hello');
```

Advanced-оптимизации

player.play()

player.a()



Advanced-оптимизации — не безопасные

*... если вы не используете
Closure Compiler
на полную мощность*



webpack

+

BABEL

=



+



Модульная система

```
goog.provide('myproject.start');

goog.require('goog.dom');
goog.require('goog.dom.TagName');

myproject.start = function() {
  var newDiv = goog.dom.createDom(goog.domTagName.H1, {'style':
'background-color:#EEE'},
  'Hello world!');
  goog.dom.appendChild(document.body, newDiv);
};

goog.exportSymbol('myproject.start', myproject.start);
```

Транспиляция и полифилы

```
java -jar compiler.jar --js hello.js  
--language_out ECMASCRIPT5
```

```
java -jar compiler.jar --js hello.js  
--language_out ECMASCRIPT_2016
```

Аннотации типов в JSDoc

```
/**  
 * Queries a Baz for items.  
 * @param {number} groupNum Subgroup id to query.  
 * @param {string/number/null} term An itemName,  
 * or itemId, or null to search everything.  
 */  
goog.Baz.prototype.query = function(groupNum, term) {  
    ...  
};  
  
function foo(/** number */ a, /** number */ b) {  
    return a - b + 1;  
}
```



webpack

BABEL



VS



Я все-таки хочу оптимизации!





Tsickle - TypeScript to Closure Translator

PASSED

build

passing

Tsickle converts TypeScript code into a form acceptable to the [Closure Compiler](#). This allows using TypeScript to transpile your sources, and then using Closure Compiler to bundle and optimize them, while taking advantage of type information in Closure Compiler.

What conversion means

A (non-exhaustive) list of the sorts of transformations Tsickle applies:

- inserts Closure-compatible JSDoc annotations on functions/classes/etc
- converts ES6 modules into `goog.module` modules
- generates externs.js from TypeScript d.ts (and `declare`, see below)
- declares types for class member variables
- translates `export * from ...` into a form Closure accepts
- converts TypeScript enums into a form Closure accepts
- reprocesses all jsdoc to strip Closure-invalid tags

In general the goal is that you write valid TypeScript and Tsickle handles making it valid Closure Compiler code.

Warning: work in progress

We already use tsickle within Google to minify our apps (including those using Angular), but we have less experience using tsickle with the various JavaScript builds that are seen outside of Google.

We would like to make tsickle usable for everyone but right now if you'd like to try it you should expect to spend some time debugging and reporting bugs.

@appmonet/tsickle-loader

```
module.exports = {
  module: {
    rules: [
      {
        test: /\.ts$/,
        use: {
          loader: "tsickle-loader",
          options: {
            tsconfig: path.resolve(__dirname, "tsconfig.json"),
            externDir: "./tmp/externs"
          }
        }
      }
    ]
  }
}
```

Tsickle - TypeScript to Closure Translator

PASSED

build

passing

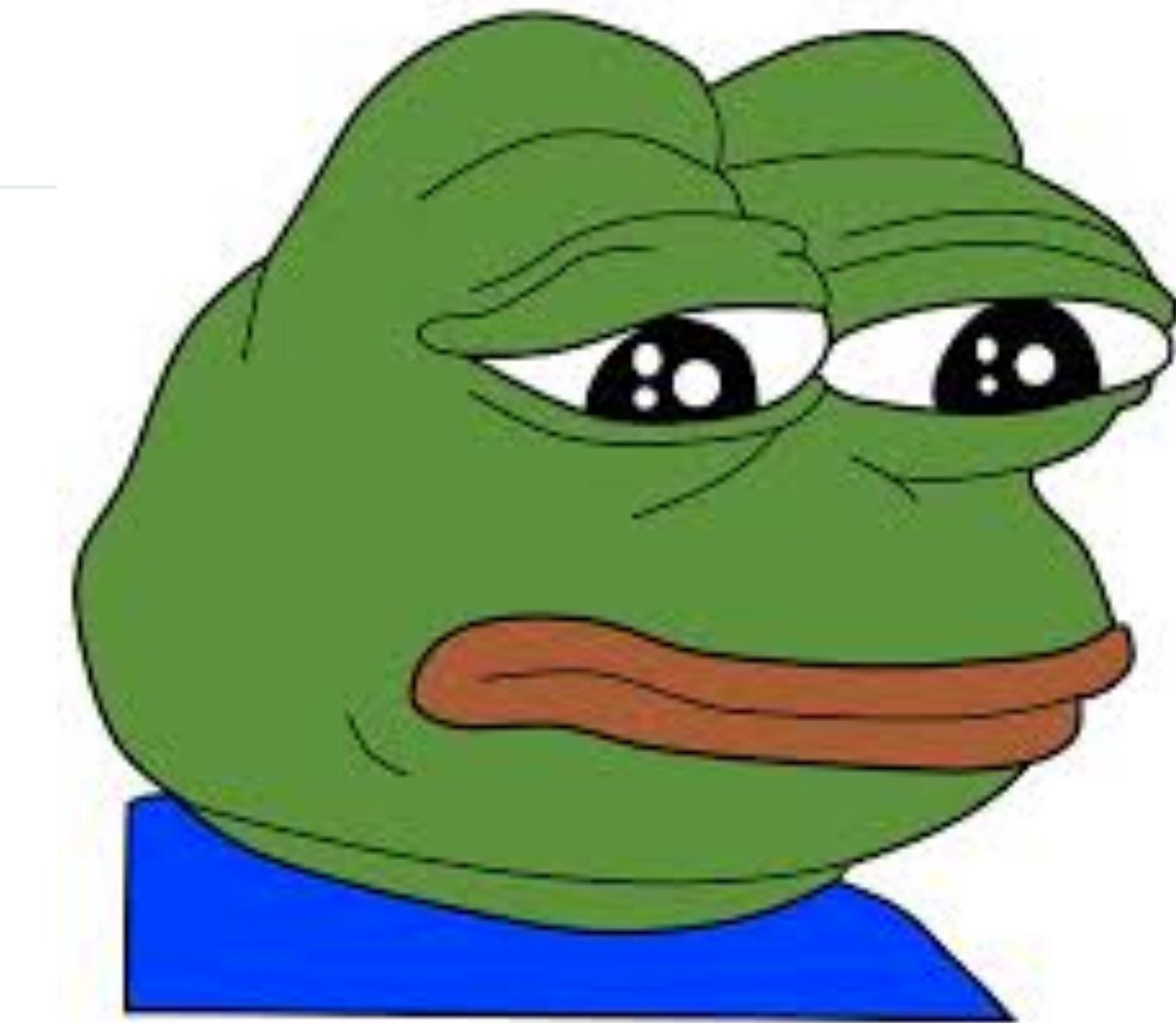
Tsickle converts TypeScript code into a form acceptable to the [Closure Compiler](#). This allows using TypeScript to transpile your sources, and then using Closure Compiler to bundle and optimize them, while taking advantage of type information in Closure Compiler.

What conversion means

A (non-exhaustive) list of the sorts of transformations Tsickle applies:

- inserts Closure-compatible JSDoc annotations on functions/classes/etc
- converts ES6 modules into `goog.module` modules
- generates externs.js from TypeScript d.ts (and `declare`, see below)
- declares types for class member variables
- translates `export * from ...` into a form Closure accepts
- converts TypeScript enums into a form Closure accepts
- reprocesses all jsdoc to strip Closure-invalid tags

In general the goal is that you write valid TypeScript and Tsickle handles making it valid Closure Compiler code.



Warning: work in progress

We already use tsickle within Google to minify our apps (including those using Angular), but we have less experience using tsickle with the various JavaScript builds that are seen outside of Google.

We would like to make tsickle usable for everyone but right now if you'd like to try it you should expect to spend some time debugging and reporting bugs.

Suggestion: minification #8

! Open

RyanCavanaugh opened this issue on 15 Jul 2014 · 145 comments



RyanCavanaugh commented on 15 Jul 2014

Member



...

TypeScript should support emitting minified JavaScript.

There are several different things we could support:

1. Just remove whitespace
2. Minify unobservable identifiers
3. Remove provably dead code
4. Whole-program minification (i.e. closure compiler)
5. (Others?)

404

13

26

4





Будем делать
свой велосипед!

BABEL

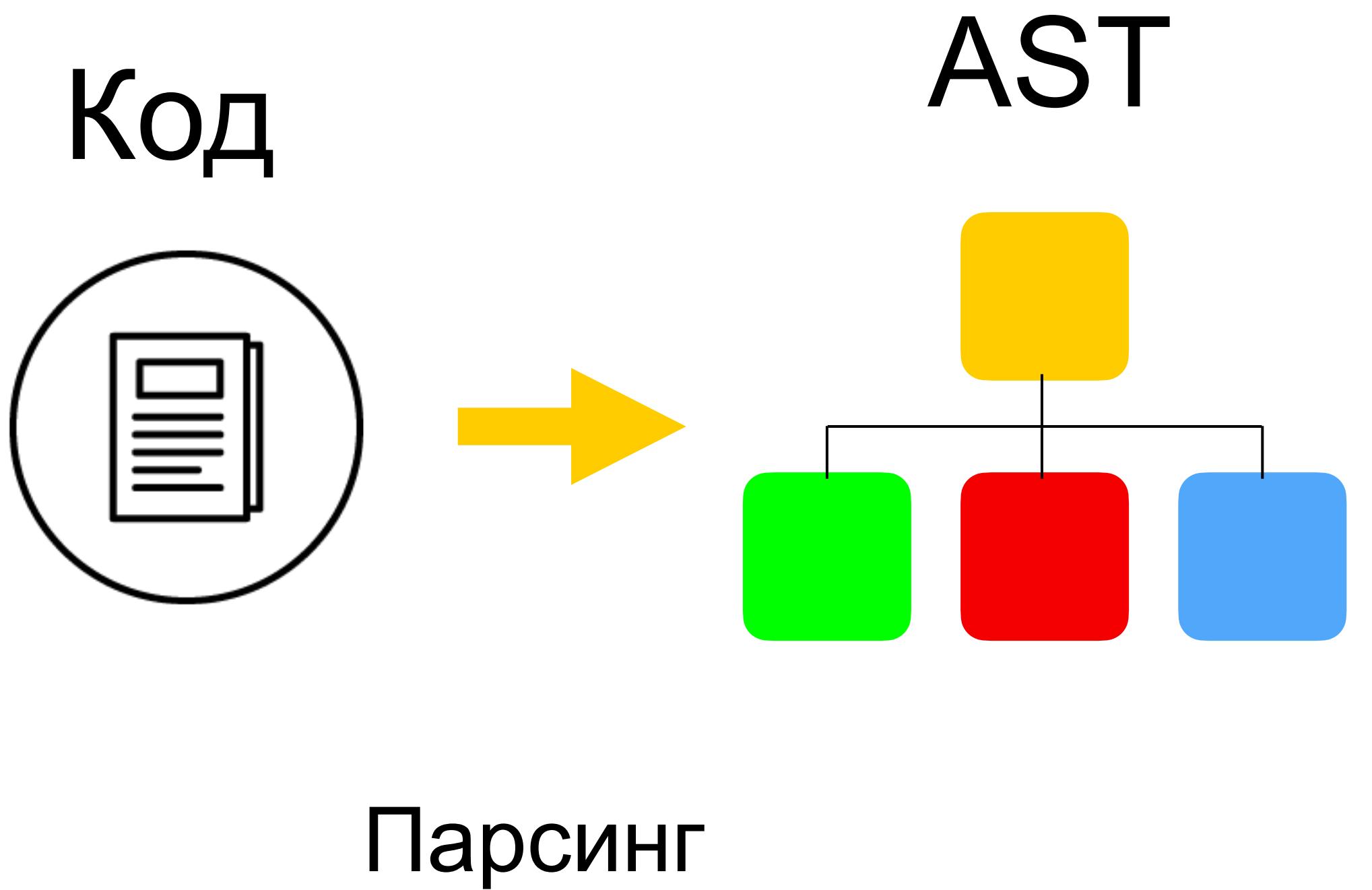
```
{  
  "presets": [  
    "@babel/preset-typescript"  
  ]  
}
```

@babel/preset-typescript

- › Ускорение сборки
- › Конкатенация модулей в webpack
- › Оптимизации
- › Разные сборки под ES5/ES6 браузеры

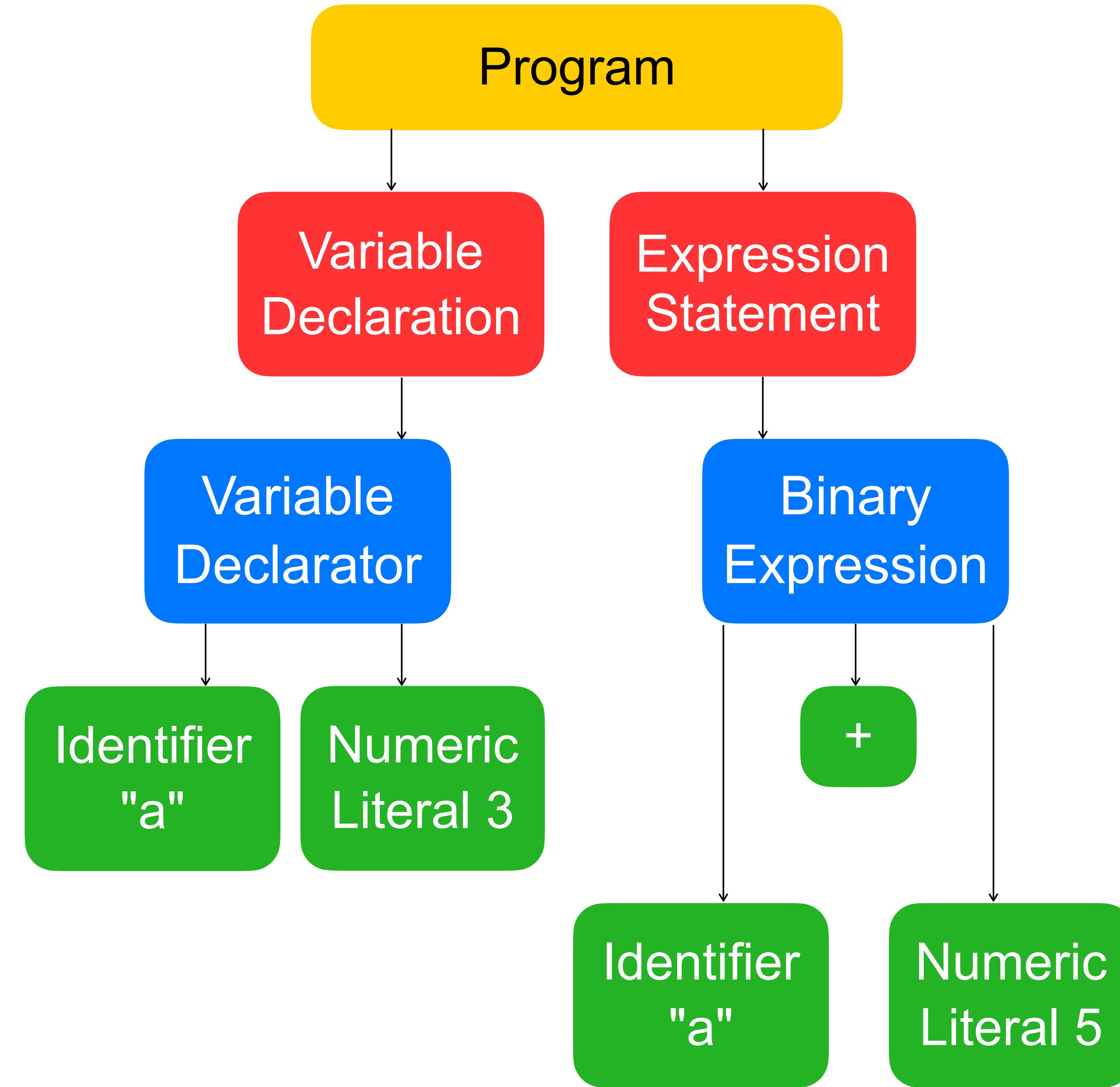
Пишем babel-plugin?
А как?

Как работает babel?



AST

```
var a = 3;  
a + 5
```



```
1 class Foo {  
2     private bar(): void {  
3         //  
4     }  
5 }
```

Tree

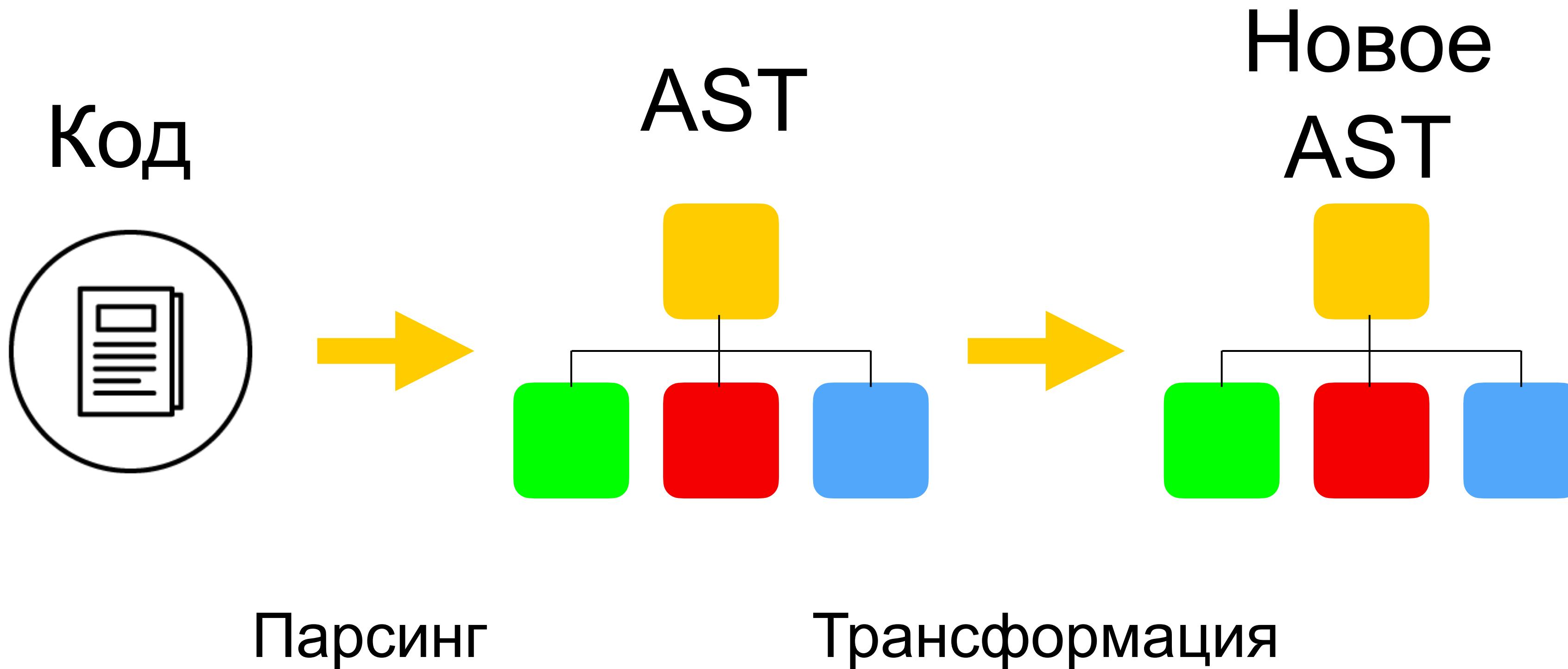
JSON

Parser: [babylon7-7.7.2](#) 8ms

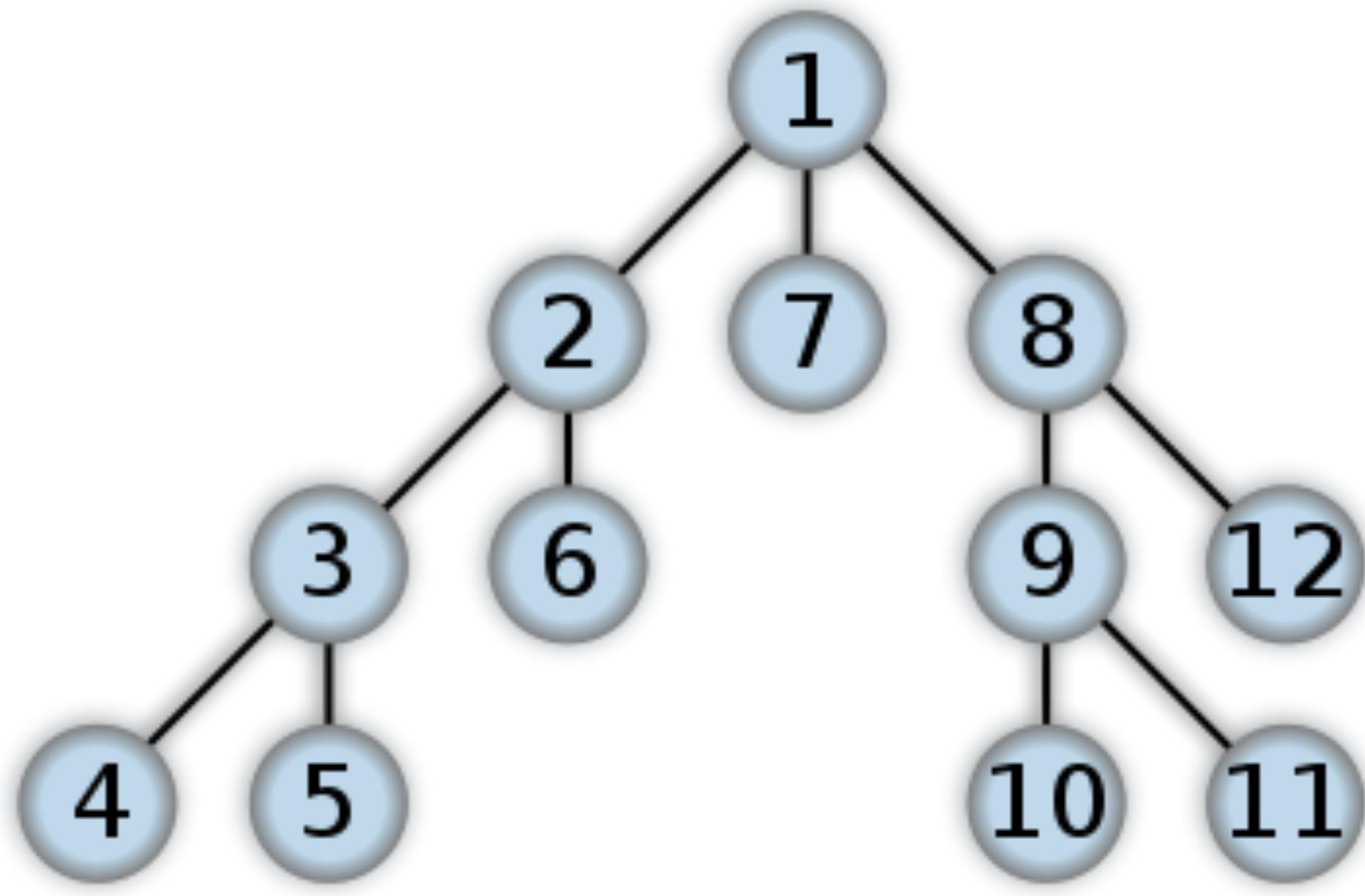
- Autofocus Hide methods Hide empty keys
 Hide location data Hide type keys

```
- program: Program {  
    type: "Program"  
  
    start: 0  
  
    end: 51  
  
    + loc: {start, end}  
    sourceType: "module"  
    interpreter: null  
  
    - body: [  
        - ClassDeclaration {  
            type: "ClassDeclaration"  
  
            start: 0  
  
            end: 51  
  
            + loc: {start, end}  
            + id: Identifier {type, start, end, loc, name}  
            superClass: null  
  
            - body: ClassBody {  
                type: "Classbody"
```

Как работает babel?



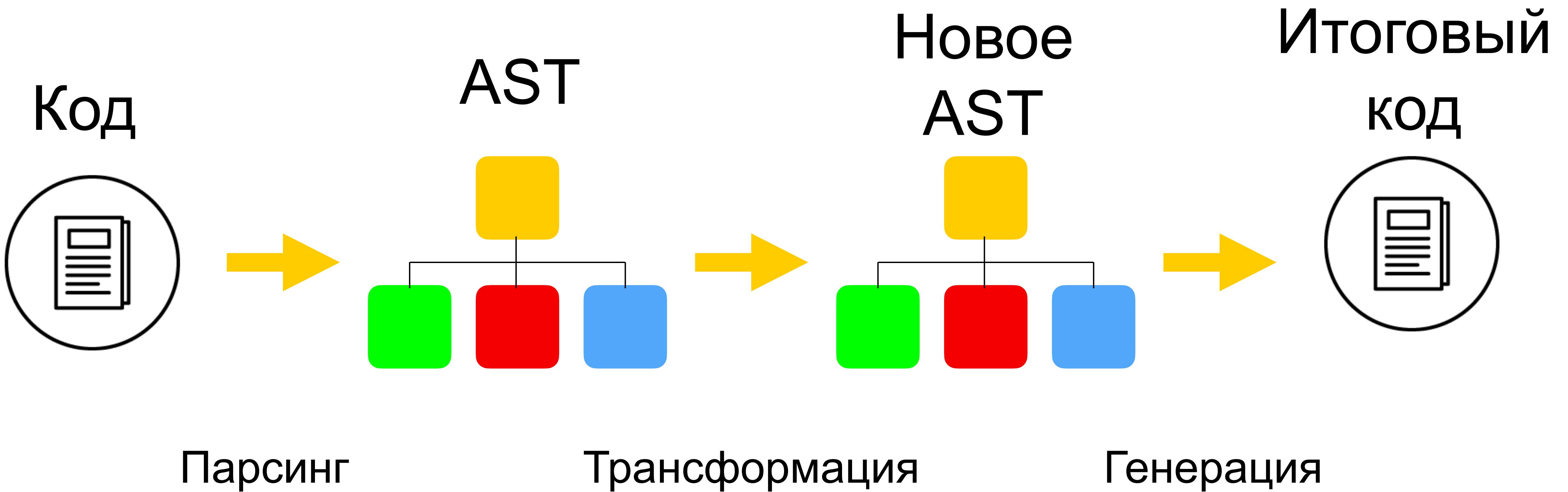
Babel Plugin - Visitor



```
function myCustomPlugin() {  
    return {  
        visitor: {  
            Identifier(path) {  
                // Обрабатываем идентификатор  
            },  
            StringLiteral(path) {  
                // Обрабатываем строку  
            },  
        },  
    };  
}
```

The code illustrates a Babel plugin using the Visitor pattern. It defines a function `myCustomPlugin` that returns an object with a `visitor` property. This `visitor` object contains two methods: `Identifier` and `StringLiteral`. The `Identifier` method is annotated with a comment indicating it processes identifiers, and the `StringLiteral` method is annotated with a comment indicating it processes strings.

Как работает babel?



Напишем плагин, который
модифицирует приватные поля?

Спойлер: ничего не получилось 😞

Минификация приватных полей через babel

```
class Foo {  
    private bar = 0;  
  
    public foobar(): number {  
        return this.bar;  
    }  
}
```

Минификация приватных полей через babel

```
class Foo {  
    private bar = 0;  
  
    public foobar(): number {  
        return this.bar; <-- MemberExpression  
    }  
}
```

Минификация приватных полей через babel

```
class Foo {  
    private bar = 0; <-- VariableDeclarator  
  
    public foobar(): number {  
        return this.bar;  
    }  
}
```

Минификация приватных полей через babel

```
class Foo {  
    private a = 0;  
  
    public foobar(): number {  
        return this.a;  
    }  
}
```



Что-то идет не так...

```
class Foo {  
    private bar = 0;  
  
    public foobar(): number {  
        return this.bar;  
    }  
  
    public otherBar(foo: Foo) {  
        return this.bar + foo.bar;  
    }  
}
```

```
class Foo {  
    private a = 0;  
  
    public foobar(): number {  
        return this.a;  
    }  
  
    public otherBar(foo: Foo) {  
        return this.a + foo.bar;  
    }  
}
```

Что-то идет не так...

```
class Foo {  
    private bar = 0;  
  
    public foobar(): number {  
        return this.bar;  
    }  
  
    public foobar2(): number {  
        const thisFoo = this;  
        return thisFoo.bar;  
    }  
}
```

```
class Foo {  
    private a = 0;  
  
    public foobar(): number {  
        return this.a;  
    }  
  
    public foobar2(): number {  
        const thisFoo = this;  
        return thisFoo.bar;  
    }  
}
```



github.com/mad-gooze/babel-typescript-mangle-private

Suggestion: minification #8

! Open

RyanCavanaugh opened this issue on 15 Jul 2014 · 145 comments



RyanCavanaugh commented on 15 Jul 2014

Member



TypeScript should support emitting minified JavaScript.

There are several different things we could support:

1. Just remove whitespace
2. Minify unobservable identifiers
3. Remove provably dead code
4. Whole-program minification (i.e. closure compiler)
5. (Others?)

404

13

26

4

Evgeniy Timokhov @tradingview



timocov commented on 10 Jun 2019

Contributor



Maybe it'll be useful for someone - I've written a transformer, which (I believe it'll rename privates at all to random name by itself in the future) renames private class' members in a form of `_prefix_originalName` (where `_prefix_` can be changed and by default is `_private_`; `originalName` - original name of the private member), so you can use terser/uglify with a mangle option to minify private properties/methods.

After smoke testing it allows me reduce application's size up to ~15% min and ~5% min.gz (from 43kB to 41kB min.gz or from 186kB to 157kB min in my case).

- <https://github.com/timocov/ts-transformer-minify-privates>
- <https://www.npmjs.com/package/ts-transformer-minify-privates>



8



2

ts-transformer-minify-privates

```
class Foo {  
    private bar = 0;  
  
    public foobar(): number {  
        return this.bar;  
    }  
}
```

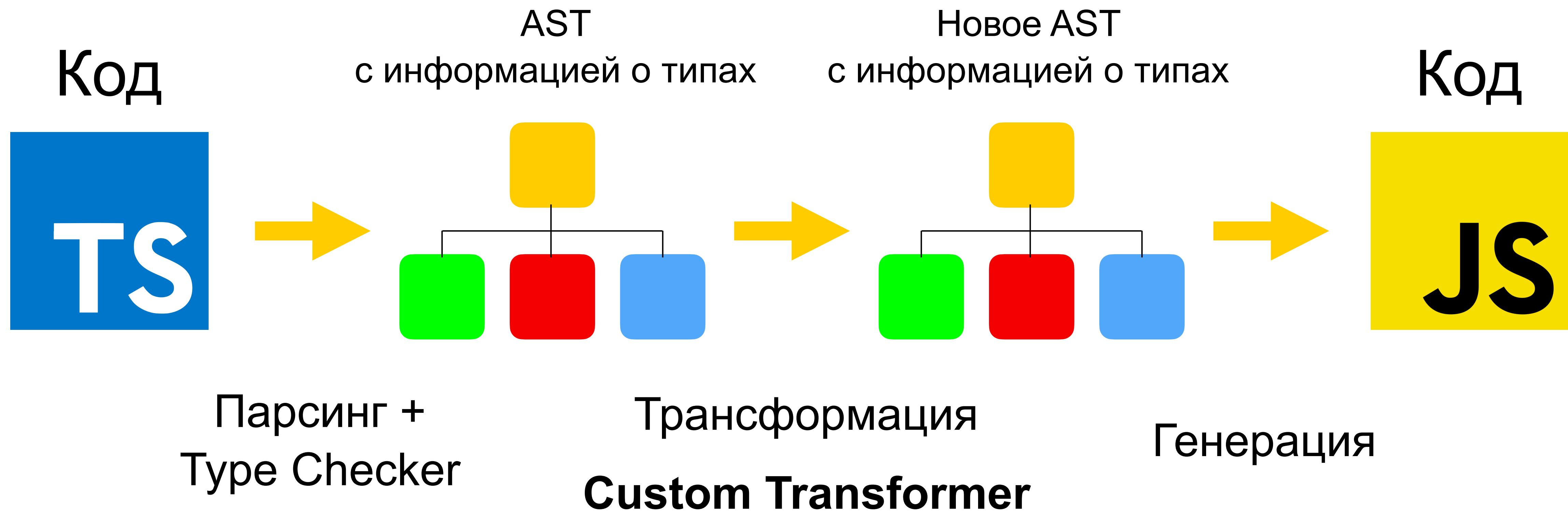
```
class Foo {  
    private _private_bar = 0;  
  
    public foobar(): number {  
        return this._private_bar;  
    }  
}
```

webpack - minimizer

```
{  
  minimizer: [  
    new TerserPlugin({  
      // ...  
      terserOptions: {  
        // ...  
        mangle: {  
          properties: {  
            regex: /^_private_/,  
          },  
          },  
        },  
      },  
    } ),  
  ],  
}
```

Почему это работает?

Как работает TypeScript?



Writing TypeScript custom AST Transformer

levelup.gitconnected.com/writing-typescript-custom-ast-transformer-part-1-7585d6916819

Как использовать TypeScript Transformer?

В CLI TypeScript — нет поддержки

Использование custom transformer с ttypescript

```
// tsconfig.json
{
  "compilerOptions": {
    // ...
    "plugins": [
      { "transform": "custom-transformer" }
    ]
  },
  // ...
}
```

Использование custom transformer с webpack

```
module: {
  rules: [
    {
      test: /\.ts$/,
      loader: 'ts-loader',
      options: {
        getCustomTransformers: program => ( {
          before: [
            customTransformer(program)
          ]
        } )
      }
    }
  ]
}
```

А у нас-то babel!

```
{  
  "presets": [  
    "@babel/preset-typescript"  
  ]  
}
```



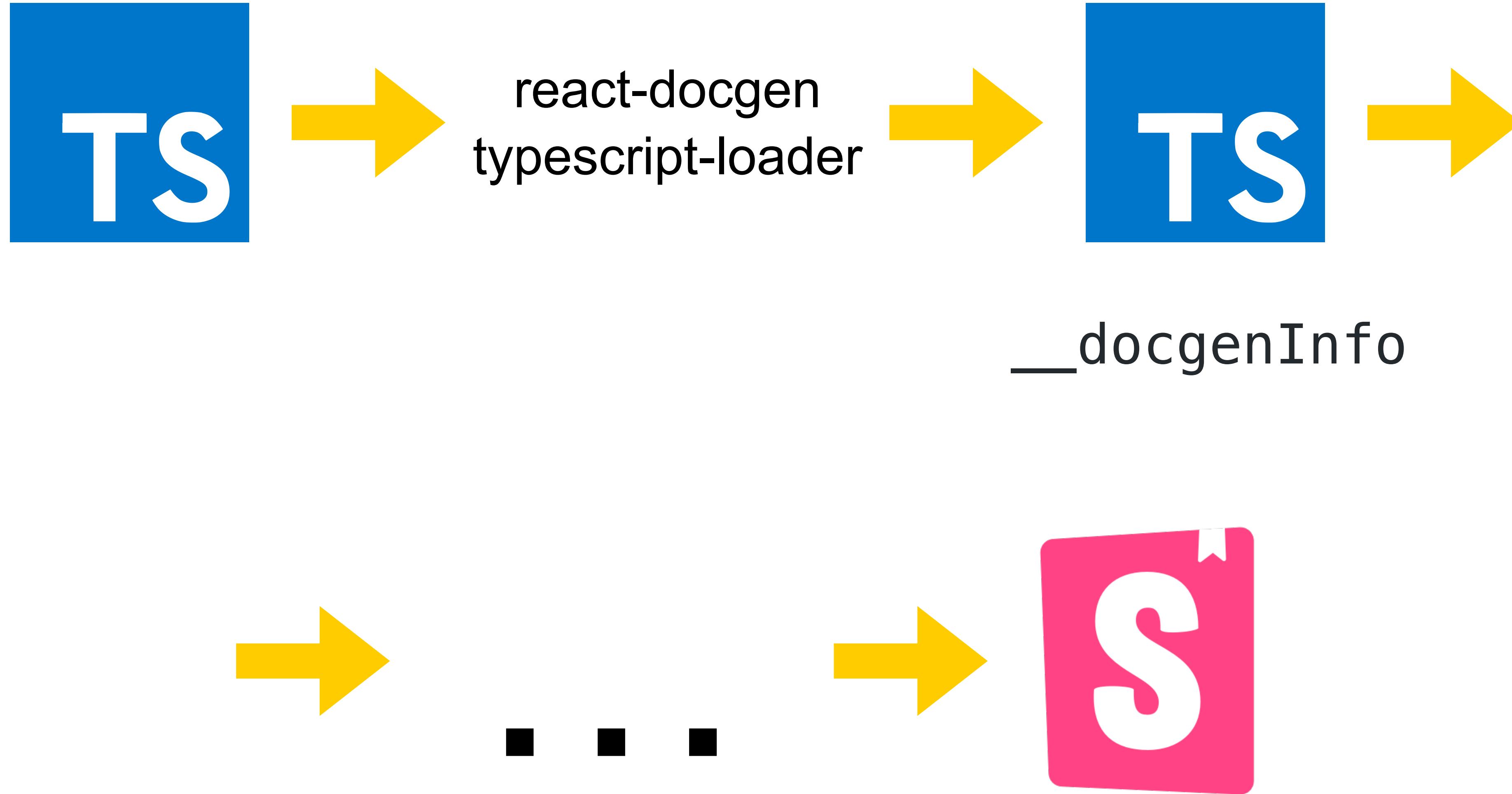
Будем делать
свой велосипед!

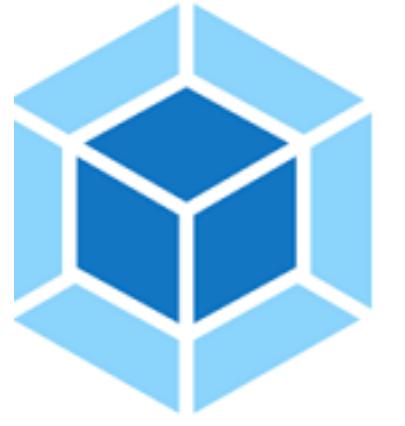
Снова.



react-docgen-typescript-loader







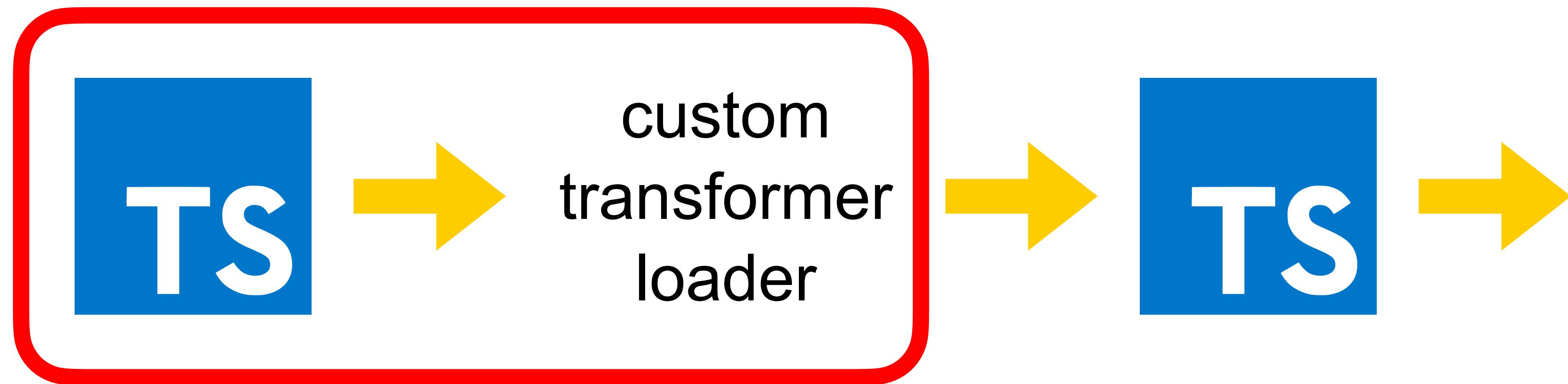
Как написать webpack-лоадер?

webpack loader

```
function loader(source: string): string {
  return 'console.log("Hello World!")';
}
```

webpack loader

```
function loader(  
    this: loader.LoaderContext,  
    source: string,  
) : void {  
    const callback = this.async();  
    // ...  
    callback(newSource);  
}
```



TypeScript Transformation API

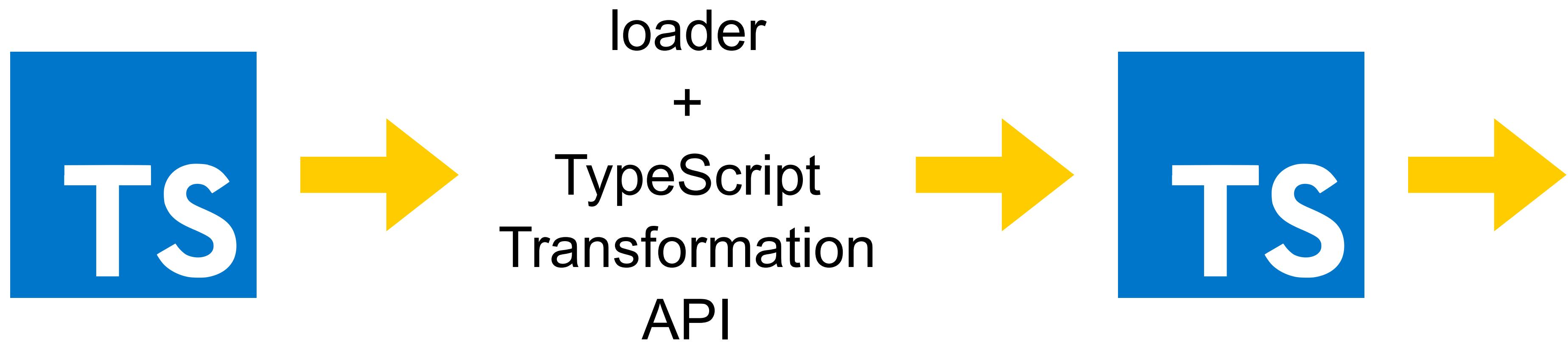
```
import ts from 'typescript';

// получаем ts.Program - объект-коллекцию файлов и настроек компилятора
const program = ts.createProgram([filePath], {});

// получаем AST
const source = program.getSourceFile(filePath);

// трансформируем AST
const result = ts.transform(source, [myCustomTransformer]);

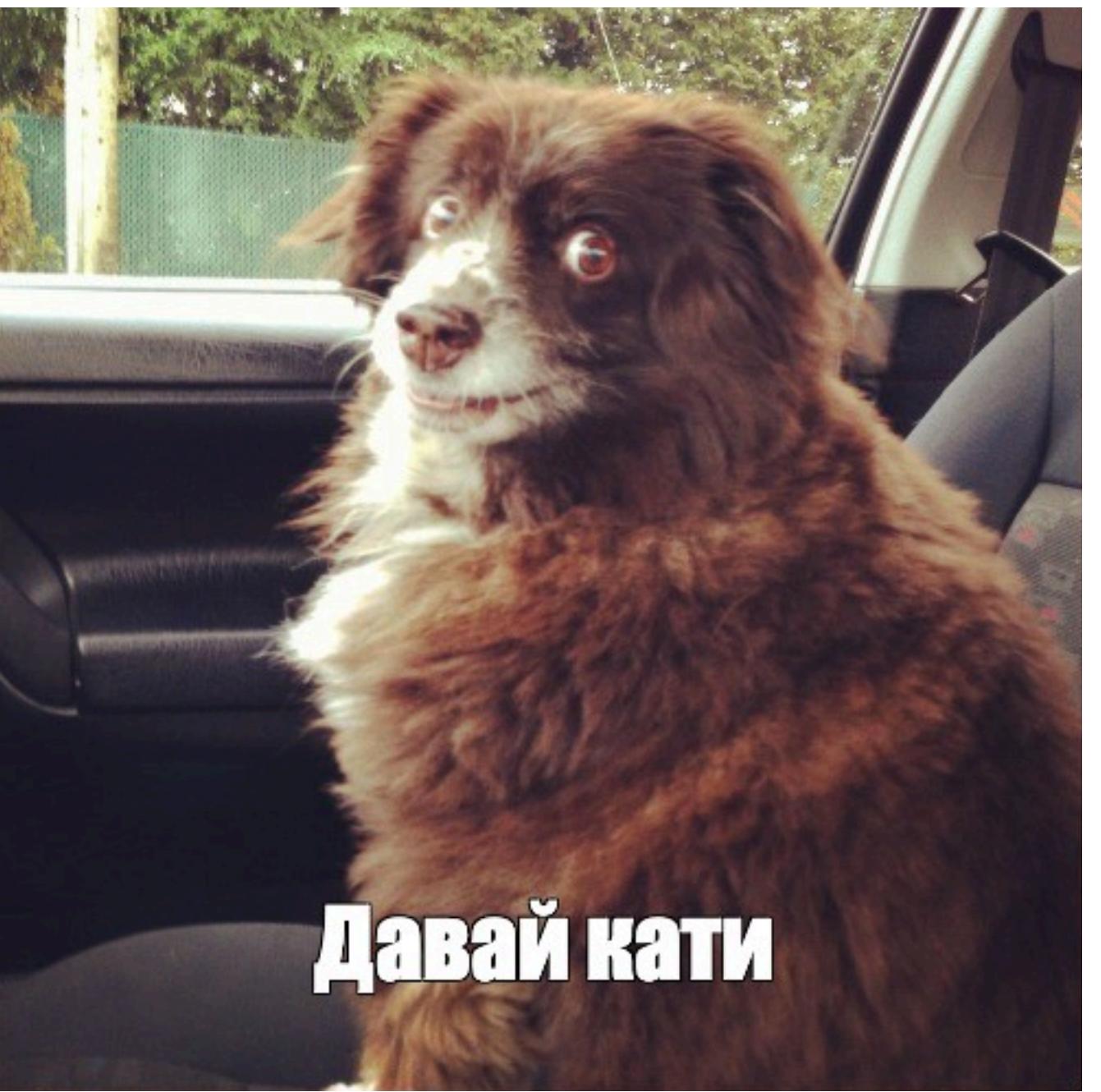
// "принтер" сериализует AST-дерево в ts-код
const printer = ts.createPrinter();
printer.printFile(result.transformed[0]);
```



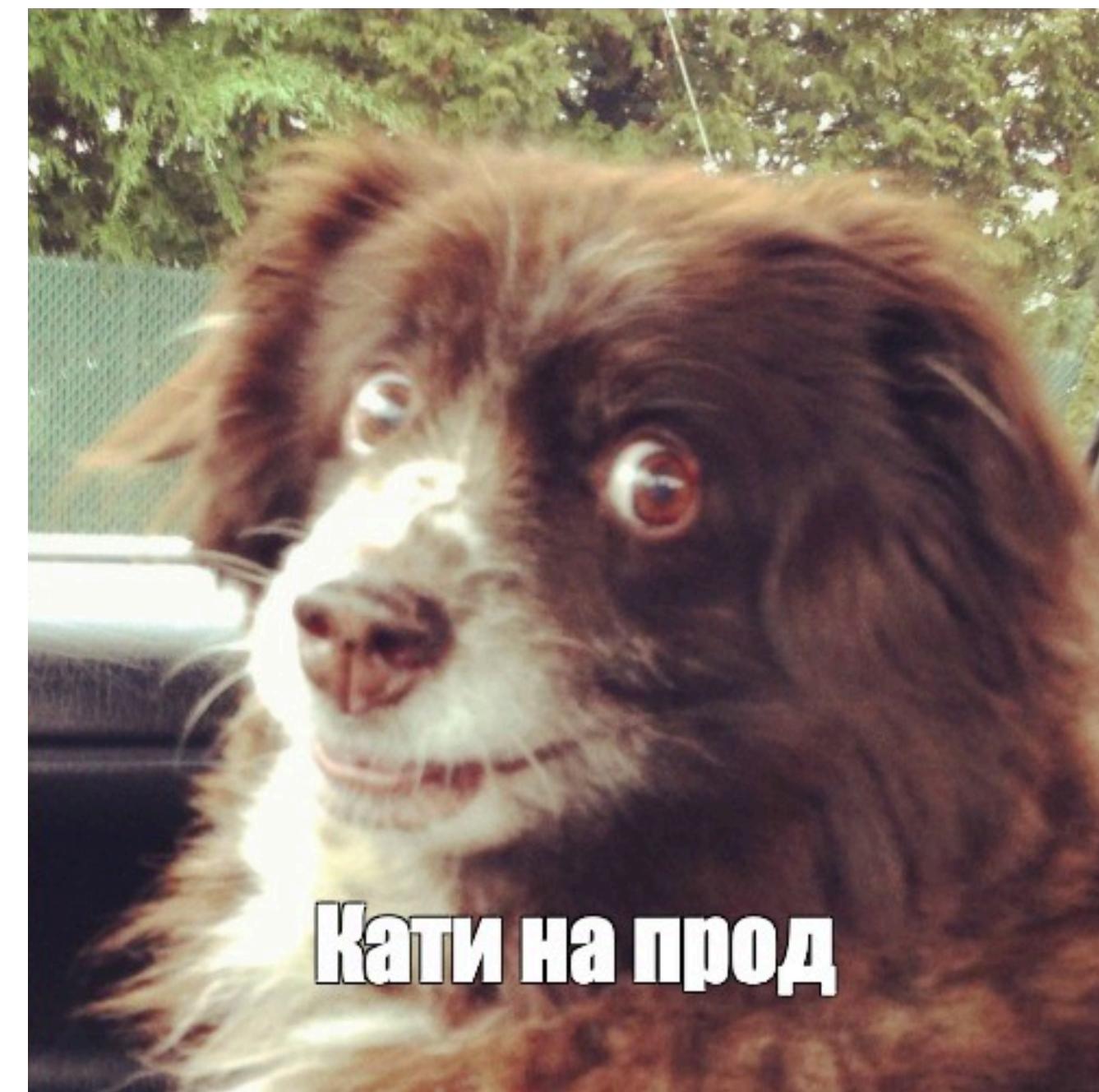
ts-transformer-loader

github.com/mad-gooze/ts-transformer-loader
npmjs.com/package/ts-transformer-loader

```
npm install -D ts-transformer-loader
```



Давай кати



Кати на прод

Вместо итогов — каков профит?

	До	После	Разница	
raw	640 КБ	609 КБ	31 КБ	-4.8%
gzip	165 КБ	162 КБ	3 КБ	-1.8%
brotli	156 КБ	153 КБ	3 КБ	-1.9%



Вопросы?

Алексей Гусев

Старший разработчик интерфейсов



alex-gusev@yandex-team.ru



[@alexey_gusev](https://t.me/alexey_gusev)



clck.ru/N5BKQ