

极客大学前端进阶训练营

程劭非 (winter)

前手机淘宝前端负责人

重学前端

JavaScript

JavaScript

- Atom
- Expression
- Statement
- Structure
- Program/Module

Atom

Grammar

- 简单语句
- 组合语句
- 声明

Runtime

- Completion Record
- Lexical Environment

Completion Record

- `[[type]]`: normal, break, continue, return, or throw
- `[[value]]`: Types
- `[[target]]`: label

简单语句

- ExpressionStatement
- EmptyStatement
- DebuggerStatement
- ThrowStatement
- ContinueStatement
- BreakStatement
- ReturnStatement

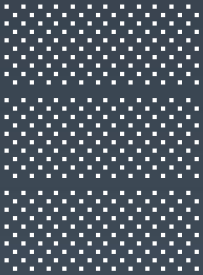
复合语句

- BlockStatement
- IfStatement
- SwitchStatement
- IterationStatement
- WithStatement
- LabelledStatement
- TryStatement

block

- BlockStatement















{



}

- [[type]]: normal
- [[value]]: --
- [[target]]: --


Iteration

- while() 
 - do  while();
 - for( ;  ; ) 
 - for( in ) 
 - for( of ) 
 - ~~for await(of)~~
- var
 - const / let
 - in

标签、循环、break、continue

- LabelledStatement
 - IterationStatement
 - ContinueStatement
 - BreakStatement
 - SwitchStatement
- `[[type]]`: break continue
 - `[[value]]`: --
 - **`[[target]]`: label**

try

```
try {  
      
} catch(  ) {  
      
} finally {  
      
}
```

- [[type]]: return
- [[value]]: --
- [[target]]: label

声明

- FunctionDeclaration
- GeneratorDeclaration
- AsyncFunctionDeclaration
- AsyncGeneratorDeclaration
- VariableStatement
- ClassDeclaration
- LexicalDeclaration

声明

- function
- function *
- async function
- async function
- var
- class
- const
- let

预处理 (pre-process)

```
var a = 2;  
void function () {  
    a = 1;  
    return;  
    var a;  
}();  
console.log(a);
```

```
var a = 2;  
void function () {  
    a = 1;  
    return;  
    const a;  
}();  
console.log(a);
```

作用域

```
var a = 2;  
void function () {  
  a = 1;  
  {  
    var a;  
  }  
}();  
console.log(a);
```

```
var a = 2;  
void function () {  
  a = 1;  
  {  
    const a;  
  }  
}();  
console.log(a);
```

THANKS! |  极客大学