**懒加载(滚动加载)**

**一、什么是懒加载？**

当页面图片太多时，加载速度就会很慢。图片懒加载的原理就是将页面内所有需要加载的图片全部换成一张大小为1\*1px图片的路径（这样就只需请求一次），只有当图片出现在浏览器的可视区域内时，才设置图片正真的路径，让图片显示出来。这就是图片懒加载。

思路：

将页面里所有img属性src属性用data-src代替，当页面滚动直至此图片出现在可视区域时，用js取到该图片的data-src的值赋给src。

**二、为什要使用这个技术？**

　　比如一个页面中有很多图片，如淘宝、京东首页等等，如果一上来就发送这么多请求，页面加载就会很漫长，如果js文件都放在了文档的底部，恰巧页面的头部又依赖这个js文件，那就不好办了。更为要命的是：一上来就发送百八十个请求，服务器可能就吃不消了（又不是只有一两个人在访问这个页面）。

因此优点就很明显了：不仅可以减轻服务器的压力，而且可以让加载好的页面更快地呈现在用户面前（用户体验好）。

offsetParent

offsetParent属性返回一个对象的引用，这个对象是距离调用offsetParent的元素最近的（在包含层次中最靠近的），并且是已进行过CSS定位的容器元素。 如果这个容器元素未进行CSS定位, 则offsetParent属性的取值为body。 当容器元素的style.display 被设置为 "none"时（译注：IE和Opera除外），offsetParent属性 返回 null。

**位运算符**

在js中所有数值都以IEEE-754 64位格式储存，但位操作符会将64位储存格式转换成32位的整数来操作。

按位非操作符NOT ~ num1 返回数值的反码。

var num1=25;NaN 会被当做0运算

var num2=-num1-1;

1.类型转换调用Number，非数值类型的需要转化为数值类型 。

2.特殊值处理，NaN 和 ∞ 都被转化为 0。

3.所有的位操作都会先将操作数转化为 32 位有符号整数，如果是浮点数，会损失小数点后面的精度

执行完 ToInt32 之后，将得到的 32 位有符号整数进行按位取反，并将结果返回。

需要注意的是，所有的位操作都会先将操作数转化为 32 位有符号整数。

因为第一次执行 ~ 时已经将操作数转化为 32 位有符号整数，第二次执行 ~ 时实际只是将按位取反的结果再次按位取反，相当于取消掉 ~ 处理过程。