

Template Strings の実装

小ネタ

Template Strings

```
const omae = "ore";  
console.log(`Hello, ${omae}`);  
// Hello, ore
```

- JavaScript のことば
- 一般的には String interpolation
 - Ruby, C#, etc...
- f-string (Python)

ほしいもの

```
type Expr = ...  
type Part = string | Expr  
type TemplateString = Part[]
```

- ``String1 ${e} String2`` を
 `["String1 ", e, " String2"]` のようにパースできればよさそう
- あとは好きにコンパイルするなり評価するなり

つまりパースがむずいと

- 一番最初に思いつくのが普通に文字列としてパースしてあとで `${ }` で囲まれた部分を再度パースする
 - 渋め
 - 二重に解析しているので無駄そう
- じゃあどのように字句解析すればいい？

字句

- ``String1``
- ``String1 ${e} String2``
 - ``String1 ${`
 - `e`
 - `} String2``
- ``String1 ${e1} String2 ${e2} String3``
 - ``String1 ${`
 - `e1`
 - `} String2 ${`
 - `e2`
 - `} String3``

パースは楽

```
TemplateString =  
  STRING_LITERAL  
  | STRING_PART_HEAD Expr (STRING_PART_MIDDLE Expr)* STRING_PART_TAIL
```

- `String1 \${e1} String2 \${e2} String3`
 - STRING_PART_HEAD = `String1 \${`
 - STRING_PART_MIDDLE = `} String2 \${`
 - STRING_PART_TAIL = `} String3`

なんかさっき PEG.js で書いたら二秒でできて何？

```
String = "\"\" parts:Part* "\"" { return parts; }  
Part = StrPart / ExpPart  
StrPart = [^"$]+ { return text(); }  
ExpPart = "${" e:Expression "}" { return e; }
```

多分字句解析を挟んでやるタイプだとパッと見どうしたらいいかわかんなくてこまるね～みたいな話だったんだと思います

まとめ

- なんか書いたときはすげーってなった覚えがあったんだけど
スライド書いててわかんなくなった
- すまん