# 认识JavaScript

1. JavaScript是一门面向web的弱类型编程语言，出生于1995年；

强类型编程语言：一种总是强制类型定义的语言。而强类型则没这样的规定,不同类型间的操作有严格定义,只有相同类型的变量才能操作,虽然系统也有一定的默认转换,当绝没有弱类型那么随便

弱类型编程语言： 一种类型可以被忽略的语言。弱类型的语言的东西没有明显的类型,他能随着环境的不同,自动变换类型

ps:弱类型代码简单,但因为变量没有确定的类型,所以容易出错!

强类型代码复杂，但因为有严格定义所以不容易出错

1. JavaScript可以称作编程史上使用最广泛的编程语言：桌面系统、游戏机、平板电脑、智能手机的浏览器均包含了JavaScript的解释器；
2. JavaScript是前端开发技术必须掌握的技术（个人认为是最核心的技术）：描述网页内容的HTML、描述网页样式的CSS和描述网页行为的JavaScript；
3. JavaScript和java是完全不同的两门编程语言，只是他们的语法看起来有点类似（逗比，很多编程语言的语法都类似好吧）；
4. ECMA：欧洲计算机制造商协会；ES:ECMAScript；
5. ECMAScript是标准（如同HTML语言W3C标准），我们学习的JavaScript是该标准的具体实现(类似通信标准：华为手机为什么可以使用移动的卡，因为有行业标准)；
6. 目前通用的标准是ES5，ES6也已经发布，更多API在完善和定稿中，浏览器还没有完全兼容（ES后面也发布了其他版本）
7. 我们现在学习的是ES5标准的JavaScript，当然还是要学习一些ES6的语法及规范，ES6也必然很快成为未来流行的新标准；
8. JavaScript的解释器（引擎，engine）也有版本号，Google将他的解释器叫做v8，每个浏览器都内置有JavaScript的解释器，所以浏览器可以执行JavaScript；

JavaScript引擎是浏览器的组成部分之一。因为浏览器还要做很多别的事情，比如解析页面、渲染页面、Cookie管理、历史记录等等。那么，既然是组成部分，因此一般情况下JavaScript引擎都是浏览器开发商自行开发的。比如：IE9的Chakra、Firefox的TraceMonkey、Chrome的V8等等。JavaScript解析引擎就是能够“读懂”JavaScript代码，并准确地给出代码运行结果的一段程序

# JS如何引入到页面

1. 行级：给标签设置对应的属性，属性值是要执行的JavaScript代码；

一般不用，用时：a标签实现锚点跳转

1. 嵌入：使用script标签，标签需要闭合，标签内容是要执行的JavaScript代码；
2. 引入：使用script标签，标签需要闭合，设置属性src，src的值是js文件路径，如：./js/my.js；
3. 关于js文件：js代码可以直接写在一个独立的文件里面，该文件就是js文件，后缀是.js；

ps:js文件指的是文件里的内容是JS代码，与文件后缀无关

# JS页面内容输出方式:

1. 页面弹窗：alert('弹出的内容');
2. 页面输出内容：document.write('输出到页面的内容')；
3. 弹出选择框：confirm('你确定执行该操作？'); 返回TRUE(确定)或FALSE（取消）；

# JS控制台信息输出方式：

1. console.log('打印"日志"信息到控制台');
2. console.error('打印"错误"信息到控制台');
3. console.warn('打印"警告"信息到控制台');
4. console.info('打印"消息"到控制台');
5. JavaScript清空控制台消息：clear();

# 如何获取页面元素？

1. 获取单个元素：document.querySelector('选择器')，如果该选择器对应多个元素，选中所有但只返回第一个（返回值是元素）；
2. 获取多个元素：document.querySelectorAll('选择器')，获取选择器对应的全部元素

PS:返回值是数组不是元素，所有不能直接arr.style,只能arr[1].style；

# 如何获取和改变元素的内容？

1. 获取元素内容：document.querySelector('选择器').innerHTML;
2. 修改元素内容：document.querySelector('选择器').innerHTML = '新内容';

# 如何改变元素的样式？

1. 语法是：document.querySelector('选择器').style.属性 = '属性值';
2. 其中属性是CSS样式中的属性，如：display、color、width、height等；
3. 如果属性有横线(-)，如：background-color、font-size、border-radius、font-weight，则把横线去掉，同时横线后面的第一个字母大写(驼峰命名法)

如：backgroundColor、fontSize、borderRadius、fontWeight；

例子：

1. 改变字体颜色：document.querySelector('选择器').style.color = '#FF0000';
2. 改变背景颜色：document.querySelector('选择器').style.backgroundColor = '#000000';
3. 字体加粗：document.querySelector('选择器').style.fontWeight = 'bolder'; 隐藏元素：document.querySelector('选择器').style.display = 'none';

# 事件的概念

事件是指可以被JavaScript监测到的行为；

如：鼠标点击、鼠标移动、鼠标移入/离开、键盘按键、页面载入等；

# JavaScript完成一个事件的三要素：

结合现实事件：小王，把灯打开一下；

1. 事件源：操作对象，名词，对应：开关;
2. 事件：做什么动作，动词，对应：摁一下；
3. 事件处理：背后要做哪些工作，具体要干什么，这里就是我们要写代码具体实现的功能了，对应：接通火线，把灯点亮；

例子：



# 如何连接字符串？

加号：+；（与Java用法相同）

# 什么是词法结构

1. 编程语言的词法结构是一套基础性规则，用来描述如何使用这门语言来编写程序；
2. 词法结构是语法的基础，定义了如何定义变量、写注释、分割语句等基本规则；

# JavaScript是严格区分大小写的

1. JavaScript区分大小写，包括关键字、变量、函数名、所有标识符，够任性； （HTML不区分大小写）
2. 所以，querySelector的S是大写，你写成小写就会报错；
3. 所以，alert()全部是小写，你写一个大写试试；
4. 所以，myname、myName、mynamE、MyName他们真不是一个东西，是四个东西；

# JavaScript会忽略标识符前后的空格；

1. 空格是为了让代码有整齐一致的缩进，形成统一的编码风格，让代码更具可读性；
2. 你可以 document . querySelector('选择器')这样；
3. 你还可以document. querySelector ('选择器')这样；
4. 但是你不可以document.query Selector('选择器')这样；
5. 所以，你要搞清楚JavaScript是忽略标识符前后的空格；
6. 在标识里面加空格，是把一个标识符分割成了两个或多个标识符；
7. 嗯嗯，一般加空格是为了代码排版风格，不要手贱乱加空格(^\_^)；

# JavaScript支持两种注释方式；

1. //这里是注释内容 ：单行注释；
2. /\*注释内容\*/ ：多行注释
3. 注释的部分不会执行，合理的注释能显著提高代码的可读性；
4. 浏览器通过源文件可以看到注释内容，所以什么该注释什么不该注释要拿捏好；

# JavaScript中直接使用的数据值叫做直接量；

1. 12306;
2. '我要学习JavaScript';
3. true;
4. null;
5. [1,2,3,4,5];

# 标识符

1. 标识符就是一个名字;
2. JavaScript使用标识符对变量和函数(参数)进行命名;
3. JavaScript的标识符必须使用字母、下划线(\_)、美元符号($)开始;
4. 后续的可以是字母、数字、下划线或美元符号($);
5. 数字不允许作为首字符出现，以便可以轻易区分开变量和数字，如：12345;
6. 合法的标识符：myname、\_age、$classname、abc、hqyj\_h5;
7. 不合法的标识符：5myname;
8. 也可以使用非英语来定义标识符：var π = 3.14;
9. 标识符最好(或者说务必)见名知意（不要拼音缩写，用英文单词或全拼）：开始只有你和上帝知道你写的代码什么意思，两个月后就只有上帝知道你的代码什么意思了;

# 保留字

1. JavaScript默认的把一些标识符拿来作为自己的关键词，如：for、if、while，这些关键词称作保留字;
2. 我们在写代码定义标识符的时候不能使用保留字;

# 可选OR必选的分号

1. JavaScript使用分号(;)将语句分隔开;
2. 分号并不是必须的，换行也可以表示一个语句结束，执行时会自动填充分号;
3. JavaScript并不是在所有的换行处都填充分号： 只有在缺少了分号无法正确解析时才会填充分号;
4. 所以，你还有必要吝啬加分号吗？
5. 所以，请使用分号来明确标记语句的结束，即使在并不完全需要分号的时候也是如此;