介绍

客户端编程语言

名称

JavaScript != Java

今天的目标

- 熟悉语法
- 学习JavaScript基础知识

今天我们将在浏览器里操作

JavaScript版本

我们将用ES6来编写代码

- ECMAScript Edition 6
- 2015年发布的（ECMAScript 2015 Language）
- ~90%浏览器支持

在浏览器上如何编写JavaScript

文件中

JavaScript

```
// in browser dev tools
console.log("Hello Le Wagon");
➜
Hello Le Wagon
```

数据类型

JavaScript

```
"Hello Le Wagon"                字符串 (String)

42                              数字  (Number)
3.14                            数字 (Number)

true                            布尔型 (Boolean)
```

检查数据类型（Data Type）

JavaScript

```
typeof("Boris");
 = 'string'

typeof(42);
 = 'number'
```

转换数据类型

JavaScript

```
Number.parseInt('42', 10);
 = 42

(42).toString();
 = '42'
```

数据结构（Data Structure）

JavaScript

```
[ 'Hello', 'Le', 'Wagon', 42 ]      数组 (Array)

{ name: 'bob', age: 42 }            对象 (Object)
{ 'name': 'bob', 'age': 42 }        对象 (Object)
```

Null和Undefined

```
let age;  //undefined
let name = null;
```

变量（Variables）

JavaScript

之前JS使用的`var`.

ES6用`let`和`const`来替代`var`

`let`

定义的变量可以任意更改

```
let counter = 1;
console.log(counter);

counter = counter + 1;
console.log(counter);
```

const

定义的变量都不可变

```
const firstName = "John";
console.log(firstName);

firstName = "Paul";  //TypeError Assignment to constant va
```

命名规则

JavaScript

```
const firstName = "Ringo";
// lowerCamelCase
```

字符串

让我们更深入地研究这种类型

参考String on MDN web docs

Length属性

```
const firstName = "Paul";
firstName.length;
// => 4
```

字符提取

```
const firstName = John;
firstName[0];
// => "J"

// 从终点index 1开始往后的字符串不被截取
firstName.substring(1);
```

字母大小写转换

```
const firstName = "Paul";
firstName.toUpperCase();
// => "PAUL"

firstName.toLowerCase();
// => "paul"
```

Split方法

```
const monthString = "Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,(

const months = monthString.split(",");
// => [ 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul',
months.length;
// => 12
```

字符串内插（Interpolation）

JavaScript

```
const firstName = "Ringo";
const lastName = "Starr";

const message = `${firstName} ${lastName} is a drummer`;
// => Ringo Starr is a drummer;
```

样板字面值

数组（Array）

参考数组

CRUD

JavaScript

```
const fruits = [];
fruits.push("Apple");  //增加对应 Create
fruits[0];             //查询 Read
fruits[0] = "Banana";  //修改 Update
fruits.splice(0, 1); //删除(0 index的一个项) Delete
```

`forEach`

JavaScript

```
const beatles = ["paul", "john", "ringo", "george"];
beatles.forEach((beatle) => {
  console.log(beatle.toUpperCase());
});
```

Array.forEach

流程控制（Control Flow）

`if` `else`

JavaScript

```
const age = 14;

if (age = 18) {
  console.log(You can vote);
} else {
  console.log(You can't vote);
}
```

错误值（Falsy Values）

JavaScript

```
false
undefined
null
0
NaN
```

三元运算符（Ternary Operator）

JavaScript

```
const raining = true;
const accessory = (raining ? "umbrella" :"sunglasses");
// => umbrella
```

JavaScript

```
if (digit === 0) {
  console.log('Zero');
} else if (digit === 1) {
  console.log('One');
} else {
  console.log("I don't know this digit, sorry!");
}
```

理解JS中的等值比较规则及 `==` 和 `===` 的区别。

对象（Objects）

JS对象指南

简单对象（Simple Object）

```
const student = {
  firstName: "Boris",
  lastName: "Paillard"
};

console.log(typeof student);
// => object

console.log(student);
```

读取和设置属性

可用点操作符（dot notation）去访

```
console.log(student.firstName);
// => "Boris"
console.log(student['firstName']); // Another way
// => "Boris"
student.firstName = "Romain";
console.log(student.firstName);
// => "Romain"
```

函数（Functions）

阅读函数指南

定义函数（Define）

JavaScript（老方式）

```
function square(x) {
  return x * x;
}
```

Note the explicit `return`

调用函数（Calling）

JavaScript

```
square(10);
// => 100
```

箭头函数（Arrow Function）

```
const square = (x) => {
  return x * x;
};

// 更短的语法，与隐式返回（implicit return）
const square = x => x * x;
// 用函数，与前面提到的一样
square(10);
```

应该用什么?

完全相同的函数可以被表示为只有一行代码的箭头函数。练习中，请使用箭头函数。

字母大小事例

如何应用箭头函数（arrow function）并将其存储为 `capitalize`

```
touch libcapitalize.js
const capitalize = (word) => {
  const firstLetter = word[0].toUpperCase();
  const restOfTheWord = word.substring(1).toLowerCase();
  return `${firstLetter}${restOfTheWord}`;
};
```

调试（Debug）

console.log()

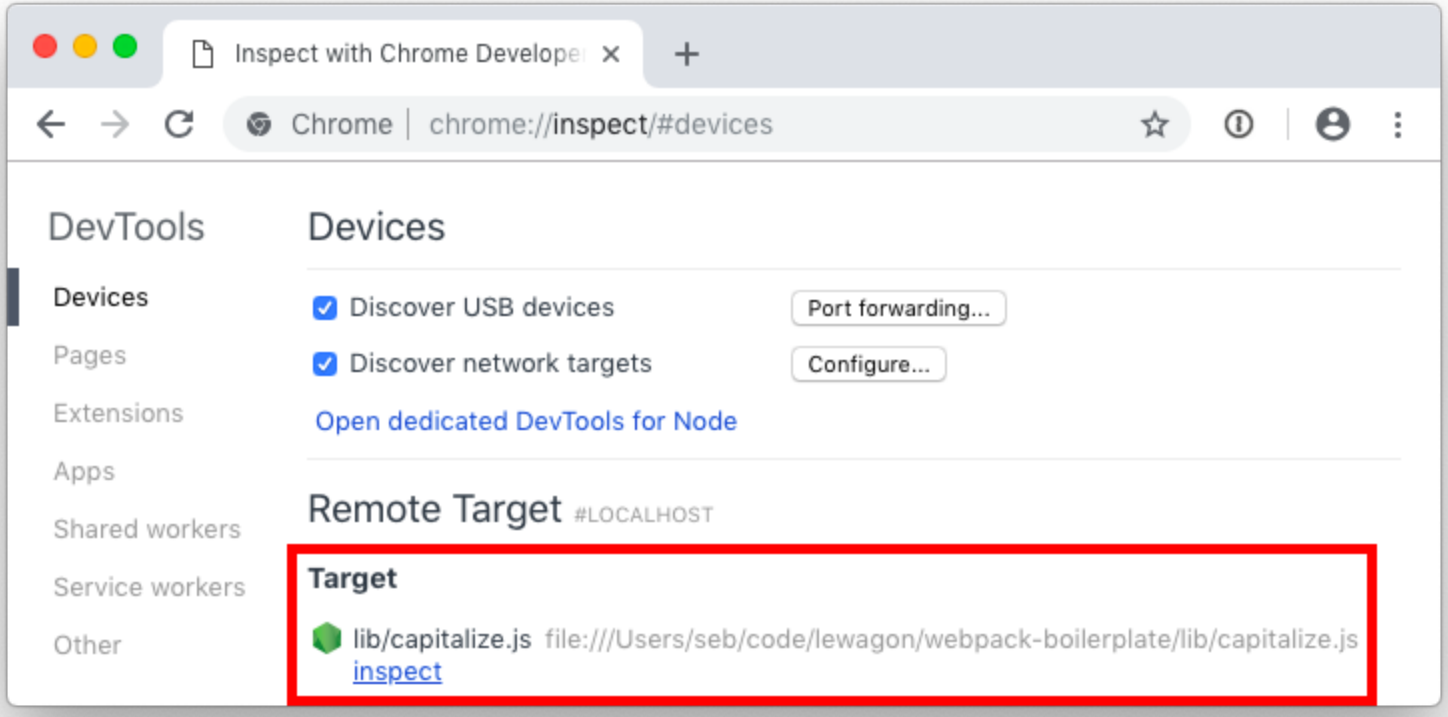
```
const capitalize = (word) => {
  const firstLetter = word[0].toUpperCase();
  console.log(firstLetter);
  const restOfTheWord = word.substring(1).toLowerCase();
  return `${firstLetter}${restOfTheWord}`;
};

capitalize("wagon");
```

Chrome调试工具（1）

在chrome中打开页面，并输入 `chrome://inspect`

- 单击"Inspect"查看需要调试的文件



Chrome调试工具（2）

- 在**Sources / Filesystem**面板上点击 **+ Add folder to workspace**
- 在文件系统中选择文件
- 点击"Allow"!

Chrome调试工具（3）

可以开始**调试**了！在代码中添加一些**breakpoints**

Happy (Back-end) JavaScripting!