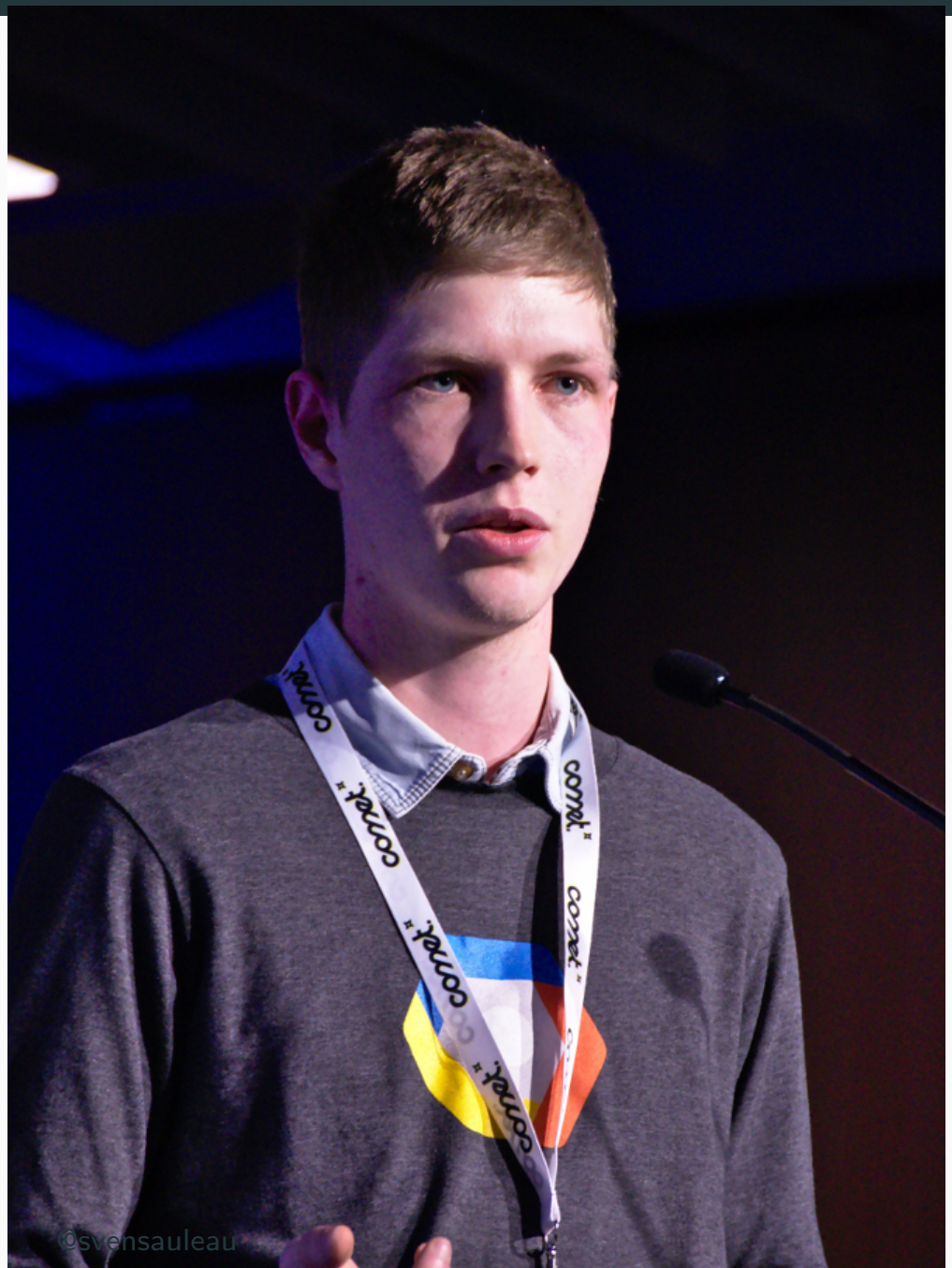
How works Babel?

Sven Sauleau

2018

Sven Sauleau



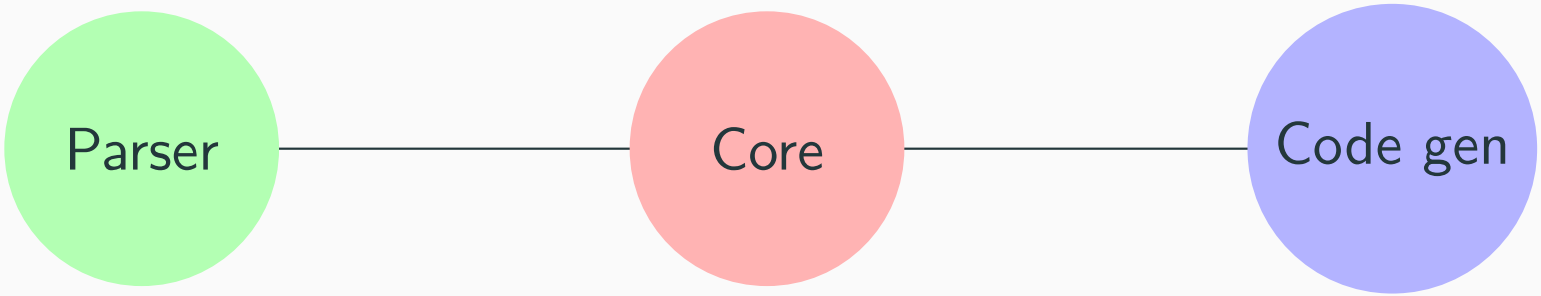
@svensauleau

Babel toolchain

$$Source = \left\{ \begin{array}{l} \text{JSX} \\ \text{ES2015} \\ \\ \vdots \end{array} \right\}$$

$$Babel = Source \rightarrow Source'$$

Babel toolchain



Babylon

@babel/core
@babel/traverse

@babel/generator

$$Parser = Source_1 \rightarrow AST_1$$

$$Core = AST_1 \rightarrow AST_2$$

$$Codegen = AST_2 \rightarrow Source_2$$

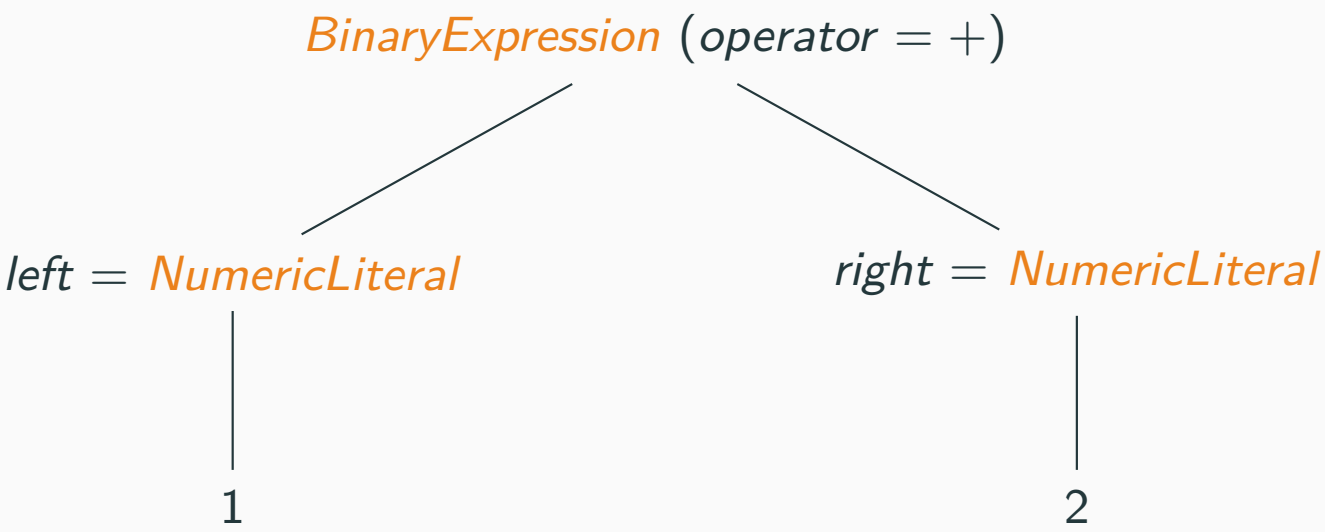
AST

Abstract Syntax Tree.

It's a representation for computers of your program.

AST

$$1 + 2$$


BinaryExpression (operator = +)
left = NumericLiteral
right = NumericLiteral
1
2

Types

@babel/types[1] aka t.

AST builder methods.

$s = t.stringLiteral("test")$

$b = t.booleanLiteral(false)$

$w = t.identifier("window")$

[1] https://github.com/babel/babel/tree/master/packages/babel-types

Template

@babel/template[2]

AST from a string template.

```
const buildLog = template(`
  console.log(TEXT)
`);

const ast = buildLog({
  TEST: t.stringLiteral("Hi")
});
```

[2] https://github.com/babel/babel/tree/master/packages/babel-template

Traversal

@babel/traverse[3]

Traversal/manipulation of the AST.

```
traverse(ast, {
  // vistors
});
```

[3] https://github.com/babel/babel/tree/master/packages/babel-traverse

Visitor pattern

"[...] the visitor design pattern is a way of separating an algorithm from an object structure on which it operates."

— https://en.wikipedia.org/wiki/Visitor_pattern

Allows to visit AST nodes.

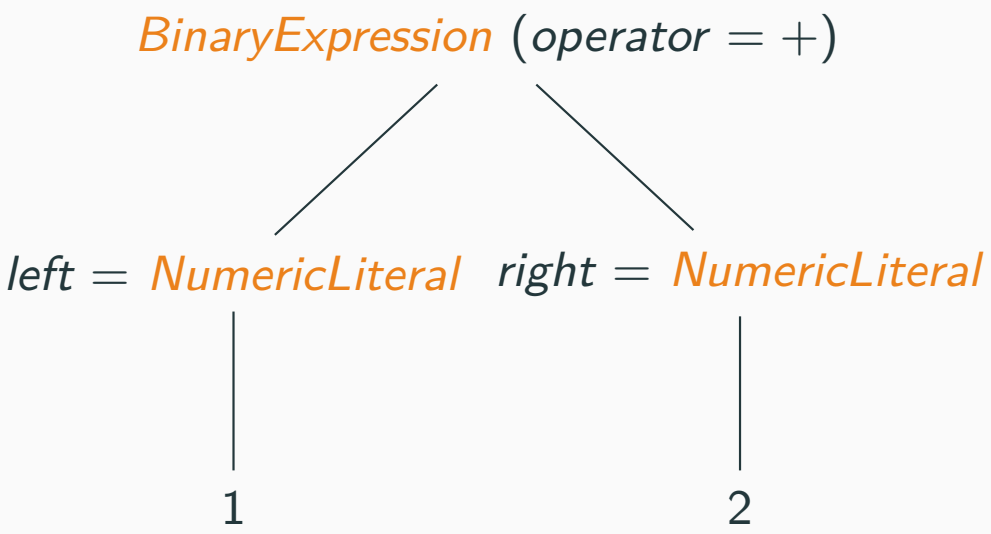
Example

BinaryExpression Visitor

Visitors

```
1 BinaryExpression({node}) {
2     node.operator = '-'
3 }
```

AST



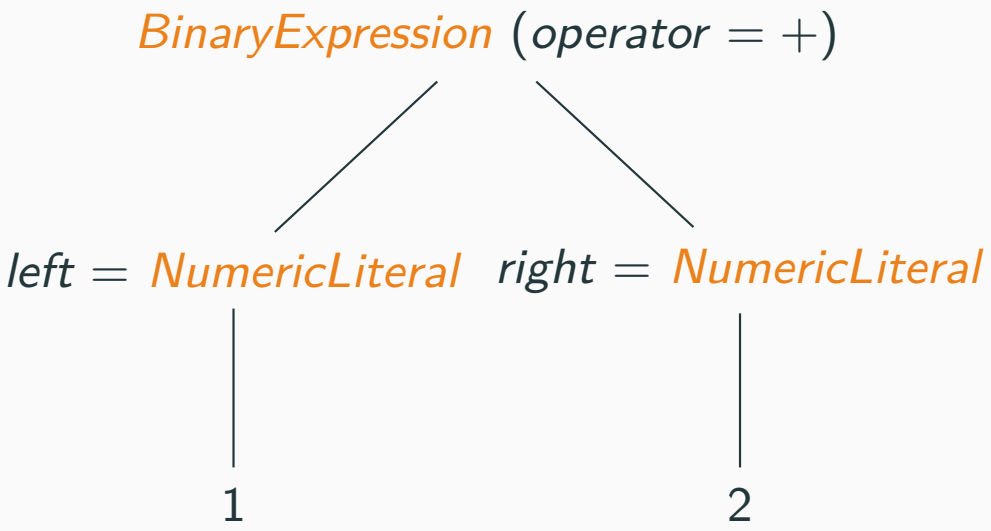
$$1 + 2 \rightarrow 1 - 2$$

NumericLiteral Visitor

Visitors

```
1 NumericLiteral({node}) {
2     if (node.value === 1) {
3         node.value = 2;
4     }
5 },
```

AST



$$1 + 2 \rightarrow 2 + 2$$

Path manipulation: remove

```
1 BooleanLiteral(path) {
2
3     path.remove();
4
5 }
```

In

```
var bar;

true;
```

Out

```
var bar;
```

Path manipulation: replaceWith

```
BooleanLiteral(path) {

    path.replaceWith(
        /* replacement */ t.identifier("bar")
    );

}
```

In

```
true;
```

Out

```
bar;
```

Path manipulation: replaceWithSourceString

```
BooleanLiteral(path) {

    path.replaceWithString(
        "(function () { var a = 1 })()"
    );

}
```

In

```
true;
```

Out

```
(function () {
    var a = 1;
})();
```

Path/scope manipulation: rename

```
FunctionDeclaration(path) {

    path.scope.rename(
        /* oldName */ "a",
       /* newName? */ "renamed_a"
    );
}
```

In

```
function t() {
    var a;
    a;
    a;
}

a;
```

Out

```
function t() {
    var renamed_a;
    renamed_a;
    renamed_a;
}

a;
```

Path manipulation: insertBefore

4

```
BooleanLiteral(path) {

    const node = t.stringLiteral("foo");
    path.insertBefore(node);

},
```

In

```
true;
```

Out

```
"foo";
true;
```

---

[4] same with after

Path/scope manipulation: hoist

```
BooleanLiteral(path) {
    path.hoist();
},
```

In

```

function t() {
    var a = true;
}
```

Out

```
var _ref = true;

function t() {
    var a = _ref;
}
```

Path manipulation: matchesPattern

```
MemberExpression(path) {

    if (path.matchesPattern("foo.bar")) {
        path.remove();
    }

}
```

In

```
foo;
foo.bar;
foo['bar'];
```

Out

```
foo;
```

Path manipulation: evaluate

```
1 BinaryExpression(path) {
2
3     const {value} = path.evaluate();
4     path.replaceWith(t.numericLiteral(value));
5
6 }
```

$$1 + 1 \rightarrow 2$$

traversal: stop

Maximum call stack size exceeded

```
1 BooleanLiteral(path) {
2     path.replaceWith(t.BooleanLiteral(true));
3 }
```

OK

```
1 BooleanLiteral(path) {
2     path.replaceWith(t.BooleanLiteral(true));
3     path.stop();
4 }
```

Babel plugin

```
module.exports = function({ types: t }) {
    return {
        visitor: {
            // visitors
        },
    };
};
```

Babel preset

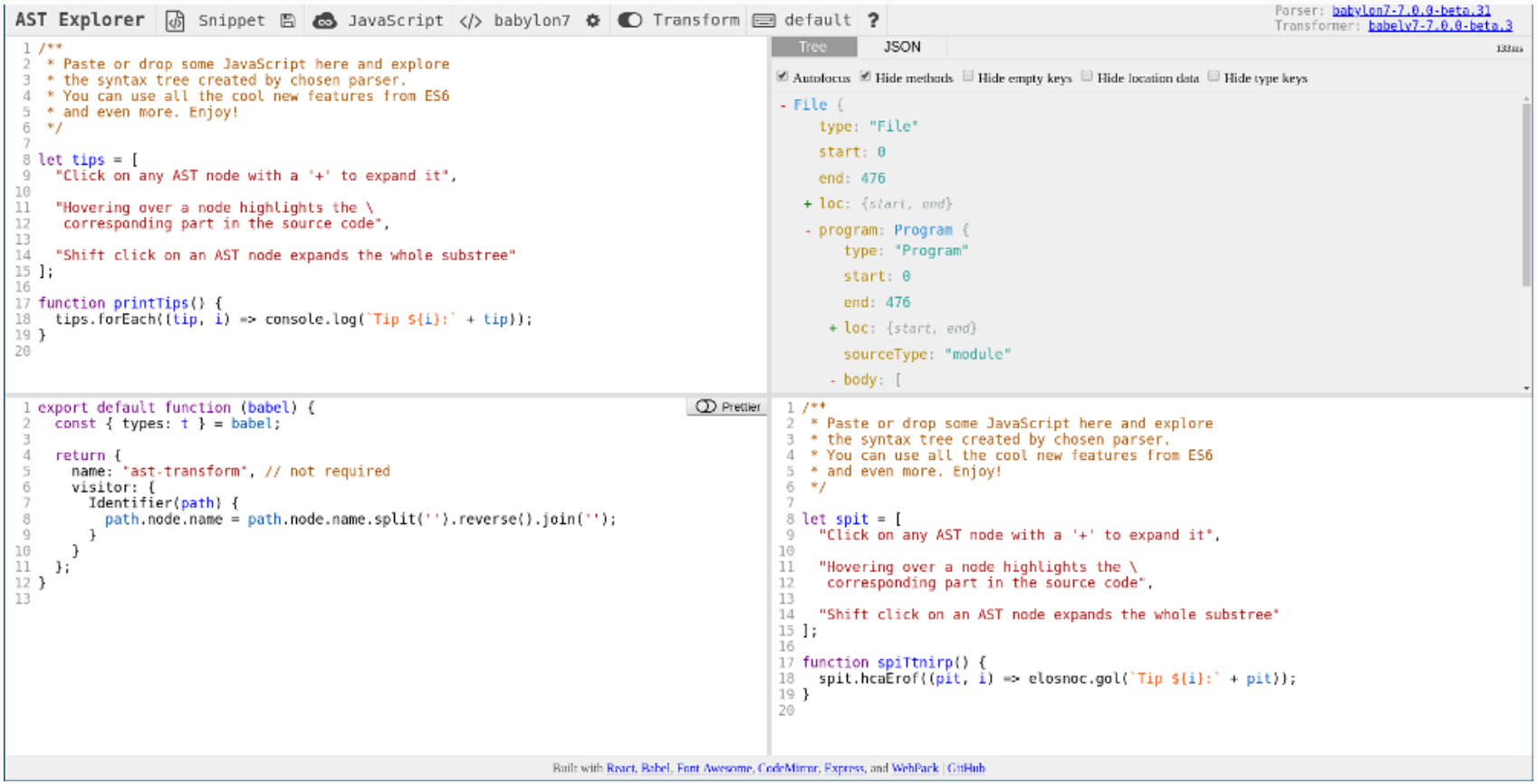
```
function myPlugin({ types: t }) {
    return {
        visitor: {
            // visitors
        },
    };
};

module.exports = function () {
    return {
        plugins: [myPlugin]
    }
}
```

More examples

- babel-plugin-remove-jquery

ASTExplorer



astexplorer.net