Atom

Grammar

- Tree vs Priority 语法树和运算符的优先级 视语法结构来判断优先级

Expressions

- Member （同级）
    - a.b
    - a[b]
    - foo'string'
    - super.b
    - super['b']
    - new.target
    - new Foo()
- New （下一级）
    - new Foo
- Call（下一级）
    - foo()
    - super()
    - foo()['b']
    - foo().b
    - foo()'abc'
    -

例： `new a()['b'] //先new出一个a兑现再访问b属性`

- Left Handside vs Right Handside 例：

```
a.b = c;

a + b = c;
```

- Update (自增自减, 属于Right Handside)
  - a++
  - a--
  - --a
  - ++a
- Unary (单目运算符)
  - delete a.b
  - void foo()
  - typeof a
  - +a
  - -a
  - ~a
  - !a
  - await a
  -
- Exponental
  - ** (右结合，后面的先运算)

例：

```
3 ** 2 ** 3
3 ** (2 ** 3)
```

- Multiplicative
  - */%
- Additive
  - +-
- Shift （位运算）
  - << >> >>>
- Relationship
  - < > <= => instanceof
  - in
- Equality
  - ==
  - !=
  - ===
  - !==
- Bitwise
  - & ^ |
- Logical
  - &&
  - ||
- Conditional
  - ? :

Reference 标准中的类型 引用类型

- Object
- Key
- delete
- assign

Runtime