

# Parser Combinator

入门级简介

# 定义

- 给你一段文本，找出其中的上下文无关文法
- 将文法通过 combinatory 转换为 parser

# GraphQL

```
{  
  "bob",  
  "alice",  
}
```

# GraphQL

```
{  
  "bob",  
  "alice",  
}
```

# GraphQL

```
{  
  "bob",  
  "alice",  
}
```

# JSON

```
{  
  "bob": "bar",  
  "alice": "nike",  
}
```

# JSON

```
{  
  "bob": "bar",  
  "alice": "nike",  
}
```

# JSON

```
{  
  "bob": "bar",  
  "alice": "nike",  
}
```



# JSON

```
{  
  "bob": "bar",  
  "alice": "nike",  
}
```

# CSS

```
body {  
    background: grey;  
    color: black;  
}
```

# Parsimmon

- 一个 Parser Combinatory 的 Node.js 实现
- 更多: <https://github.com/jneen/parsimmon>

# 初始化

```
var regex = Parsimmon.regex;  
var string = Parsimmon.string;  
var optWhitespace = Parsimmon.optWhitespace;  
var seq = Parsimmon.seq;  
  
function lexeme(p) {  
  return p.skip(optWhitespace);  
}
```

# GraphQL

```
var start = lexeme(string('{'));
var end   = lexeme(string('}'));
var colon = lexeme(string(','));
var value = seq(
  lexeme(string('"')),
  lexeme(regex(/[a-z0-9]+/)),
  lexeme(string('"'))
).map((values) => values[1]);
var line = seq(
  value,
  colon
).map((values) => values[0]);
var graphql = seq(
  start,
  line.many(),
  end
).map((values) => values[1]);
```

# JSON

```
var start = lexeme(string('{'));
var end   = lexeme(string('}'));
var colon = lexeme(string(','));
var value = seq(
    lexeme(string('"')),
    lexeme(regex(/[a-z0-9]+/)),
    lexeme(string('"'))
).map((values) => values[1]);
var line = seq(
    value,
    lexeme(string(':')),
    value,
    colon
).map((values) => {
    return {
        name: values[0],
        value: values[2]
    };
});
var json = seq(
    start,
    line.many(),
    end
).map((values) => values[1]);
```

# CSS

自己写吧

谢谢观看