目录

[一、资料笔记 2](#_Toc4297)

[组件间通信 2](#_Toc28366)

[二、 课程笔记 3](#_Toc28545)

[1.1 Html模板 3](#_Toc1489)

[1.2 getElementById、getElementsByTagName 3](#_Toc32665)

[1.3事件onmouseover、onmouseout 4](#_Toc2639)

[2.1 标签id 4](#_Toc14192)

[2.2 className 4](#_Toc26608)

[3.1 函数传参 4](#_Toc4023)

[3.2 操作属性的方法 4](#_Toc2498)

[4.1 行间 与 样式表 4](#_Toc12274)

[4.2 style 与 className 4](#_Toc1819)

[4.3 提取行间事件 与 匿名函数 5](#_Toc20577)

[4.4 window.onload 5](#_Toc17533)

[4.5 document.getElementsByTagName 5](#_Toc981)

# 一、资料笔记

## 组件间通信

设信号变量名：single

发送信号

组件a ——————————> 组件b

↓ ↓

在函数中使用 在mounted中监听

$emit(“single”, [xxx, ...] ); $on( “single”, (变量 or 函数) )；

# 课程笔记

### Html模板

<!DOCTYPE HTML>

<html>

<head>

<metacharset="utf-8">

<title>无标题文档</title>

<style>

</style>

<script>

</script>

</head>

<body>

</body>

## 1.2 getElementById、getElementsByTagName

Document.getElementById 、Document.getElementsByTagName

Id和name都是元素中的属性，id相当于人的身份证，是唯一的，name则可以重复，所以Document.getElementById获取到一个元素，而Document.getElementsByTagName可以获取多个元素

在FF等一些浏览器下直接使用ID存在问题，Document.getElementId引入在任何浏览器下均可使用。

Document.getElementsByTagName获取到的元素存在于数组中，eg.

Var aDiv=Document.getElementsByTagName(‘Div’);

aDiv[0].xxx = xxx;

....

## 1.3事件onmouseover、onmouseout

鼠标移入，鼠标移出

## 标签id

任何标签都可以加id

可通过给标签加id，通过js更改其标签属性

## className

标签中的class在js中在用className，因为js中也有关键字class，用于区别

## 3.1 函数传参

定义形参时不需要制定参数类型，eg. Function setColor(color){ }

## 3.2 操作属性的方法

第一种方法：用 ’ . ’，eg. document.getElementById('div1').style.display

第二种方法：用 “ [‘ ’] ” eg. document.getElementById('div1').style[‘display’]

一切用‘. ’的地方都能用” [ ] “

再eg. Document.getElementById(‘txt1’) => document[‘getElementById’](‘txt1’);

## 4.1 行间 与 样式表

行间：html中的行代码 < xxx >

样式表: <style></style>中的的样式

## 4.2 style 与 className

Style加样式和取样式值都在行间中处理，如果style放在样式表中，则取不出来

--元素.style.属性=xxx 是修改行间样式

--修改完行间样式之后再修改className不会有效果

## 4.3 提取行间事件 与 匿名函数

提取行间事件，把行间的事件放在js中处理

匿名函数，不取名字的函数，eg.

<sricpt>

oBtn.onclick = function()

{

.....

}

</script>

## 4.4 window.onload

Window.onload也是一个事件，在整个页面加载完成的时候才会执行

可以把js代码都放在该函数中处理eg.

window.onload=function() {

...

}

## 4.5 document.getElementsByTagName

见1.3

## 5.1 innerHTML

表示便签中属于HTML的东西, 可以往里面写HTML也可以读出来

## 5.2 alert

# 6 课程基础

## 6.1 Js的组成

ECMAScript 翻译解析器

DOM （Document Object Model） Html Document

Bom （Browser Bbject Model） 浏览器 window

## 6.2 变量类型

常见类型： number string boolean undefined object function

### 6.2.1 显式类型转换（强制转换）：

**parseInt(string)** 转换成整数它从左到右扫描字符串，直到遇到非数字即停下来。

Eg. parseInt(“123px456”) => 123

Nan 与 Nan不相等，即Nan == Nan为false，判断用 isNan() 判断

**ParseFloat(string)** 转换成小数

### 6.2.2 隐式转换

== 先自动转换类型后比较 !=

=== 比较类型和数值 !==

- （减号） 先转换后相减，+加号则不会，加号作用：1.字符串连接 2.相加

## 6.3闭包

通俗理解就是：子函数可以使用父函数中的局部变量

# 6.4 程序流程控制

判断： if switch...case:break; ?:

循环：while for

Eg. for(var i in xxx)

跳出：break continue

## 6.5 函数不定参

arguments 它是系统提供的一个数组，函数中可以传任意参数，形参中没有名字的参数可以通过arguments来获取，按顺序排列

## 6.6 取非行间样式

### 6.6.1兼容性

Var oDiv = document.getElementById(‘div1’);

IE: oDiv.currentStyle.width

Chrome FF: getComputedStyle(oDiv, xxx).width ps.第二个参数xxx是个垃圾不用管

为兼容两种不同浏览器，可作以下处理：

If(oDiv.currentStyle){

//IE

oDiv.currentStyle.width

}else{

//Chrome FF

getComputedStyle(oDiv, xxx).width

}

只能取单一样式

### 6.6.2 样式分类

复合样式：background、border

单一样式：width height position

## 6.7 数组

### 6.7.1 基本操作

创建：

Var a=[1,2,3];

Var a= new Array(1,2,3);

属性：

Length: 既可以获取也可以设置

Eg. 快速清空数组 a.length=0;

添加：

push a.push(xx)

unshift a.unshift(xx)

删除：

pop a.pop() 尾部删除

shift a.shift() 头部删除

### 6.7.2 万能的splice(起点，长度，元素)

Splice(起点，长度) 中间删除

splice(起点，长度，元素)，先从起点删除n个元素，再添加

### 6.7.3 数组连接 concat() join()

b.concat(a), 把a连接到b后面

b.join(‘-’) ,分隔符

### 6.7.4 数组排序

sort()

a.Sort() 按首字母assic码从小到大排序

如何排列数字？

A.sort(fuction(n1,n2){

Return n1-n2;})