DOM

属性操作

1. 样式属性
    1. 行内（用的少）

        元素.style.css属性 = 值
    2. 样式表（用的多）

        元素.className = 类名

        className操作相对style的优势：

        1. 效率高（绘制的次数少）
        2. 操作简捷

2. 内容操作

   innerHTML

   innerText/textContent

value

3. 合法属性与非法属性的操作

   合法属性操作:

   获取: 元素.属性名

   修改: 元素.属性名 = 新值

   属性名是对象存在的属性

   增加: 元素.属性名 = 新值

   属性名是对象不存在的属性

```
<a href="https://www.baidu.com"
      target="_blank"
      title="baiduyixia" id="a1"
      goudan="dachui">百度一下</a>
<script>
  var oA =
document.getElementById('a1');
  // 获取
  console.log(oA.href);
  console.log(oA["href"]);
  // console.log(oA.target);
  // console.log(oA.title);
  // console.log(oA.id);
  // 修改
```

```
    // oA.title = "百度一下";

    // 新增
    oA.index = 100;
    console.log(oA.index);// 100
</script>
```

非法属性（开发自己往标签上添加上属性）的操作：

不能用点操作，获取不到，结果是undefined

```
<a id="a1"
    class="box"
     goudan="dachui">
  百度一下
</a>
<script>
  var oA =
document.getElementById('a1');
  console.log(oA.goudan);//
undefined
</script>
```

操作非法属性需要借助下面一组属性操作的方法：

下面这一组方法既可以操作合法属性，又可以操作非法属性，

这一组方法通常用来操做非法属性，如果是合法属性，用对象.属性的方式获取，写着简单

getAttribute(key)：获取属性值

```
<a id="a1"
    goudan="dachui"
    abc="123">百度一下</a>
<script>
  var oA =
document.getElementById('a1');
  var val =
oA.getAttribute('goudan');
  console.log(val);// "dachui"
  val = oA.getAttribute('abc');
  console.log(val);// "123"
</script>
```

setAttribute(key,value)：设置属性

removeAttribute(key)：删除属性

```
<a id="a1"
      goudan="dachui"
      abc="123">百度一下</a>
<script>
  var oA =
document.getElementById('a1');
  // 获取
```

```
  var val = oA.getAttribute('abc');
  // console.log(val);
  // 修改 /设置
  // oA.setAttribute("abc","456");
  // 新增
  // oA.setAttribute("index","1");
  // 删除
  oA.removeAttribute("abc");

  console.log(oA.getAttribute("class"));// "box"
  console.log(oA.className);// "box"
</script>
```

事件

1. 概念：

   预先定义好的一组行为，在特定的时机下被触发，例如：单击，双击，移入，移出...

2. 三要素：

   1. 事件源：事件作用在谁身上
   2. 事件名：单击、双击、移入、移出...

    3. 事件处理程序：事件触发了要做的事
3. 如何给元素添加事件有两种：
    1. 事件注册

        元素.事件名 = function() {

        // do something

        };
    2. 事件绑定
4. 例子；

```
<div id="wrap">

</div>
<script>
  // 需求：  单击wrap弹窗1
  /*
      事件源：div#wrap
      事件名：单击  onclick
      事件作用的结果（事件处理程序）：事件触发了，你要做什      么，弹窗1
    */
  var oWrap = document.getElementById("wrap");
  oWrap.onclick = function() {
    alert(1);
  };
```

```
15 </script>
```

5. 事件分类

   鼠标事件：

   单击(onclick)、双击(ondblclick)、移入(onmouseenter/onmouseover)、移出(onmouseleave/onmouseout)、按下(onmousedown),

   抬起(onmouseup)、移动中（onmousemove)

   键盘事件：

   按键按下：onkeydown、onkeypress

   onkeydown、onkeypress二者没有本质的区别，

   onkeydown：键盘上的有两个键按下不会触发改事件

   onkeypress(很少用）：功能键不会触发（F1 - F12）

   按键抬起：onkeyup

```
<input type="text" id="ipt">
<script>
var oIpt =
 document.getElementById("ipt");
oIpt.onkeydown = function() {
  console.log('1-onkeydown触发了');
};
oIpt.onkeypress = function() {
  console.log('2-onkeypress触发了');
};
oIpt.onkeyup = function() {
  console.log('3-onkeyup触发了');
};
</script>
```

窗口事件:

onload: 窗口加载事件

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <script>
    window.onload = function() {

 console.log(document.getElementById("box"));
    };
  </script>
</head>
<body>

  <div id="box">

  </div>

</body>
</html>
```

onresize: 窗口尺寸大小改变的时候触发

onscroll：窗口滚动的时候触发

```
<script>
 window.onload = function() {

console.log(document.getElementById
("box"));
 };
  // window.onresize = function() {
  //  console.log("窗口尺寸变了");
  // };

  window.onscroll = function() {
    console.log("页面滚动了");
  };
</script>
```

表单事件：

onchange： 当表单元素值发生变化触发

```
<!-- <input type="radio" id="ipt">
<input type="checkbox" id="ipt"> --
>
<input type="text" id="ipt" >
<script>
```

```
  var oIpt =
 document.getElementById("ipt");

  /*
      如果想每次输出值，就触发就用键盘事
件
    */
  // 类似onblur，失去焦点触发
  oIpt.onchange = function() {
    console.log("onchange被触发了");
  };
</script>
```

onchange的触发时机：

1. 如果元素是文本框、密码框、文本域，失焦的时候触发，此时跟blur一样
2. 如果元素是下拉列表、单选框、复选框的时候，当值发生变化的时候就会触发

onfoucs：聚焦

onblur：失焦

```
<input type="text"
       placeholder="请输入用户名:"
       id="ipt">
<script>
  var oIpt =
document.getElementById("ipt");
```

```
  // 聚焦
  oIpt.onfocus = function() {
    console.log("我获得焦点了");
  };
  // 失焦
  oIpt.onblur = function() {
    console.log("我失去焦点了");
  };
</script>
```

onsubmit(form专属的事件)：提交

```
<form id="form">
  <input type="text"
         placeholder="手机号/邮箱/账号"><br/>
  <input type="submit" value="注册">
</form>
<script>
  var oForm = document.getElementById("form");
  // form的提交事件  只要点击submit按钮就出触发form的submit事件, submit触发了之后默认会刷新页面
  oForm.onsubmit = function() {
    alert(1);
  };
  </script>
```