const

```
> const PI = 3.1415926
> PI = 3.0
> console.log(PI)
// VM9886:1 Uncaught TypeError:
   Assignment to constant variable
> 3.1415926

// 注意数组和对象
> const constArr = []
> constArr.push(PI)
> console.log(constArr)
> [3.1415926]

> const constObj = {}
> constObj.pi = 3.14
> console.log(constObj.pi)
> Object {pi: 3.14}
```

Arrow Function

```
> setTimeout(() => {
    console.log('arrow function')
  }, 100)
// 等价于一个匿名函数

> setTimeout(function() {
    console.log('arrow function')
  }.bind(this), 100)
```

let

```
> var ss = 5
> var ss = (ss * 2)/10
> ss
> 1

> let val = 'let value'
> let val = 'another value'
> VM1677:1 Uncaught SyntaxError:
Identifier 'val' has already been
declared


// 注意暂时性死区: Temporal Dead Zones
> console.log(tdz) // undefined
> var tdz = 'value'
> console.log(tdz) // 'value'

// 换成let或者const声明变量
// 不会产生变量提升 hoisting
> console.log(tdz)
// throw ReferenceError
```

Object 扩展

```
// 属性名计算
> let key = new Date().getTime()
> let o = { [key]: 'value'}
> o
> {1480505758842: "value"}

// 对象字面量
> let parents = { mom, dad }
// 等价于

> parents = {
    mom: mom,
    dad: dad
  }

// 更简洁的函数写法

> let main = {
    init(x, y) { … },
    ajax(m, n) { … }
  }
```

模板字符串

```
> const day = 'Monday'
> const month = 'Aug'
> console('today is ${day}, of month
  ${month}.')
> today is Monday, of month Aug.

// 多行书写
> let string = (`this
  is
  string`)
```

二/八/十六进制

```
> 0b1101 // 13
> 0o1024 // 532
> 0xABCD // 43981
```

新数据类型、数据结构

```
Symbol

Set WeakSet
Map WeakMap
```

Promise

```
new Promise((resolve, reject) => {
  fetch('url', (err, res) => {
    if(err) reject(err)
    resolve(res.body)
  })
})
.then(() => { … })
.then(() => { … })
.catch((err) => throw err)


Promise.all([
  promise1, promise2, promise3
]).then(() => {
  // 所有任务结束
})

Promise.race([
  p1, p2, p3
]).then(() => {
  // p1, p2, p3 某一个率先完成
})
```

新的函数作用域

```
> {
  let v = 'there is one value'
  console.log(v)
  }

// 等价于 IIFE
> (function(){
  let v = 'there is one value'
  console.log(v)
  }())
```

函数默认参数

```
> function fn (param=“default”){
  console.log(param)
  }

> fn()
// default
> fn('value')
// value
```

Class extends Setter\Getter

```
// 矩形
Class Rectangle extends Shape {
  constructor(x, y, w, h) {
    super(x, y)
    this.width = w
    this.height = h
  }
  // getter setter
  set width(w) { this._width = w }
  get width() { return this._width }
}
// 圆形
Class Circle extends Shape {
  constructor(x, y, radius) {
    super(x, y)
    this.radius = radius
  }
  moveTen(x){
    super.move(x + '10')
  }
  static fn() { … }
}
Circle.fn()
```

Destructuring/解构赋值

```
// Array Destructuring
> let [a, b, c] = [1, 2, 'cccc']
> console.log(a, c)
// 1 'cccc'

// Object Destructuring

> let boy = { name:'Peter',
  like: 'eating' }
> let { name, like } = boy
> console.log(name, like)
// Peter eating
```

扩展运算符

```
> function prin(...args){
  args.forEach(console.log)
  // arg[0] arg[1] arg[2] ...
}
// 数组 push
> let arr = [1, 2, 3]
> [...arr, 4, 5, 6] // [1,2,3,4,5,6]
// 取值
> let [a, b, c] = ['a1', 'b2', 'c3',
  'c4', 'c5']
> c // ['c3', 'c4', 'c5']
```

生成器 Generator

```
// 1. ajax
function* main(){
  let result = yield request.get('/a.json')
  let res = JSON.parse(result)
}

function request(){
  ajax(url, (res) => {
    it.next(res)
  })
}
let it = main()
it.next()

// 2. a async task of node
const fetch = require('node-fetch')

function* gen(){
  const url = 'https://api.github.com/users/github'
  let result = yield fetch(url)
  console.log(result)
}
let g = gen()
let result = g.next()

// fetch 返回的是 Promise 对象 所以用 then 调用 next
result.value.then((data) => {
  return data.json()
}).then(data => {
  g.next(data)
})
```

Async & Await

```
const fs = require('fs')

const readFile = (...args) => {
return new Promise((resolve, reject)
  => {
    fs.readFile(...args, (err, data)
     => {
       if(err) reject(err)
       resolve(data)
     })
  })
}

async function(){
  const f = 't.text'
  const data = await readFile(f)

  console.log(data) // <Buffer ...>
}
```

export import

```
// export
export function sum(a, b){
  return a + b
}

export const data = 'one data'

let params = { sum, data }
export { params as default }


// import

import 'Vue' from 'vue'

import data from './export'

import * as params from './export'
// 等价于
import params from './export'
// { sum, data }
```