

## ES6

---

- 编译、转换
  - 在线转换
  - 提前编译
  - babel==browser.js
- \*\*\*\*\*
- ES6:
  - 变量
    - var
      - 可以重复声明 会覆盖
      - 无法限制修改 无常量
      - 没有块级作用域 {}
        - let const 都不能重复声明，都有块级作用域{}
        - let 变量 不限制修改
        - const 常量，限制修改 assignment to constant variable
        - 块级作用域
  - 函数
    - 箭头函数 简写
      - 如果只有一个参数，圆括号可以省略
      - 如果只有一条return，花括号可以省略
    - 函数参数
      - 参数扩展/数组展开 ...args剩余参数 rest parameter must be last formal parameter
        - 收集剩余的参数 function show(a , b , ...args ){ } //args 接收剩余参数，数组，必须是最后一个
        - 展开数组 let arr1=[1, 2, 3]; let arr2=[4, 5, 6]; let arr=[...arr1,...arr2]; console.log( arr); //[1, 2, 3, 4, 5, 6]
      - 默认参数
        - function show( a, b=2, c); //设置b的默认参数2
  - 解构赋值 let [a, b, c]=[1, 2, 3]; let {a, b, c}={ a:1, b:2, c:3} 可以自己控制粒度
    - let [ {a,b}, [ n1,n2,n3 ], num, str ]=[ { a:12 , b:5 } , [ 12 , 5 , 8 ] , 8 , 'cxzcy' ]
    - 左右两边结构必须一样

- 右边必须是一个东西
- 声明和赋值不能分开（必须在一句话里完成）      Missing initializer in destructuring declaration
- 数组
  - map 映射 一个对一个
  - reduce 汇总 平均值, 总数, .....
  - filter 过滤器
  - forEach 循环, 迭代
- 字符串
  - startsWith endsWith
  - 字符串模板 `let a=12; alert(`a是 ${a}`)`
    - 可以直接把变量塞到`\${xx}`中
    - 大段HTML字符串
- 面向对象
- JSON
- promise——承诺
  - 封装异步的操作 `Promise.all([])`
  - 异步 操作之间没关联, 同时进行多个操作      代码复杂
  - 同步 操作之间相关, 同时只能做一件事      代码简单
- 举例:
  - 有了Promise之后的异步使用方法
    - `Promise.all([ $.ajax(), $.ajax() ]).then(`
      - `results=>{ },`
      - `err=>{ //错了 }`
    - `)`
    - `Promise.race` 或, 先到者优先
- yield generator
  - generator——生成器
    - 普通函数——一路到底
    - generator函数——中间能停 \*
      - `function *show(){`
        - `alert('a');`
        - `yield;`
        - `alert('b');`
      - `}`

- `let genObj=show();`
    - `genObj.next();//a`
    - `genObj.next();//b`
  - `yield`可以传参且可以有返回值
- Promise——一次读一堆
- generator——逻辑性
- 模块化