HTML-CSS-JS

HTML

2020-02-17

|  |  |
| --- | --- |
| 属性 样式 |  |
| 链接属性及其使用 |  |
| 表格使用 |  |
| 列表 块 布局 |  |
|  |  |
|  |  |

2020-02-18

|  |  |
| --- | --- |
| 1.HTML 块 | 1.块元素p h;2. 内联元素b a ;3.Div块元素容器 配合CSS样式使用； 4.span 文本容器 |
| 2.HTML布局 | 1.使用div布局；container-heading-content\_menu-content\_body-foot  2.table布局 |

2020-02-19

|  |  |
| --- | --- |
| 1.表单的创建 | 1.HTML表单（复选框，单选按钮，下拉列表，文本域，创建按钮）；2PHP环境搭建；3.表单提交数据与PHP交互 |
| 2.PHP环境搭建 |  |
| 3.表单提交数据与PHP交互 | HTML中action获得PHP中的地址，用get/post方法，使用post安全，get资源定位。 |
| 4.HTML框架，背景颜色 | 1.框架标签（frame），2内联框架（iframe） |
| 5. 实体 | &lt;html &gt; |
| 6.XHTML | XHTML是什么？可扩展的超文本标记语言；好处：代码的完整性和良好性。文档申明。元素语法：嵌套，始终关闭，小写，有根元素 |
| 7.推出的理由及目标语法的改变（HTML5与4 的区别） | 1.web浏览器之间的兼容性很低；2.文档不够明确；3.web应用程序的功能收到了限制。 |
| 8.新增的元素和废除的元素 | 新增结构元素：section,article,aside,header,hgroup,footer,nav,figure  /其他元素:video,audio,canvas,  /input元素email,url,number  不在使用frame框架 |
| 9.全局属性 | contentEditable,designMode,hidden,spellcheck,tabindex(获取焦点) |
| 10.article元素 | 可嵌套使用，可做插件 |
| 11.section元素 |  |
| 12.nav元素 | 页面导航的链接组，传统导航条；侧边栏导航，页内导航翻页操作 |
| 13.aside元素 | 用来表示当前页面或文章的附属信息部分 |
| 14.time元素与微格式 PUBDATE属性 |  |
| 15．Header元素 | Header:具有引导和导航作用的结构元素，放置整个页面或页面内的一个内容区块的标题。  address |
| 16. Footer, hgroup元素 | Footer：包括脚注信息，作者版权等  Hgroup：把标题和子标题进行分组； |
| 17.address元素网页编排 | Address：地址，邮箱，电话号码 |
| 18.form | Form  Formaction属性： |
| 19.fommethod属性 | Fommethod：用该方法来提交不同元素  Fomenctype： |
| 20.formtarget | Formtarget属性：在何处打开所需加载的界面；  Autofocus属性： |
| 21.required和labels |  |
|  |  |
|  |  |
|  |  |
|  |  |

2020-03-05

|  |  |
| --- | --- |
| Control，Placeholder， | Control:  Placeholder：文本框未输入时现实的文字。 |
| List-Datalist | Datalist :可以输入的表单 |
| Autocomplete |  |
| Pattern | 正则规则的验证 |
| Indeterminate |  |
| Image –height &width | 指定照片的宽度与高度 |

2020-03-05

CSS

|  |  |
| --- | --- |
| CSS介绍 | 层叠样式表  提高工作效率 |
| 基础语法 | 1.选择器分组；2.继承（无法改变已有的属性） |
| 高级语法 |  |
| 派生选择器 | 改变当前标签里的标签的属性 |
| Id选择器 | 以#来定义；派生选择器和id选择器一起使用 |
| 类选择器 | 以一个点来显示；class也可以； |
| 属性选择器 | 自己新增属性用[] 然后在其他元素中使用 |

CSS样式背景

|  |  |
| --- | --- |
| CSS背景 |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

QUESTION AND RESOLVE.

|  |  |
| --- | --- |
|  |  |
| 2020-03-12 | HTML:页面的创建以及布局；  CSS外联的使用？  表单的创建：table开始，tr定义有几行，td每一行有几格，th定义表头；  CSS内部样式表的使用：class和id怎么选择？  内联是指：在写标签时就加进去CSS样式。如<p style=”font-size:10”></p>  隔行变色问题？  对象包括属性(Properties)和方法(Methods)，属性就是需要记忆的信息，方法就是对象能够提供的服务。 |
| 2020-03-13 | 1盒子模型:margin（外边框）,border（边框）,padding（内边距）,content（内容）；  2.CSS分组和嵌套选择器  P{}：为所有p元素指定样式；  为所有class指定样式；  Class选择器用. ID选择器用#  3.为不同元素指定尺寸**Dimension属性**；  4.可见性与不可见性；  5.CSS定位  6.CSS伪类：是用来添加一些选择器的特殊效果  hover 必须被置于 a:link 和 a:visited 之后，才是有效的  active 必须被置于 a:hover 之后，才是有效的  7.CSS导航栏 hover等属性，鼠标操作时改变其背景色  8.网站的布局：导航栏，头部，尾部，正文， |
| 2020-03-17 | 1. 网站：头部，导航条，正文，尾部；   其中的导航条布局，a.hover{}鼠标悬浮在上面时改变其颜色， |
| 2020-03-18 | 1. 设置相同的属性可以放在一起设置，大标签使用class，小标签具体到有多个相同时使用id 2. 对div设置 如# main{};对main这个div里面的所有div进行设置#main div{}; 3. 响应式设计：通过给标签添加class来实现（表格部分）；   表单的提交   1. 伪元素::first-line/letter,::after/before,::selection, 2. 伪类选择器-动态伪类选择器：:link/:visited/:hover/:active/:foucus鼠标操作时的提示 3. UI伪类选择器：:enabled/:disabled,:checked |
| 2020-03-19 | 1. 盒子模型：每个盒子有四个边界：*内容边界* *Content edge*、*内边距边界* *Padding Edge*、*边框边界* *Border Edge*、*外边框边界* *Margin Edge*   盒子边框：none: 没有边框即忽略所有边框的高度（默认值） solid：边框为单实线（最常用） dashed： 边框为虚线 dotted：边框为点线 double：边框为双实线  **内边距（padding）**  padding属性用于设置内边距。 是指 边框与内容之间的距离。 |
| 03-23 | 1. 下拉菜单功能的实现。   在div里嵌套div然后实现； |
| 03-24 | 1. 定位（static fixed absolute relative sticky） 2. Overflow:控制内容溢出元素框时在对应的区间内添加滚动条。 3. Float 4. 组合选择符： 后代选择器(以空格分隔)，子元素选择器(以大于号分隔），相邻兄弟选择器（以加号分隔），普通兄弟选择器（以破折号分隔） 5. 伪类：first-child 伪类（匹配任何元素的第一个子元素）， 6. 伪元素：：first-line/first-letter/before/after/link/visited/active/hover/focus/first-child/lang 7. 导航栏： 8. 下拉菜单： 9. 提示工具： 10. 图片廓：响应式 11. 透明与不透明：opacity设置filter |
| 03-25 | 1. 媒体类型 2. 属性选择器 |
| 03-26 | 1. 表单 2. 网页布局：头部，导航栏，内容，尾部   CSS3   1. 边框：圆角，盒阴影，边界图片 2. 圆角 3. 背景：不同的背景图像和图像用逗号隔开，所有的图片中显示在最顶端的为第一张 4. 渐变 5. 文本效果 |
| 03-27 | 1. 字体： 2. 2D转换：也就是让图像旋转倾斜，不是在一条直线就可以 3. 3D转换：XYZ都转换 4. 过渡：长，宽，旋转，时间的等待 5. 动画： 6. 多列： 7. 用户界面： 8. 图片： 9. 按钮： 10. 分页： 11. 框大小： 12. 弹性盒子： 13. 多媒体查询： 14. 多媒体查询实列。 |

JS

|  |  |
| --- | --- |
|  |  |
| 2020-03-18 | 1. JS的三种书写方式   行内式/内嵌式/外部式   1. 输入：prompt 输出：alert 控制台输出console； 2. 变量（存放数据的容器） |
| 03-27 | 1. JS用法 2. 输出 3. 语法 4. 语句 |
| 03-29 | 1. 注释 2. 变量：用于存放值和表达式 3. 数据类型：string,Boolean,null,undefined,symbol,object,array,function 4. 数组/对象（用花括号分割，在括号内部，对象的属性以名称和值对的形式来定义（name：value）。属性由逗号分隔），当申明变量时，可用关键词new来申明其类型，变量均为对象，申明一个变量时，就创建了一个新对象； 5. 对象：在js中一切都可以作为对象，对象有他的属性和方法；JavaScript 对象是键值对的容器（也是属性变量的容器），键值对通常写法为 **name : value** ；访问对象属性；对象方法(定义一个函数作为对象的属性存储)； 6. 函数：函数是事件驱动的或者当它被调用时可执行的可重复使用的代码块。函数语法 function myFunction{可执行的代码块}； 带参数的函数myFunction(argument1,argument2){};带返回值的函数mFunction{ return x};变量（局部，全局）； 7. 作用域：可访问变量的集合； 8. 事件：（页面加载完成/input字段改变时/按钮被点击），常见事件（onchange/onclick/onmouseover/onmouseout/onkeydown/onload）; 9. 字符串:字符串用于存储和处理文本（字符串的索引从 0 开始，这意味着第一个字符索引值为 [0],第二个为 [1], 以此类推，可以使用内置属性 **length** 来计算字符串的长度）；字符串方法； 10. 运算符：+,-,\*,/,%,++,--, 11. 比较：==,!=,<,>,&&,||,!, 12. 条件语句：不同条件用于执行不同的动作，if,I if.else, if .else if. else,swith; 13. Switch：switch(n){case1: …case2:...default}; 14. For:for(var i=0;i<cars.length;i++){…}; 15. While:while(condition) {…} 只要指定条件为true，循环一直执行代码块；do/while ,do{…}while(condition)先执行后判断； 16. Break continue: 17. Typeof,null,undefined:用typeof来检测变量的数据类型。 18. 类型转换：6种不同的数据类型：string/number/Boolean/object/function/symbol，3种对象类型：object,date,array,2个不包含任何值的数据类型：null,undefined. Constructor属性 19. 正则函数：使用单个字符串来描述、匹配一系列符合某个句法规则的字符串搜索模式。正则表达式是由一个字符序列形成的搜索模式。Search(),replace(),正则表达式参数可用在以上方法中 (替代字符串参数)。 正则表达式使得搜索功能更加强大(如实例中不区分大小写)。 20. 错误：throw,try,catch, **try** 语句测试代码块的错误。   **catch** 语句处理错误。  **throw** 语句创建自定义错误。  **finally** 语句在 try 和 catch 语句之后，无论是否有触发异常，该语句都会执行。   1. 调试：console.log()/debugger（用于停止调试相当于设置断点）   变量提升：JavaScript 中，函数及变量的声明都将被提升到函数的最顶部。  JavaScript 中，变量可以在使用后声明，也就是变量可以先使用再声明。   1. 严格模式："use strict" 指令在 JavaScript 1.8.5 (ECMAScript5) 中新增。 2. 使用误区： 3. 表单：表单验证/验证输入的数字； 4. 表单验证： 5. 验证API:约束验证 DOM 方法/validity属性/ 6. 保留关键字：在 JavaScript 中，一些标识符是保留关键字，不能用作变量名或函数名。 7. This: 面向对象语言中 this 表示当前对象的一个引用。但在 JavaScript 中 this 不是固定不变的，它会随着执行环境的改变而改变。在方法中，this 表示该方法所属的对象。如果单独使用，this 表示全局对象。在函数中，this 表示全局对象。在事件中，this 表示接收事件的元素。 8. Let/const: 9. Json: JSON 是用于存储和传输数据的格式。通常用于服务端向网页传递数据 . 英文全称 **J**ava**S**cript **O**bject **N**otation是一种轻量级的数据交换格式 10. Void:不返回值 11. 代码规范： |
| 03-31 | 1. 函数：函数可以通过声明定义，也可以是一个表达式。函数表达式可以存储在变量中 函数是对象，箭头函数 2. 函数参数：显示参数与阴式参数：比较大小，求和， 3. 函数调用：在Javascript中，this指向函数执行时的当前对象。函数作为方法调用，使用构造函数调用函数， 4. 闭包：JavaScript 变量可以是局部变量或全局变量。私有变量可以用到闭包。全局变量：函数可以访问由函数内部定义的变量， 5. DOM:通过DOM，可以访问HTML文档的所有元素。HTML DOM树， |
| 04-01 | 1. DOM:   通过 id 找到 HTML 元素  通过标签名找到 HTML 元素  通过类名找到 HTML 元素   1. DOM改变HTML元素的内容：改变输出流/HTML内容/HTML属性/ 2. DOM改变CSS:改变HTML样式/使用事件（元素被点击，页面加载完成，输入框被修改） 3. 4DOM事件：对事件做出反应，  * 当用户点击鼠标时 * 当网页已加载时 * 当图像已加载时 * 当鼠标移动到元素上时 * 当输入字段被改变时 * 当提交 HTML 表单时 * 当用户触发按键时 事件属性，分配事件，onload和onunload/onchange/onmouseover/onmouseout(鼠标事件方法字面颜色等的改变)  1. DOM EventListener:用户点击按钮时触发监听事件；   addEventListener() 方法用于向指定元素添加事件句柄。  addEventListener() 方法添加的事件句柄不会覆盖已存在的事件句柄。   1. DOM元素节点：创建新的 HTML 元素 (节点) - appendChild() 要创建新的 HTML 元素 (节点)需要先创建一个元素，然后在已存在的元素中添加它 。创建新的 HTML 元素 (节点) - insertBefore()以上的实例我们使用了 appendChild() 方法，它用于添加新元素到尾部。如果我们需要将新元素添加到开始位置，可以使用 **insertBefore()** 方法:移除已经存在的元素，需要知道该元素的父元素。替换元素。 2. DOM集合（Collection）:HTMLCollectoin.length属性返回数量， 3. DOM节点列表：NodeList.length属性，**HTMLCollection 与 NodeList 的区别**[HTMLCollection](https://www.runoob.com/js/js-htmldom-collections.html) 是 HTML 元素的集合。NodeList 是一个文档节点的集合。NodeList 与 HTMLCollection 有很多类似的地方。NodeList 与 HTMLCollection 都与数组对象有点类似，可以使用索引 (0, 1, 2, 3, 4, ...) 来获取元素。NodeList 与 HTMLCollection 都有 length 属性。HTMLCollection 元素可以通过 name，id 或索引来获取。NodeList 只能通过索引来获取。只有 NodeList 对象有包含属性节点和文本节点。 4. JS对象：JavaScript 中的所有事物都是对象：字符串、数值、数组、函数...也可以自定义对象，访问对象的属性，属性是与对象相关的值，访问对象的方法，创建对象，对象构造器，在JavaScript中，this通常指向的是我们正在执行的函数本身，或者是指向该函数所属的对象（运行时），创建对象实例，把属性添加到JS对象，把方法添加到JS对象，for...in 循环中的代码块将针对每个属性执行一次。 |
| 04-02 | 1. Prototype 原型对象:使用对象构造器，在已存在的构造器中是不能添加新的属性的。要添加一个新的属性需要在在构造器函数中添加。所有的 JavaScript 对象都会从一个 prototype（原型对象）中继承属性和方法， 2. Number 对象： 数字均为64位，如果前缀为 0，则 JavaScript 会把数值常量解释为八进制数，如果前缀为 0 和 "x"，则解释为十六进制数。无穷大：Infinity.数字可以是数字或者对象，数字{属性，方法} 3. 字符串对象：String 对象用于处理已有的字符块，一个字符串用于存储一系列字符就像 "John Doe".使用位置（索引）可以访问字符串中任何的字符，字符串的索引从零开始, 所以字符串第一字符为 [0],第二个字符为 [1]，字符串的属性（.length,prototype,constructor） ，方法（indexOf，match，replace，toUpperCase,toLowerCase,split,）对象的属性和方法； 4. Date对象：如需从 JavaScript 访问某个 HTML 元素，您可以使用 document.getElementById(*id*) 方法。请使用 "id" 属性来标识 HTML 元素，并 innerHTML 来获取或插入元素内容， 要使用一个对象先要实例化， Date 对象用于处理日期和时间。 5. Array 对象：数组的定义数组对象的作用是：使用单独的变量名来存储一系列的值。数组可以用一个变量名存储所有的值，并且可以用变量名访问任何一个值，数组中的每个元素都有自己的的ID，以便它可以很容易地被访问到，创建数组，访问数组（通过指定数组名以及索引号码） 6. Boolean对象：用于将非布尔值转换为布尔值（true 或者 false）。Boolean 对象代表两个值:"true" 或者 "false"， 7. Math 对象：(方法round/random/max/min/,Math)（算数）对象的作用是：执行普通的算数任务。 8. RegExp 对象：RegExp：是正则表达式（regular expression）的简写，定义：正则表达式描述了字符的模式对象。修饰符用于执行不区分大小写和全文的搜索。**i** - 修饰符是用来执行不区分大小写的匹配。**g** - 修饰符是用于执行全文的搜索（而不是在找到第一个就停止查找,而是找到所有的匹配） 9. Window 浏览器对象模式： 浏览器对象模型 (BOM) 使 JavaScript 有能力与浏览器"对话"，由于现代浏览器已经（几乎）实现了 JavaScript 交互性方面的相同方法和属性，因此常被认为是 BOM 的方法和属性。**Window 对象**所有浏览器都支持 window 对象。它表示浏览器窗口。所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。全局变量是 window 对象的属性。全局函数是 window 对象的方法。Windows 方法：open/close/moveto/resizeto 10. Window Screen: window.screen 对象包含有关用户屏幕的信息。一些属性：availWidth/availHeight. 11. Window Location:hostname/pathname/port/protocol/,assign申请打开新的地址 12. Window History:包含浏览器的历史，history.back() - 方法加载历史列表中的前一个 URL，与在浏览器点击后退按钮相同，history.forward() - 与在浏览器中点击向前按钮相同 13. Navigator: 包含有关访问者浏览器的信息;对象中的方法； 14. 弹窗：警告框(alert)、确认框(confirm)、提示框(prompt),\n来换行。 15. 计时事件：在一个设定的时间间隔之后来执行代码，setInterval() - 间隔指定的毫秒数不停地执行指定的代码，setTimeout() - 在指定的毫秒数后执行指定代码， 16. Cookie: Cookie 是一些数据, 存储于电脑上的文本文件中, 当 web 服务器向浏览器发送 web 页面时，在连接关闭后，服务端不会记录用户的信息。Cookie 的作用就是用于解决 "如何记录客户端的用户信息": 创建cookie/读取cookie/修改/删除/ 17. JavaScript库： jQuery、Prototype、MooTools, 18. 测试jQuery：主要的 jQuery 函数是 $() 函数（jQuery 函数）。如果您向该函数传递 DOM 对象，它会返回 jQuery 对象，带有向其添加的 jQuery 功能， 19. 测试Prototype: 20. 实例：显示时间/用js改变html内容/外部的js/提示框/（语句/注释/代码块/变量/条件语句/消息框/函数/循环/事件/错误处理） 21. 对象实例：JS内置的(string/date/array/Boolean/math/) 22. Broswer 对象实例：Window/Navigator/Screen/History/Location 23. DOM 实例：Document/Anchor/Area/Base/Button/Form/Frame,IFrame/Image/Event/Option,Select/Table,TableHeader,TableRow,TableData/ 24. 总结：DOM/BOM/AJAX/JQUERY/ASP/ |
|  |  |

jQuery

|  |  |
| --- | --- |
|  |  |
| 04-02 | 1.简介：包含一下功能HTML元素选取/操作，CSS操作，HTML事件函数，JS特效和动画，DOM遍历和修改，AJAX,Utilities.  2.安装：  3.语法：jQuery 语法是通过选取 HTML 元素，并对选取的元素执行某些操作，基础语法： $(*selector*).*action*()  美元符号定义￥，选择符号查询和查找HTML元素，action执行对元素的操作。所有的jQuery函数都位于document ready函数中：$(document).ready(function(){});  4 jQuery选择器：jQuery 选择器允许对 HTML 元素组或单个元素进行操作，jQuery 选择器基于元素的 id、类、类型、属性、属性值等"查找"（或选择）HTML 元素。 它基于已经存在的 [CSS 选择器](https://www.runoob.com/cssref/css-selectors.html)。 |
| 04-03 | 1. jQuery 事件：页面对不同访问者的响应叫做事件，事件处理程序指的是当 HTML 中发生某些事件时所调用的方法。（在元素上移动鼠标，选取单选按钮，点击元素） 在 jQuery 中，大多数 DOM 事件都有一个等效的 jQuery 方法 常用事件函数：click/dblclick/mouseenter/mouseleave/mousedown/mouseup/hover/focus/blur 2. jQuery效果：隐藏、显示、切换，滑动，淡入淡出，以及动画 ，hide:隐藏，带有speed参数的hide方法$().hide(1000,”linear”,function(){}) ; $(*selector*).hide(*speed,callback*); toggle 来回切换hide和show；$(*selector*).toggle(*speed,callback*); callback 参数是隐藏或显示完成后所执行的函数名称； 3. jQuery效果-淡入淡出：Fading方法包括fadeIn/fadeout/fadeToggle/fadeTo 四种（方法中增加时间和回调函数，不透明度） 4. jQuery效果-滑动：滑动方法：slideDown/slideUp/slideToggle; 5. jQuery效果-动画：animate方法$(*selector*).animate({*params*}*,speed,callback*); 操作多个属性/使用相对值/使用预定义值/队列功能（左右上下移动），使用该方法时设置{属性：值}； 6. jQuery-停止动画：$(*selector*).stop(*stopAll,goToEnd*); stop用于所有的动画效果；可以在 stop() 中设置 stopAll 的参数为 true，这样就可以停止所有动画效果而不是只停止当前动画； 7. jQuery Callback:动画执行完成后执行该函数。 8. jQuery-链Chaining：把要执行的动作链在一起$().css().slideUp().slideDown() 9. jQuery-获取内容和属性：jQuery 拥有可操作 HTML 元素和属性的强大方法，jQuery 中非常重要的部分，就是操作 DOM 的能力， 获得内容text/html/val/通过id或者元素标签，属性-attr(); 10. jQuery-设置内容和属性： 11. jQuery-添加元素：添加新内容的四个方法append结尾插入/prepend开头插入/after被选元素之后插入/before被选元素之前（后两个具有自动换行）； 12. jQuery-删除元素：remove（删除被选元素及其子元素）/empty（从被选元素中删除子元素）；过滤被删除的元素（.类名）$("p").remove(".italic"); 13. jQuery-获取并设置类：addClass/removeClass/toggleClass/css;增加/删除/切换 类（定义属性比如字体，颜色等）； 14. jQuery-CSS()方法: css() 方法设置或返回被选元素的一个或多个样式属性, css("*propertyname*"); $("p").css("background-color");设置css属性css("*propertyname*","*value*"); 选择好要设置的元素然后用该语句，设置多个CSS属性css({"*propertyname*":"*value*","*propertyname*":"*value*",...}); 15. jQuery-尺寸：处理元素和浏览器窗口的尺寸，处理尺寸的方法有：width/height/innerWidth/innerHeight/outerWidth/outerHeight; 16. jQuery-遍历：jQuery 遍历，意为"移动"，用于根据其相对于其他元素的关系来"查找"（或选取）HTML 元素。以某项选择开始，并沿着这个选择移动，直到抵达您期望的元素为止。遍历方法中最大的种类是树遍历（tree-traversal）。 17. jQuery-遍历-祖先：向上遍历DOM数方法：parent/parents/parentsUntil 18. jQuery-遍历-后代：向下遍历DOM数方法：children/find; 19. jQuery-遍历-同胞：方法：siblings/next/nextAll/nextUntill/prev/prevAll/prevUntil； 20. jQuery-遍历-过滤：缩小搜索元素的范围，三个最基本的过滤方法是：first(), last() 和 eq()，它们允许您基于其在一组元素中的位置来选择一个特定的元素。其他过滤方法，比如 filter() 和 not() 允许您选取匹配或不匹配某项指定标准的元素。方法:first/last/eq/filter/not/ 21. jQuery-AJAX简介：AJAX 是与服务器交换数据的技术，它在不重载全部页面的情况下，实现了对部分网页的更新。AJAX = 异步 JavaScript 和 XML（Asynchronous JavaScript and XML）。 |
| 04-07 | 1. jQuery-AJAX load方法：load() 方法从服务器加载数据，并把返回的数据放入被选元素中。$(selector).load(URL,data,callback); 2. jQuery-AJAX get/post方法：jQuery get() 和 post() 方法用于通过 HTTP GET 或 POST 请求从服务器请求数据。 Get：从指定的资源请求数据，基本上用于从服务器获得（取回）数据。注释：GET 方法可能返回缓存数据；post：从指定的资源提交要处理的数据，也可用于从服务器获取数据。不过，POST 方法不会缓存数据，并且常用于连同请求一起发送数据。 3. jQuery-noConfilct 方法：noConflict() 方法会释放对 $ 标识符的控制，这样其他脚本就可以使用它了。 4. jQuery-JSONP: Jsonp(JSON with Padding) 是 json 的一种"使用模式"，可以让网页从别的域名（网站）那获取资料，即跨域读取数据。同源策略，它是由 Netscape 提出的一个著名的安全策略，现在所有支持 JavaScript 的浏览器都会使用这个策略。 5. jQuery 实例:jQuery选择器/事件(click,dbclick,mouseenter,mouseleave,mousedown,mouseup,hover,focus,blur)/隐藏显示(hide/show/toggle)/淡入淡出(fadeIn/fadeout/fadeToggle/fadeTo) /滑动(slideDown/slideUp/slideToggle)/动画 (animate/设置多个CSS属性/使用相关值/使用预定义值/)/停止动画 (stop/带参数 )/HTML获取和属性 (cal/attr)/HTML设置内容和属性 (attr/text/html )/HTML添加元素内容 (append/prepend/after/bofore)/移除元素内容 (remove/empty/带参数的)/get和设置CSS类 ( addClass/多个类/removeClass/toggleClass)/CSS方法 (返回CSS属性/设置CSS属性)/尺寸 (width/height/innerWidth/innerHeight/outerWidth)/遍历-祖先 (parent/parents/parentsUntil )/遍历-后代 (children/find)遍历-同胞 (siblings/next/nextAll/nextUntil)/Ajax-load(load/callback)/Ajax get post (get/post)/ 6. 参考手册-选择器：元素/id/类/ 7. 事件方法：<https://www.runoob.com/jquery/jquery-ref-events.html>； 8. 效果方法：animate/clearQueue/delay/dequeue/fadeIn/fadeout/fadeTo/fadeToggle/finish/hide/queue/show/slideDown/slideToggle/slideUp/stop/toggle; 9. HTML/CSS方法：增加移除改变，<https://www.runoob.com/jquery/jquery-ref-html.html>； 10. 遍历方法：<https://www.runoob.com/jquery/jquery-ref-traversing.html>； 11. AJAX方法：<https://www.runoob.com/jquery/jquery-ref-ajax.html>； 12. 杂项方法： 13. 属性： 14. 插件：<https://www.runoob.com/jquery/jquery-plugin-validate.html>； |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Json: 用于存储和传输数据的格式，通常用于服务端向网页传输数据。

Ajax：不重载全部页面的情况下实现对部分网页的更新。

Dom：访问HTML文档所有元素。

Bom：浏览器对象模型使js与浏览器对话。

DOM:

|  |  |
| --- | --- |
|  |  |
| 04-07 | 1. DOM-简介："W3C 文档对象模型 （DOM） 是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。"，*HTML DOM 是关于如何获取、修改、添加或删除 HTML 元素的标准*  2. DOM-节点：在 HTML DOM 中，所有事物都是节点。DOM 是被视为节点树的 HTML。整个文档是一个文档节点，每个 HTML 元素是元素节点，HTML 元素内的文本是文本节点，每个 HTML 属性是属性节点，注释是注释节点。节点树，节点树中的节点彼此拥有层级关系。我们常用**父（parent）**、**子（child）**和**同胞（sibling）**等术语来描述这些关系。父节点拥有子节点。同级的子节点被称为同胞（兄弟或姐妹）。在节点树中，顶端节点被称为根（root）。每个节点都有父节点、除了根（它没有父节点）。一个节点可拥有任意数量的子节点。同胞是拥有相同父节点的节点。  3. DOM-方法：HTML DOM 方法是我们可以在节点（HTML 元素）上执行的动作。HTML DOM 属性是我们可以在节点（HTML 元素）设置和修改的值。一些常用的 HTML DOM 方法：getElementById(id) - 获取带有指定 id 的节点（元素）appendChild(node) - 插入新的子节点（元素）removeChild(node) - 删除子节点（元素）一些常用的 HTML DOM 属性：innerHTML - 节点（元素）的文本值parentNode - 节点（元素）的父节点 childNodes - 节点（元素）的子节点 attributes - 节点（元素）的属性节点；  4. DOM-属性：属性是节点（HTML 元素）的值，能够获取或设置。可通过 JavaScript （以及其他编程语言）对 HTML DOM 进行访问。所有 HTML 元素被定义为对象，而编程接口则是对象方法和对象属性。获取元素内容的最简单方法是使用 innerHTML 属性。innerHTML 属性对于获取或替换 HTML 元素的内容很有用。  5．DOM-访问：主要方法有： getElementById() / getElementsByTagName()/ getElementsByClassName()  6. DOM-修改：修改 HTML = 改变元素、属性、样式和事件。  7. DOM-修改HTML内容：通过 HTML DOM，JavaScript 能够访问 HTML 文档中的每个元素。改变HTML内容/样式，使用事件来修改；  8. DOM-元素：添加、删除和替换 HTML 元素，如需向 HTML DOM 添加新元素，首先必须创建该元素，然后把它追加到已有的元素上。创建新元素p->创建文本节点->向p元素追加文本节点->向已有元素追加新元素->向已存在的元素追加新元素。在DOM中增删查改都是针对节点而言。  9. DOM-事件：HTML事件的例子：当用户点击鼠标/网页已经加载时/图片已经加载/鼠标移动到元素上/输入字段被改变/表单被提交/用户触发按键/  10.DOM-导航：getElementsByTagName() 方法返回*节点列表*。节点列表是一个节点数组。childNodes 和 nodeValue  11.DOM-总结：使用 HTML DOM 来增强网站的动态交互性，在客户端（在浏览器中）使用脚本来创建动态网页，也可以通过在服务器上使用脚本来增加网页的动态性，通过服务器端脚本，您能够编辑、添加或更改网页内容。您能够对提交自 HTML 表单的数据做出响应，访问数据或数据库，并向浏览器返回结果，为不同的用户定制页面。  12.DOM-实例：Document对象/Anchor对象/Area对象/Base对象/Button对象/Form对象/Frame，IFrame对象/Image对象/Event对象/Option，select对象/Table，TableHeader，TableRow，TableData对象。  13.对象：<https://www.runoob.com/jsref/jsref-tutorial.html>； |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

AJAX:

|  |  |
| --- | --- |
| 04-07 | 1.简介：AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术，通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新，XMLHttpRequest 对象 (异步的与服务器交换数据)，JavaScript/DOM (信息显示/交互)，CSS (给数据定义样式)，XML (作为转换数据的格式)；  2.实例：  3.创建对象：XMLHttpRequest 是 AJAX 的基础。创建XMLHttpRequest 对象  4.向服务器发送请求：如需将请求发送到服务器，我们使用 XMLHttpRequest 对象的 open() 和 send() 方法：open(*method*,*url*,*async*)，与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。可以使用post的情况：无法使用缓存文件（更新服务器上的文件或数据库），向服务器发送大量数据（POST 没有数据量限制），发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠。异步访问。  5.服务器响应：如需获得来自服务器的响应，请使用 XMLHttpRequest 对象的 responseText（获得字符串形式的响应数据） 或 responseXML（获得XML形式响应数据） 属性。  6.readyState：当请求被发送到服务器时，我们需要执行一些基于响应的任务。每当 readyState 改变时，就会触发 onreadystatechange 事件。每当 readyState 改变时，就会触发 onreadystatechange 事件。readyState 属性存有 XMLHttpRequest 的状态信息。 XMLHttpRequest 对象的三个重要的属性：onreadystatechange（存储函数（或函数名），每当 readyState 属性改变时，就会调用该函数。），readyState（存有 XMLHttpRequest 的状态。从 0 到 4 发生变化。0: 请求未初始化，1: 服务器连接已建立，2: 请求已接收，3: 请求处理中，4: 请求已完成，且响应已就绪），status（200: "OK"，404: 未找到页面）  7.ASP/PHP:  8.数据库：AJAX 可用来与数据库进行动态通信。  9.XML |

JSON:

|  |  |
| --- | --- |
| 04-07 | 1.简介：JSON: JavaScript Object Notation(JavaScript 对象表示法) 是存储和交换文本信息的语法。对于 AJAX 应用程序来说，JSON 比 XML 更快更易使用，  2.语法：JSON 语法是 JavaScript 语法的子集。语法规则：数据在名称/值对中，数据由逗号分隔，大括号保存对象，中括号保存数组，JSON 数据的书写格式是：名称/值对。JSON 值可以是：数字/字符串/逻辑值/数组/对象/null；JSON 数字可以是整型或者浮点型；JSON 对象在大括号（{}）中书写，对象可以包含多个名称/值对：{ "name":"菜鸟教程" , "url":"www.runoob.com" }，JSON数组在中括号中书写，数组可包含多个对象，JSON 文件的文件类型是 ".json"，JSON 文本的 MIME 类型是 "application/json"；  3.JSON vs XML: JSON 和 XML 都用于接收 web 服务端的数据。JSON 和 XML在写法上有所不同，JSON优点：不需要结束标签，更加简短，读写速度更快，可以使用数组。XML 需要使用 XML 解析器来解析，JSON 可以使用标准的 JavaScript 函数来解析。  4.JSON对象：JSON 对象使用在大括号({})中书写。对象可以包含多个 key/value（键/值）对。key 必须是字符串，value 可以是合法的 JSON 数据类型（字符串, 数字, 对象, 数组, 布尔值或 null）。访问对象值：用点号（.）用中括号（[]）来访问对象的值；for-in 来循环对象的属性，嵌套JSON对象，修改值，删除（delete；）对象属性，  5.JSON数组：JSON 数组在中括号中书写。JSON 中数组值必须是合法的 JSON 数据类型（字符串, 数字, 对象, 数组, 布尔值或 null）。使用索引值来访问数组，使用 for-in 来访问数组，JSON 对象中数组可以包含另外一个数组，或者另外一个 JSON 对象，使用索引值来修改数组值，使用 delete 关键字来删除数组元素；  6.JSON.parse：JSON 通常用于与服务端交换数据。在接收服务器数据时一般是字符串。我们可以使用 JSON.parse() 方法将数据转换为 JavaScript 对象。JSON.parse(text[, reviver])（text:必需， 一个有效的 JSON 字符串。reviver: 可选，一个转换结果的函数， 将为对象的每个成员调用此函数。）使用AJAX从服务段获取JSON数据，JSON 不能存储 Date 对象。  7. JSON.stringify：用 JSON.stringify() 方法将 JavaScript 对象转换为字符串。JSON.stringify(value[,replacer[,space]]),value：必需， 要转换的 JavaScript 值（通常为对象或数组）。replacer: 可选。用于转换结果的函数或数组。space: 可选，文本添加缩进、空格和换行符。Stringify和prase相反。  8.使用：JSON 最常见的用法之一，是从 web 服务器上读取 JSON 数据（作为文件或作为 HttpRequest），将 JSON 数据转换为 JavaScript 对象，然后在网页中使用该数据。  9.JSONP: Jsonp(JSON with Padding) 是 json 的一种"使用模式"，可以让网页从别的域名（网站）那获取资料，即跨域读取数据。同源策略，它是由 Netscape 提出的一个著名的安全策略，现在所有支持 JavaScript 的浏览器都会使用这个策略。  10.PHP JSON:  11.JSON 格式化工具： |

VUE

|  |  |
| --- | --- |
|  |  |
| 2020-03-18 | 1.框架和库的区别， |
| 04-07 | 2.Vue.js起步：每个 Vue 应用都需要通过实例化 Vue 来实现。语法格式：var vm=new Vue({  代码块  }) |
| 04-13 | 3.模板语法：Vue.js 使用了基于 HTML 的模版语法，允许开发者声明式地将 DOM 绑定至底层 Vue 实例的数据。Vue.js 的核心是一个允许你采用简洁的模板语法来声明式的将数据渲染进 DOM 的系统  插值/属性/表达式/指令/参数/用户输入/过滤器/缩写  **v-model** 指令用来在 input、select、textarea、checkbox、radio 等表单控件元素上创建双向数据绑定，根据表单上的值，自动更新绑定的元素的值。  v-bind:href=:href v-on:click=@click  4.条件语句：v-if,v-else,v-else-if;  5.循环语句：v-for 指令需要以 **site in sites** 形式的特殊语法， sites 是源数据数组并且 site 是数组元素迭代的别名。v-for 可以绑定数据到数组来渲染一个列表 |
| 04-14 | 6.计算属性:computed/method  7.监听属性：watch  8.样式绑定：class 与 style 是 HTML 元素的属性，用于设置元素的样式，我们可以用 v-bind 来设置样式属性。Vue.js v-bind 在处理 class 和 style 时， 专门增强了它。表达式的结果类型除了字符串之外，还可以是对象或数组。绑定在data中 {属性：value}，通过数组的形式绑定元素样式，也是在data中写样式。  9.事件处理器：事件监听可以使用 v-on 指令 v-on 可以接收一个定义的方法来调用 除了直接绑定到一个方法，也可以用内联 JavaScript 语句 Vue.js 为 v-on 提供了事件修饰符来处理 DOM 事件细节，如：event.preventDefault() 或 event.stopPropagation()。Vue.js通过由点(.)表示的指令后缀来调用修饰符。 按键修饰符  10.表单：用 v-model 指令在表单控件元素上创建双向数据绑定。  11.组件：可以扩展 HTML 元素，封装可重用的代码。  组件系统让我们可以用独立可复用的小组件来构建大型应用，几乎任意类型的应用的界面都可以抽象为一个组件树  注册一个全局组件语法格式：Vue.component(tagName, options) //子组件，（组件名称，对象），把一段公共的模块提出处理，封装在一起，然后多个页面去引用这个模块，避免重复写代码。组件传参，父子组件相互访问等。 |
| 04-15 | 1. 自定义指令：核心指令（v-model/v-show），钩子函数，   bind: 只调用一次，指令第一次绑定到元素时调用，用这个钩子函数可以定义一个在绑定时执行一次的初始化动作；  inserted: 被绑定元素插入父节点时调用（父节点存在即可调用，不必存在于 document 中）。  update: 被绑定元素所在的模板更新时调用，而不论绑定值是否变化。通过比较更新前后的绑定值，可以忽略不必要的模板更新（详细的钩子函数参数见下）。  componentUpdated: 被绑定元素所在模板完成一次更新周期时调用。  unbind: 只调用一次， 指令与元素解绑时调用。   1. 路由：指向的意思，eg：点击页面上的home按钮时，页面中就要显示home的内容；home按钮=>home内容；也算是一种映射；分为两部分（点击部分与点击之后部分），三个基本感念（route,routes,router）route:一条路由，home按钮=>home内容；routes：一组路由把每一条路由组合起来，形成一个数组[{home按钮=>home 内容}]，router 是一个机制，是用来管理路由的。 2. 过渡动画： |
| 04-16 | 1. 混入（node.js/npm/vue的安装与应用）/挂载点：需要操作的元素eg<div id=app></div>. Vue实例只会处理挂载点下的内容，挂载点内部的内容叫模板内容。 2. Ajax(axios):get方法/post/执行多个并发请求/ 3. 指令：v-if/v-for/v-on/ :模板(<标签 v-show="条件">如果达到条件要显示的内容</标签>)，当表达式的值改变时=>响应式的作用于DOM， |
| 04-17 | 1. Ajax(vue-resource): 2. 响应借口：vue.set设置对象，用来解决vue无法检测添加属性的限制，vue.set(target,key,value) 3. Vue.js实例： 融会贯通。 |

2020-04-20

|  |  |
| --- | --- |
|  | 1. 布局比较合理，兼容，兼顾标签语义化，JS交互效果（焦点图，选项卡，表单验证之类的简单效果）。  2.css3：选择器/盒模型/背景和边框/文字特效/2d3d转换/动画/多列布局/用户界面。 |
| 05-09 | 1.css选择器：id/类/标签/后代/子代，先选好类然后在选标签（.nav a（中间用空格隔开））; |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |