回顾

```
页面中,超链接,图片,列表
    超链接:
        <a href="路径位置" target="_self _blank">
    图片:
        <img src="图片路径" alt="当图片显示不出来显示问题">
    列表:
        无序列表:
            <ul>
                <li>
        有序列表:
            <ol>
                <li>
    使用表格:
        <table border="1px" cellspacing="单元格之间的空隙" cellpadding="内容与单元格的空隙"
          bgcolor="red" align="right">
            <caption>
            <tr>
                <td>
                <th>
        合并单元格:
            rowspan="2" 行合并
            colspan="2" 列合并

HTML表单【重点】
    <form action="提交的位置"  method="get post">
        get:
            不安全。参数放在url地址栏上。对长度有限制
        post:
            安全。参数放在请求体中。对长度没限制
        <input>
            不同的type去表示不同的效果
            type="text"     普通文本框
            type="password" 密码框
            type="radio"    单选框  value
            type="checkbox" 多选框  value
            type="file"     文件
            type="date"     日期
            type="hidden"   隐藏域  value
            type="submit"   提交
            type="image"    图片提交
            type="reset"    重置
            type="button"   按钮
        <select>            下拉列表
            <option>        选项    value
        <textarea>          文本域

        所有要传递给后台的数据，都是以键值对的方式，name是键  value是值
```

```
CSS基本选择器
    标签选择器
        div{
            获取所有的div元素
        }
    类选择器
        .aaa{
            获取class=aaa的元素
        }
    id选择器
        #aaa{
            获取id=aaa的元素
        }
```

一 JavaScript概述

作用：可以来增强用户和html页面的交互过程，可以来控制html元素，让页面有一些动态的效果，增强用户的体验。

1.JavaScript历史

- 1995年，Netscape（网景）开发了一门客户端脚本语言：LiveScript。后来，请来SUN公司的专家，修改LiveScript，命名为JavaScript
- 1996年，微软抄袭JavaScript开发出JScript语言
- 1997年，ECMA(欧洲计算机制造商协会)，制定出客户端脚本语言的标准：ECMAScript，就是统一了所有客户端脚本语言。
- 我们现在学习的JavaScript = ECMAScript + (BOM+DOM) JavaScript自己特有的东西

2.JavaScript特点

1. js源码不需要编译，浏览器可以直接解释运行
2. js是弱类型语言，js变量声明不需要指明类型

   （在java中定义的时候要定义不同的类型int byte char ，在js中所有变量都用同一个词来表示）

3.JavaScript组成

组成部分	作用
ECMA Script	构成了JS核心的语法基础
BOM	Browser Object Model 浏览器对象模型，用来操作浏览器上的对象
DOM	Document Object Model 文档对象模型，用来操作网页中的标签

二 JavaScript基础语法

1. HTML引入JS

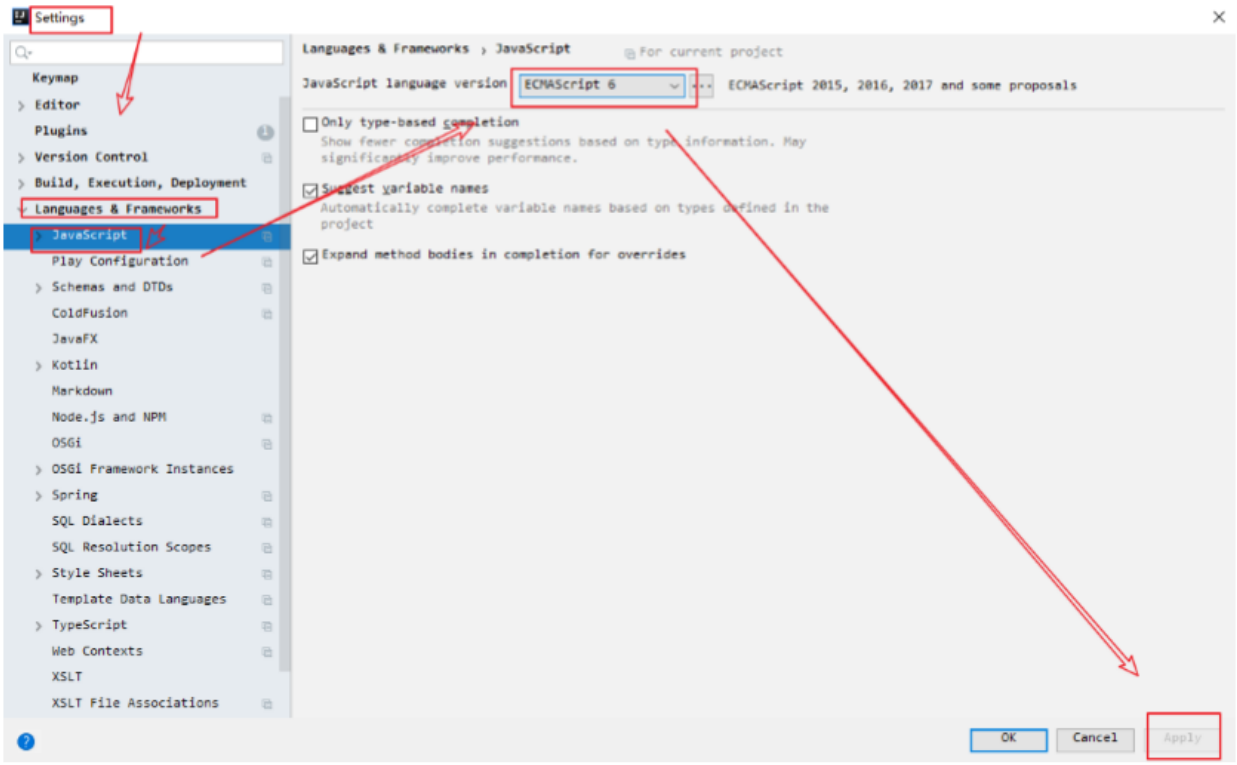
```
1.内部js
        <head>
            <meta charset="UTF-8">
            <title>Title</title>

            <!--内部js-->
            <script>
                alert(123);
            </script>
        </head>

2.外部js
    <script src="js的路径"></script>

注意事项:
    1.上面的js不能获取下面的标签，页面是从上往下加载的
    2.外部js和内部js不要写在同一个<script>中
```

2. JS三种输出方式

```
// 弹出窗口
alert("石原里美");
// 在body中显示
document.write("石原里美");
// 在控制台输出
console.log("石原里美");
```

3. JS变量声明

- 使用之前先设置idea适配es6的语法
    - 在es5以及之前数据的定义都使用var，在es6中变量可以使用let, 常量可以使用const(var依然可以使用)



- 数据声明的格式

```
变量:
      ES5: var a = 10;
      ES6: let a = 10;
常量:
     const a = 10;
         注意事项:
              定义的时候必须赋值，赋值之后不能再修改
弱类型语言:
     在声明的时候不用指定数据的类型，所有变量都定义成let或var
typeof:
     可以获取变量的数据类型
      let a = "sdfsdfs";
      alert(typeof a); //获取a的数据类型
```

4. JS数据类型

js也分为了基本数据类型(原始数据类型)和引用数据类型

```
原始数据类型:
     number    :数字
     string    :字符串
     boolean   :布尔
     undefined:未定义的值
     null      :代表不存在
         null打印的数据类型显示object,是js的一个bug

其他都成为是对象类型
     object
```

5. JS运算符

```
js中【数据】如果不符合【运算符要求的类型】，那么会自动转成对应的【运算符要求类型】。

1.一元运算符
    ++   --  +(正号)
    前置++: 先++，后运算
    后置++: 先运算,后++

    别的类型转成数字:
        string-->
            按照字符串字面值转成对应的数字，如果字面值不是数字转成NaN(not a number)
        boolean-->
            true是1，false是0

2.算术运算符
    +  -  *  /   %

    和java基本一致
    区别: js整数和小数都是number,两个整数计算的结果也可以出现小数

3.赋值运算符
    =    +=   -=   *=    /=    %=

    和java中用法一样

4.比较运算符
    >    <    >=    <=   ==  !=  ===

    ==: 等于    比较值是否等于
    ===:全等于 比较值和类型是否等于

    null和undefined的值是相同的，类型是不同的

5.逻辑运算符
    &&    ||    !
    别的类型转成boolean:
        number-->:  0是false,别的都是true
        string-->:  ""空字符串是false,别的都是true
        null和undefined都是false


6.三元运算符
    let 变量 = 表达式 ? 结果1 : 结果2;

    和java中用法一样
```

6. JS流程控制

js中的代码流程分为顺序结构，判断结构，循环结构

1. 条件判断

```
if条件判断
    if()...
    if()...else...
    if().. else if()...
    和java中用法一样

switch条件判断
    和java中基本一样
    js中switch写原始数据类型 一般就是数字和字符串
```

2.循环语句

```
1.普通for循环
        for(let i=0; i<5; i++){
            document.write(i+" ");
        }
2.增强for循环
         for(let a of arr){
             document.write(a+" ");
         }

3.索引for循环
        for(let i in arr){
            document.write(arr[i] + " ");
        }

4.while循环
        let a = 0;
        while(a < 5){
            alert(a);
            a++;
        }

5.do..while循环
        let a = 0;
        do{
            alert(a);
            a++;
        }while(a < 5);

6.循环跳转语句
        break;
            跳出当前循环，循环结束
        continue;
            跳过本次循环，进入下次循环

        for(let i = 0 ; i < 10; i++){
            //偶数被跳过
            //if(i % 2 == 0){
            //    continue;
            //}
```

```
            //i等于4就把循环结束
            if(i == 4){
                break;
            }
            alert(i);
        }
```

三. JS常用对象

1.JS函数

JS函数也可以理解为JS方法，JS中函数也是一种对象

```
函数的定义:
    方式一:
        function 函数名(参数){
            方法体
            return 值;
        }
    方式二:
        let 函数名 = function(参数){
            方法体
            return 值;
        }
        这个方式二相对于定义对象,必须要在上面定义在下面调用
    例子:
        //方式一:
        function method() {
            alert("石原里美");
        }

        //方式二:
        let method2 = function () {
            alert("王众");
        }

函数的属性:
    length:表示的是形式参数的个数
    例子:
        alert(method.length);
        alert(method2.length);
函数的调用:
    方法名(参数);

函数的特殊:
    1. 方法定义是，形参的类型不用写,返回值类型也不写
    2. 方法是一个对象，如果定义名称相同的方法，会覆盖
    3. 在JS中，方法的调用只与方法的名称有关，和参数列表无关
    4. 在方法声明中有一个隐藏的内置对象（数组），arguments,封装所有的实际参数
```

2. String 对象

```
字符串拼接:
    let s3 = "黑马" + 6 + 6 + 6;
    let s4 = `黑马${666}`;    //直接把要拼接的内容放在{}中
方法:
    substring():   截取
    toLowerCase(): 转成小写字母
    toUpperCase(): 转成大写字母
    trim():        去除两端的空白
    charAt():      根据索引获取一个字符
    java中的字符串方法在这里基本一样。。。
```

3. Array 对象

```
创建对象:
    1.直接定义数字
        let arr = [11,22,33,44,55];
    2.new方式
        //let arr = new Array(11,22,33,44,55);
   3.new方式指定数组长度
        //如果()中只有一个数字,认为这是长度
        let arr = new Array(5);

方法:
    join(): 把数组变成字符串打印，如果不指定分隔符默认是,
    push(): 给数组的末尾添加元素

属性:
    length: 数组的长度

特点:
    1.数组的长度可变
    2.数组的元素可以是任意类型
```

4. Date 对象

```
创建对象:
    let date = new Date();
方法:
    toLocaleString():   根据系统的国家转成对应的格式
    getTime():          获取时间的毫秒值    距离1970年1月1日的时间。。
```

5. Math 对象

```
Math使用不需要创建对象

方法:
    random():   生成一个0~1的随机数。包含0但是不包含1
    ceil():     向上取整
    floor():    向下取整
    round():    四舍五入

属性:
    PI  :圆周率
```

6. 全局函数

```
全局函数就是不需要对象调用的函数(方法)
方法:
    encodeURI():    编码【把能看懂的文字变成看不懂的】
    decodeURI():    解码【把看不懂的文字变成看得懂的】
        URI 统一资源标志符    URL 统一资源定位符
            URL是一种URI

    parseInt():     把字符串转成整数数字
        从前往后转，如果遇到不是数字的字符就停止

    isNaN():        判断一个变量是否是NaN
        alert(NaN == NaN);   //NaN六亲不认,连自己都不认。 false

    eval():         把字符串解析成js代码去执行
```

7.案例：轮播图

一共3张图片，实现每过3秒中切换一张图片的效果

```
<body>
<!-- 图片标签 -->
<img id="a1" src="img/banner_1.jpg">
<script>
    //根据元素的id获取一个元素
    let img = document.getElementById("a1");

    //定义数字
    let num = 1;
    //定义方法
    function aaa() {
        num++;
        if(num > 3){
            num = 1;
        }
        //改变img的src属性
        img.src = "img/banner_"+num+".jpg";
    }
```

```
    //计时器
    //方法里面有两个参数,第一个参数是方法对象,第二个参数是时间的毫秒值
    //每3秒执行一次aaa方法
    setInterval(aaa,3000);

</script>
</body>
```

四. JS事件

功能：JS可以监听用户的行为,并调用函数来完成用户交互功能

1.事件监听机制

```
概念：某些元素被执行了某些操作后，触发某些代码的执行。
xxx被xxx了,我就xxx
    我方水晶被摧毁了,我就骂队友
    敌方水晶被摧毁了,我就吹自己

    事件：    某些操作，比如单击，双击
    事件源：某些元素/组件，比如按钮，图片
    监听器：某些代码，当事件发生之后执行的代码
    注册监听：把事件、事件源、监听器联系起来，当事件源上发生了事件就执行监听器代码
```

2. 常用事件

```
1. 点击事件:
        1. onclick: 单击事件
        2. ondblclick: 双击事件
2. 焦点事件
        1. onblur: 失去焦点
        2. onfocus:元素获得焦点。

3. 加载事件:
        1. onload: 一张页面完成加载
            window表示整个页面,可以等整个页面加载完再让js执行

4. 鼠标事件:
        1. onmousedown   鼠标按钮被按下。
        2. onmouseup     鼠标按键被松开。
        3. onmousemove   鼠标被移动。
        4. onmouseover   鼠标移到某元素之上。
        5. onmouseout    鼠标从某元素移开。
5. 键盘事件:
        1. onkeydown     某个键盘按键被按下。
        2. onkeyup       某个键盘按键被松开。
            特殊点:
                如果用的是键盘事件，方法有参数是被按下的键
                    有一个属性叫keyCode是按下键的对应的数值

6. 选择和改变
```

        1. onchange 域的内容被改变【文本框内容改变或下拉框选项改变】
        2. onselect 文本被选中。 【文本框的内容被选中】

7. 表单事件
        1. onsubmit 表单被提交。【加在form上】
        2. onreset 表单被重置。【加在form上】