各浏览器下Hack的写法

1. Firefox

@-moz-document url-prefix() {

.selector {

property: value;

}

}

1. Webkit枘核浏览器(chrome and safari)

@media screen and (-webkit-min-device-pixel-ratio:0) {

Selector { property: value; }

}

1. Opera浏览器

@media all and (min-width:0) {

Selector {property: value;}

}

1. IE9浏览器

:root Selector {property: value\9;}

1. IE9以及IE9以下版本

Selector {property:value\9;}

1. IE8浏览器

@media \0screen{

Selector {property: value;}

}

1. IE8以及IE8以上的版本

Selector {property: value\0;}

1. IE7浏览器

\*+html Selector{property:value;}

或

\*:first-child+html Selector {property:value;}

1. IE7及IE7以下版本浏览器

Selector {\*property: value;}

1. IE6浏览器

Selector {\_property*/\*\*/:/\*\*/*value;}

或者：

Selector {\_property: value;}

或者：

\*html Selector {property: value;}

1. 清除input边框

input{outline: none; border: none; /\*IE6\*/ background: none;}

selector{background-color:#fff;}

1. CSS3的兼容性写法

border-radius

-webkit-border-radius:10px;

-moz-border-radius:10px;

IE6/7/8:引入ie-css3兼容文件ie-css3.htc,不支持除了黑色(#000)以外的其他颜色

box-shadow

-webkit-box-shadow:10px 10px 5px #888;

-moz-box-shadow:10px 10px 5px #888;

IE6/7/8:

方法一:

filter:progid:dXImageTransform.Microsoft.DropShadow(color=#888888,offX=10,offY=10,positives=true);

缺点：阴影不能边缘模糊

方法二：

filter:progid:DXImageTransform.Microsoft.Shadow(color='#0099ff', Direction='120', Strength='10');

类似于投影效果，缺点：阴影不能边缘模糊

方法三(推荐)、引入ie-css3兼容文件behavior:url(ie-css3.htc)

缺点：不支持除了黑色(#000)以外的其他颜色

box-sizing

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

transform-style

-webkit-transform-style: preserve-3d;

-moz-transform-style: preserve-3d;

transform-style: preserve-3d;

backface-visibility

-webkit-backface-visibility: hidden;

-moz-backface-visibility: hidden;

backface-visibility: hidden;

border-image

-webkit-border-image:url("http://www.w3school.com.cn/i/border.png") 30 30 stretch;

-moz-border-image:url("http://www.w3school.com.cn/i/border.png") 30 30 stretch;

-o-border-image:url("http://www.w3school.com.cn/i/border.png") 30 30 stretch;

background-size

-webkit-background-size:10px 10px 5px #888;

-moz-background-size:10px 10px 5px #888;

IE8：

方法一(推荐)、引入backgroundsize.min.htc兼容文件

.size {

width: 400px;

height: 400px;

margin: 20px auto 0;

background: green url(img/1.jpg) no-repeat scroll center 0;

background-size: cover;

-ms-behavior: url(js/backgroundsize.min.htc);

}

方法二、针对IE8的hack

<!--[if IE 8]>

<style type="text/css">

.size{

-ms-filter: "progid:DXImageTransform.Microsoft.AlphaImageLoader(src='img/1.jpg', sizingMethod='scale')";

}

//该函数只能模仿cover值,IE8下效果与IE9/10稍有不同,图片占满背景大小且不超出

</style>

<![endif]-->

background-origin

-moz-background-origin:padding | border;

background-clip

-moz-background-clip:padding | border;

text-shadow

-webkit-text-shadow:1px 1px 1px #000;

-moz-text-shadow:1px 1px 1px #000;

IE6/7/8:

方法一、引入ie-css3兼容文件behavior:url(ie-css3.htc);

方法二、 filter:shadow(color=blue,direction=225),direction表示投影角度

transform

.transform{

-webkit-transform: x,y;

-moz-transform: x,y;

-ms-transform: x,y;

-o-transform: x,y;

transform: x,y;

}

IE8及以下：用IE滤镜

{

filter:fliph;//水平翻转相当于transform:rotateY(180deg)

filter:flipv;//垂直翻转相当于transform:rotateX(180deg)

}

transition

p {

-webkit-transition: all .5s ease-in-out 1s;

-moz-transition: all .5s ease-in-out 1s;

-o-transition: all .5s ease-in-out 1s;

transition: all .5s ease-in-out 1s;

}

animation

-webkit-

-moz-

@keyframes

-webkit-

-moz-

linear-gradient radial-gradient

-webkit-

-moz-

IE9及以下可使用 IE 滤镜处理:

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#000000',endColorstr='#ffffff');

rgba(r,g,b,a)

IE6/7/8不支持使用 rgba 模式实现透明度，可使用 IE 滤镜处理

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr=#7fff0000,endColorstr=#7fff0000);

使用后我们又发现了个小问题，IE9下颜色更深一点，仔细研究发现在IE9下rgba颜色和IE滤镜都能被识别，所以被覆盖了颜色就更深一点

解决办法：

.demo{filter:progid:DXImageTransform.Microsoft.gradient(startColorstr=#7fff0000,endColorstr=#7fff0000);}

:root .demo{filter:none;background-color:rgba(255,0,0,.2);}//IE8及以下不识别

flex

-webkit- 或者 -webkit-box-flex

-moz-box-flex:1;

-ms-flex:1;

flex-flow

box-orient:horizontal + box-direction:normal 达到新版本 flex-direction:row 的效果；

可以通过box-orient:horizontal + box-direction:reverse 达到新版本 flex-direction:row-reverse 的效果；

可以通过box-orient:vertical + box-direction:normal 达到新版本 flex-direction:column 的效果；

可以通过box-orient:horizontal + box-direction:reverse 达到新版本 flex-direction:column-reverse 的效果；

align-content

-webkit-

align-items

-webkit-

align-self

-webkit-

order

-webkit-

JS

bind方法向下兼容代码

if (!Function.prototype.bind) {

Function.prototype.bind=function(oThis){

if (typeof this !== 'function') {

throw new TypeError('What is trying to be bound is not callable');

}

var aArgs=Array.prototype.slice.call(arguments,1),

fToBind=this,

fNOP=function () {

},

fBound =function () {

return fToBind.apply(this instanceof fNOP?this:oThis,aArgs.concat(Array.prototype.slice.call(arguments)));

};

fNOP.prototype=this.prototype;

fBound.prototype=new fNOP();

return fBound;

};

}

Object.create向下兼容方法

if (!Object.create) {

Object.create = function(proto) {

function F() {}

F.prototype = proto;

return new F();

};

}

模拟重载方法

//依据参数列表来严格地检查一个变量列表的类型

function strict(types, args) {

//确保参数的数目和类型核匹配

if (types.length != args.length) {

//如果长度不匹配，则抛出异常

throw "Invalid number of arguments. Expected " + types.length + ", received " + args.length + " instead.";

}

//遍历每一个参数，检查基类型

for (var i = 0; i < args.length; i++) {

//如JavaScript某一项类型不匹配，则抛出异常

if (args[i].constructor != types[i]) {

throw "Invalid argument type. Expected " + types[i].name + ", received " + args[i].constructor.name + " instead.";

}

}

}

//上述方法的使用

function doFunction(id, name) {

//检测参数个数和类型

strict([Number, String], arguments);

}

\*得到window尺寸（不包括工具栏和滚动条）

var w = window.innerWidth || document.documentElement.clientWidth || document.body.clientWidth;

var h = window.innerHeight || document.documentElement.clientHeight || document.body.clientHeight;

在Web页面制作中尽量不要使用CSS Hack来处理兼容问题。因为任何浏览器下出现渲染不一致都极有可能是我们自己的结构或样式不符合W3C的某些要求，或者说违背了浏览器的某些规则而造成的，所以我们应该尽量通过结构或CSS的修改来达到各浏览器渲染一致效果，除非到了万不得已情况下，才考虑CSS的Hack。