

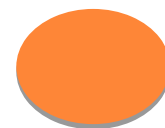


第二次课：JS+CSS+SVG 互操作

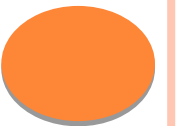
李春芳

上次回顾

- W3C
- HTML : 基本元素
- CSS : 样式, 格式
- SVG 绘制 : 矢量图
- JS : 交互脚本
- D3.JS 是一种 JS 对可视化的封装 API
- 如何互操作呢 ?

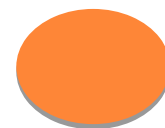


HTML&CSS



JAVASCRIPT 基本语法

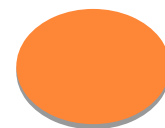
- 区分大小写
- 注释：`//` `/* */`
- 分号结尾：`;`
- `{ }` 定义函数
- 定义变量：`var`
- 弱类型变量：可以是数值、布尔型、字符串
- 数据类型：`undefined`(未定义的变量)、`null` (空对象)、`boolean`、`number`、`string`、`object`
- 查看变量类型：`console(typeof a)`



NUMBER

- 十进制数 : `var num=120;`
- 八进制数 : `var num8=020;`
- 十六进制 : `var numf=0x1A;`
- 浮点数 : `f=3.1415;`
- 指数表示 : `num m=3e5;`

- 最大值 : `Number.MAX_VALUE`
- 最小值 : `Number.MIN_VALUE`
- NaN : Not a Number

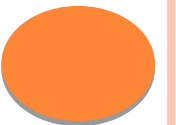


STRING

- `Var str="China";`
- `Console.log(str.length);`

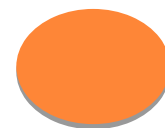
○ **Object : 属性和方法**

- `Var person=new Object();`
- `person.name="wang";`
- `person.age=10;`
- `person.growUp=function(){`
- `this.age+=1;`
- `}`



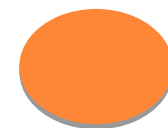
操作符

- 算术操作符：+、-、*、/、%、++、--
- 赋值操作符：=、+=、-=、*=、/=、%=
- 布尔操作符：!、&&、||
- 关系操作符：>、<、>=、<=、==（恒等）、===（全等）、!=（不等）、!==（不全等）



语句

- If-else
- While 和 do-while
- For 和 for-in
- Switch
- Break 、 continue 和 Label



函数

定义

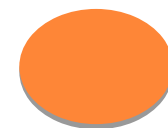
```
function add(num1,num2){  
    Return num1+num2  
}
```

调用

```
var a=add(3,5)
```

无名函数

```
var a=function(num1,num2){  
    return num1+num2;  
}
```



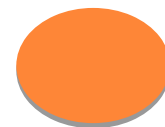
对象

```
var person={  
    name:"Rose",  
    Age:20,  
    growUp:function(){  
        this.age+=1;  
        }// 如果之后还有属性，这里要添加逗号  
}
```

```
function Person(name,age)={  
    this.name=name;  
    this.age=age;  
    This.growUp=function(){  
        this.age+=1;  
    }  
}
```

创建对象

```
var rose=new Person("rose",18);  
console.log(rose.name);  
console.log(rose["name"]);  
rose["growUp"]();
```



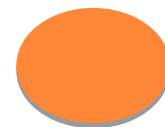
数组

- JS 的数组实际是一种对象，使用 `typeof` 可检测
- `var a=[1,2,3,4,5];`
- `console.log(typeof a);`

- 数组可以类型不同
- `var a=[1,"apple",false,4,5];`

- `var a=new Array(1,2,3,4);`
- `var b=new Array(20);`

- `var city=["beijing","上海","Guangzhou"];`
- `console(city);`
- `console(city.length);`





DOM : 文档对象模型 (DOCUMENT OBJECT MODEL)

DOM

- 文档对象模型 (Document Object Model)
- 结构化文档的一个接口
- 允许程序和脚本动态地访问和修改文档
- 针对 HTML 的：HTML DOM
- JS 与 HTML 的交互使用 DOM：对谁操作



站住 说你呢



课程表

HTML DOM

DOM 教程

DOM 节点

DOM 方法

DOM 属性

DOM 事件

DOM 修改

DOM 内容

DOM 元素

DOM 事件

DOM 导航

DOM 总结

DOM 实例

DOM 实例

DOM 参考手册

DOM 参考

选修课

HTML DOM 教程

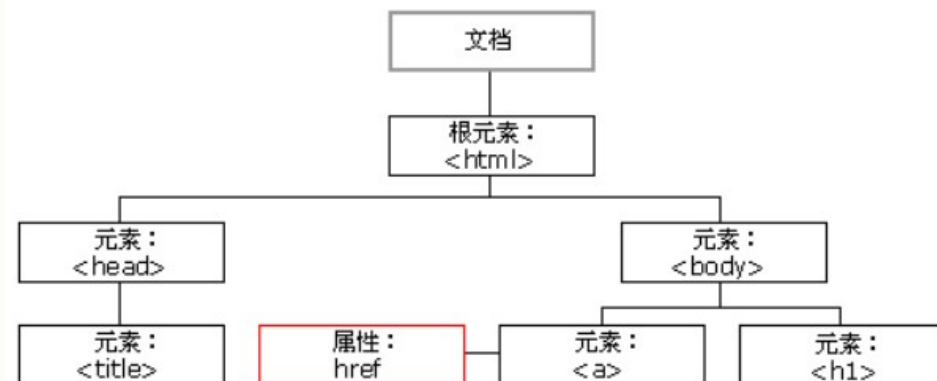
[← 上一节](#)[下一节 →](#)

HTML DOM 定义了访问和操作 HTML 文档的标准方法。

DOM 将 HTML 文档表达为树结构。

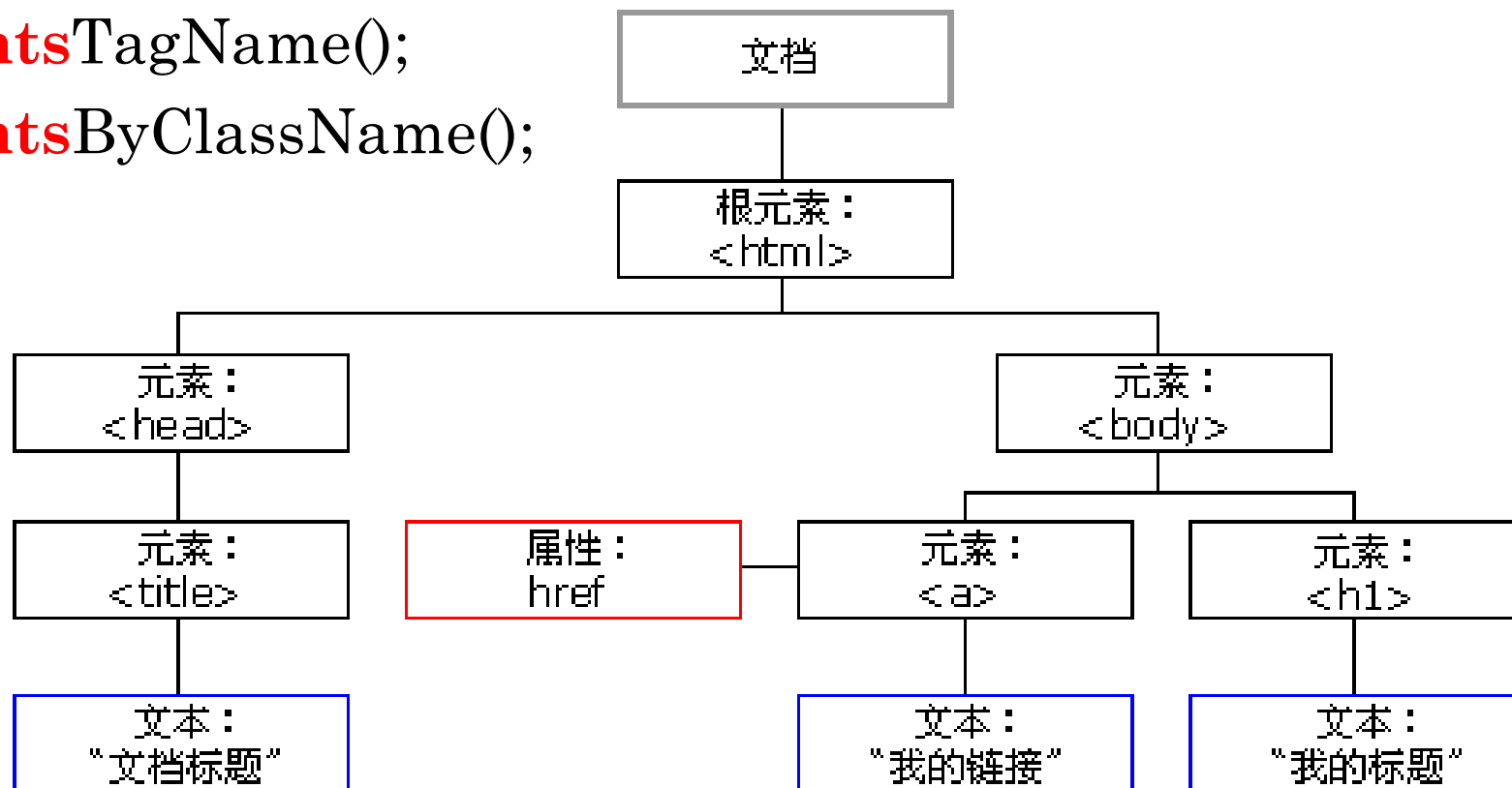
[开始学习 HTML DOM!](#)

HTML DOM 树



HTML DOM 树

- document.get**Element**ById();
- document.get**Elements**TagName();
- document.get**Elements**ByClassName();



例

<p> 中国传媒大学 </p>

<p> 计算机学院 </p>

<script>

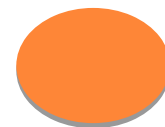
```
var para=document.getElementsByTagName("p");
```

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

```
    console.log(para[i].innerHTML);
```

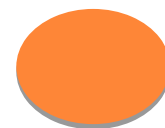
```
}
```

</script>



DOM 常用属性

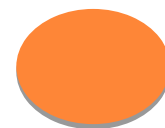
- innerHTML : 元素内部文本，包括 HTML 标签
- innerText : 元素内部文本，**不包括 HTML 标签**
- **outerHTML**
- **outerText**
- **nodeName**
- **parentName**
- **childNodes**
- **nextSibling**
- **previousSibling**
- **style**



例

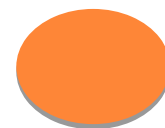
- `var body=document.getElementsByTagName("body");`
- `console.log(body);`
- `Body[0].style.backgroundColor="blue";`

- 注意：返回的是数组



添加节点

- appendChild()
- removeChild()
- var para=document.**createElement**("p")
- para.innerHTML=" 传媒大学" ;
- var body=document.getElementById("mybody");
- body.**appendChild**(para);



删除节点

- `<body id="mybody">`
- `<p id="mypara"> 中传 </p>`
- `<p> 计算机 </p>`
- `<script>`
 - `var para=document.getElementById("mypara");`
 - `Var body=document.getElementById("mybody");`
 - `body.removeChild(para);`
- `</script>`
- `</body>`



事件

- 与用户交互：鼠标单击、鼠标移入、页面加载
- `<p id="mypara"> 单击交互 </p>`

```
<script>
```

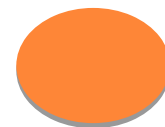
```
    var para=document.getElementById("mypara");
```

```
    para.onclick=function(){
```

```
        this.innerHTML="Thank you";
```

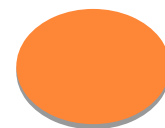
```
    }
```

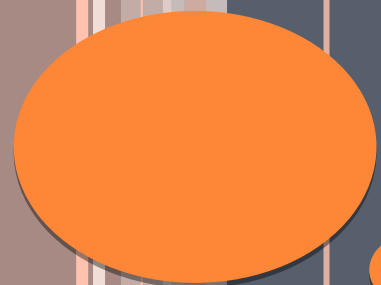
```
</script>
```



常用事件

- **onload** : 加载
- **onclick** : 单击
- ondblclick : 双击
- onkeydown : 键盘按键按下
- onkeyup : 键盘按键按下又松开
- onmousedown : 鼠标按下
- **onmousemove** : 鼠标移动
- onmouseout : 鼠标移开
- onmouseover : 鼠标移入
- onmouseup : 鼠标松开





SVG 图形

Scalable Vector Graphics

SVG 图形

○ 矩形

- `<rect x="20" y="20" width="200" height="300" />`
- `<rect x="20" y="20" rx="20" ry="30" width="200" height="300" style="fill:yellow;stroke:black;stroke-width:4;opacity:0.5" />`

○ 圆形 `<circle cx="150" cy="150" r="80" />`

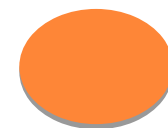
○ 椭圆 `<ellipse cx="350" cy="150" rx="110" ry="80" />`

○ 线段 `<line x1="20" y1="20" x2="300" y2="100" />`

○ 折线 & 多边形

- `<polyline points="100,20 20,90 60,160 140,160 180,90" />`
- `<polygon points="100,20 20,90 60,160 140,160 180,90" />`

○ 路径



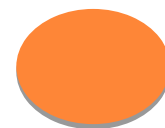


SVG 直方图

加载数据 & 全屏显示

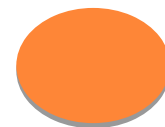
获取屏幕分辨率 & 浏览器可用宽度

- `<!DOCTYPE html>`
- `<html>`
- `<body>`
- `<script>`
- `document.write(" 可用宽度 : " + screen.availWidth+"
");`
- `document.write(" 可用宽度 : " + screen.availHeight);`
- `</script>`
- `</body>`
- `</html>`



获取屏幕分辨率 & 浏览器可用宽度

- `<p id="demo"></p>`
- `<script>`
- `var w=window.innerWidth`
- `|| document.documentElement.clientWidth`
- `|| document.body.clientWidth;`
- `var h=window.innerHeight`
- `|| document.documentElement.clientHeight`
- `|| document.body.clientHeight;`
- `x=document.getElementById("demo");`
- `x.innerHTML="浏览器的内部窗口宽度： "+ w + " ， 高度： "+ h + "。 "`
- `</script>`

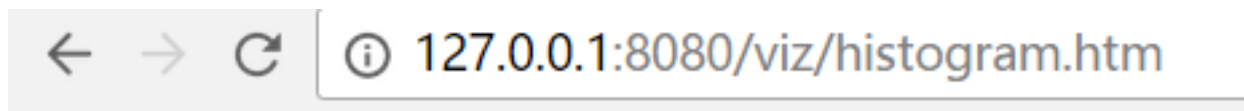


绘制一个矩形

- `<svg id="wcSvg" width="1520" height="721">`
- `<rect width="100" height="30"/>`
- `</svg>`

- 添加样式

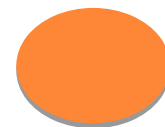
- `<rect width="100" height="30" style="fill:rgb(0,0,255);stroke-width:1;stroke:rgb(0,0,0)" / >`



可用宽度：1536;可用宽度：759



- `<rect x="20" y="50" rx="5" ry="5" width="100" height="400" style="fill:rgb(0,0,255);stroke-width:1;stroke:rgb(0,0,0)" />`

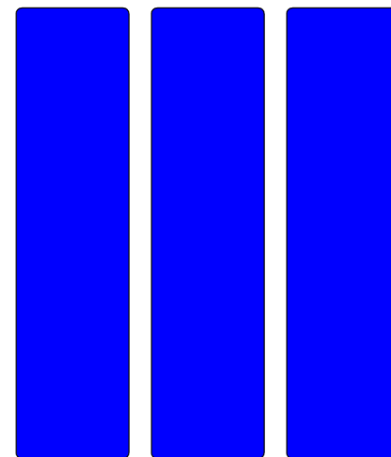


多个矩形

- `<svg id="wcSvg" width="1520" height="721">`
- `<rect x="20" y="50" rx="5" ry="5" width="100" height="400" style="fill:rgb(0,0,255);stroke-width:1;stroke:rgb(0,0,0)" />`
- `<rect x="140" y="50" rx="5" ry="5" width="100" height="400" style="fill:rgb(0,0,255);stroke-width:1;stroke:rgb(0,0,0)" />`
- `<rect x="260" y="50" rx="5" ry="5" width="100" height="400" style="fill:rgb(0,0,255);stroke-width:1;stroke:rgb(0,0,0)" />`
- `</svg>`

← → ↻ ⓘ 127.0.0.1:8080/viz/histogram.htm

可用宽度 : 1536;可用高度 : 759



在 JS 中动态添加 SVG 的文本

- `<script>`
- `var w=window.innerWidth || document.documentElement.clientWidth || document.body.clientWidth;`
- `var h=window.innerHeight || document.documentElement.clientHeight || document.body.clientHeight;`
- `</script>`
- `<svg id="histSvg" width="1520" height="721" version="1.1" xmlns="http://www.w3.org/2000/svg" ></svg>`
- `<script>`
- `var svgText=document.createElement("text");`
- `var hist=document.getElementById("histSvg");`
- `hist.appendChild(svgText);`
- `svgText.outerHTML="<text x=0 y=20 style='font-family: 微软雅黑 ;fill:rgb(0,0,255);font-size:20'>"+w+" </text>";`
- `</script>`



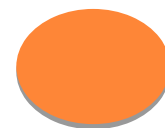
在 JS 中动态添加 SVG 的矩形

- `<script>`
- `var hist=document.getElementById("histSvg");`
- `//SVG 添加文本`
- `var svgText=document.createElement("text");`
- `hist.appendChild(svgText);`
- `svgText.outerHTML="<text x=0 y=20 style='font-family: 微软雅黑 ;fill:rgb(0,0,255);font-size:20'> "+w+" </text>";`
- `//SVG 添加图形`
- `var myRect=document.createElement("rect");`
- `hist.appendChild(myRect);`
- `myRect.outerHTML="<rect x=20 y=50 width=100 height=300 style='fill:blue'/>";`
- `</script>`



在 JS 中动态添加 SVG 的矩形

- `<script>`
- `var hist=document.getElementById("histSvg");`
- `//SVG 添加文本`
- `var svgText=document.createElement("text");`
- `hist.appendChild(svgText);`
- `svgText.outerHTML="<text x=0 y=20 style='font-family: 微软雅黑 ;fill:rgb(0,0,255);font-size:20'> "+w+"</text>";`
- `//SVG 添加直方图`
- `var myRect=new Array();`
- `for(var i=0;i<6;i++){`
- `console.log(i);`
- `myRect[i]=document.createElement("rect");`
- `hist.appendChild(myRect[i]);`
- `myRect[i].outerHTML="<rect x=20 y=50 width=100 height=300 style='fill:blue'/>";`
- `}`
- `</script>`

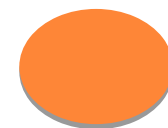


1536

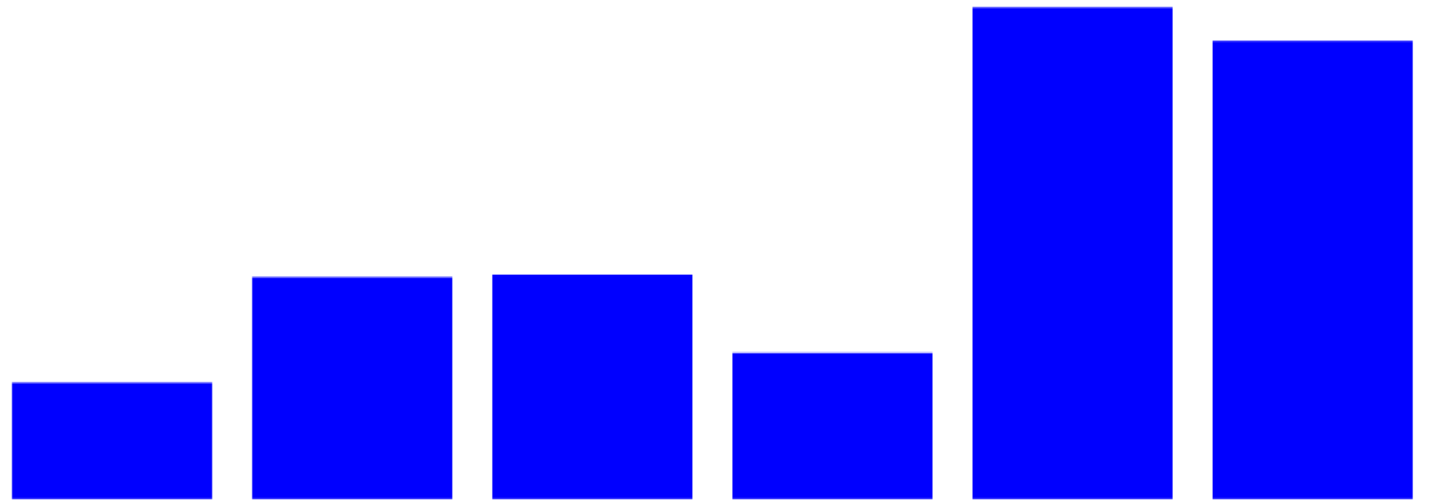


一组随机高度的矩形

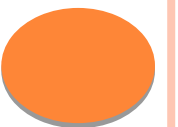
- //SVG 添加直方图
- `var myRect=new Array();`
- `for(var i=0;i<6;i++){`
- `console.log(i);`
- `myRect[i]=document.createElement("rect");`
- `hist.appendChild(myRect[i]);`
- `//myRect[i].outerHTML="<rect x=20 y=50 width=100 height=300`
`style='fill:blue'/>";`
- `//myRect[i].outerHTML="<rect x="+i*120+" y=50 width=100 height=300`
`style='fill:blue'/>";`
- **`var rand=Math.random()*300;`**
- **`myRect[i].outerHTML="<rect x="+i*120+" y=50 width=100`
`height="+rand+" style='fill:blue'/>";`**
- `}`



1536

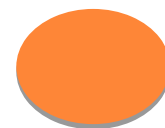


- `myRect[i].outerHTML=<rect x="+ (i*120) +" y="+ (300-rand) +" width=100 height="+ rand +" style='fill:blue'/>;`



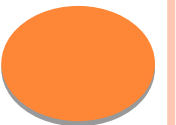
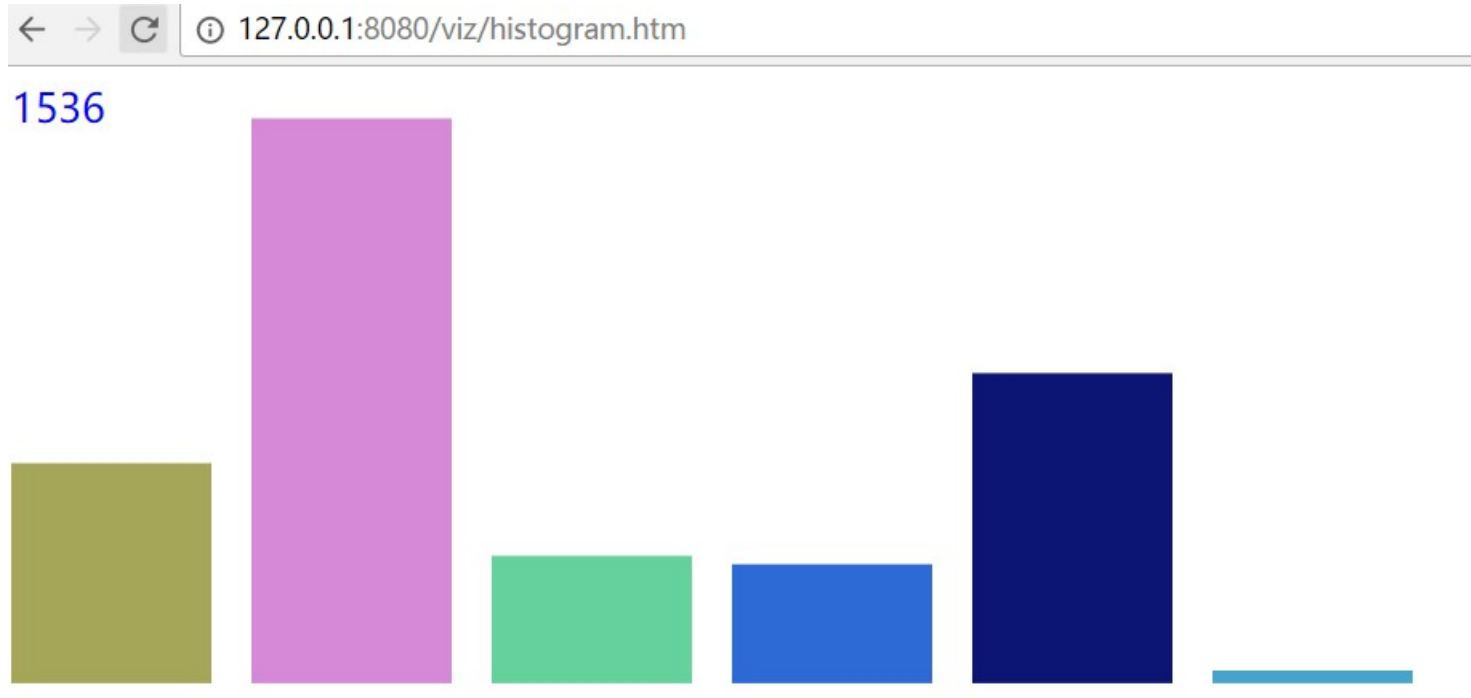
随机色彩

- `var hiscolor=Math.round(Math.random()*255);`
- `myRect[i].outerHTML="<rect x="+i*120+" y="+300-
rand)+" width=100 height="+rand+" style='fill:rgb(0,0,"+
(hiscolor)+")'/>";`



全屏显示

- `myRect[i].outerHTML=<rect x="+ (i*120)+" y="+ (300-rand)+" width="+ (Math.round(w/myRect.length))+" height="+ rand+" style='fill:rgb("+ (hiscolorR)+" ,"+ (hiscolorG)+" ,"+ (hiscolorB)+")'/>`;



全屏显示

← → ↻ ⓘ 127.0.0.1:8080/viz/histogram.htm



1536

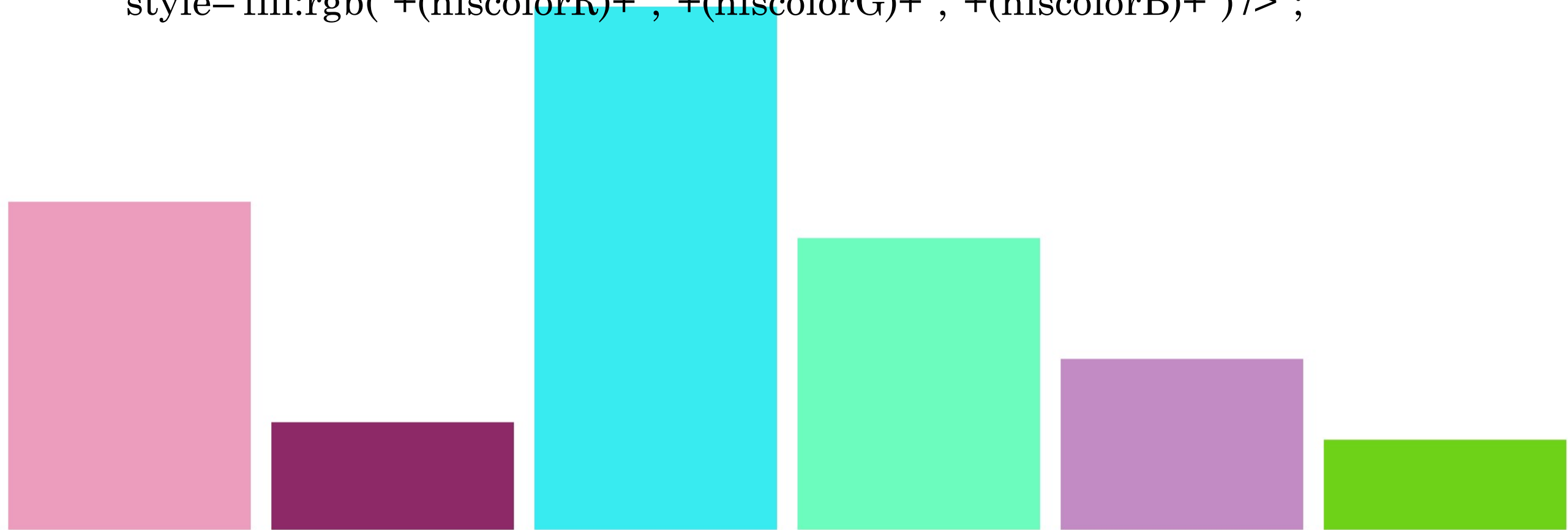
- `var histWidth=Math.round(w/myRect.length);`
- `myRect[i].outerHTML="<rect x="+(i*histWidth)+" y="+
(histHeight-rand)+" width="+(histWidth)+" height="+rand+"
style='fill:rgb("+hiscolorR+", "+hiscolorG+", "+hiscolorB+")'/>"`



空隙

- `myRect[i].outerHTML=<rect x="(i*histWidth)+" y="(histHeight-rand)+" width="(histWidth-20)+" height="+rand+" style='fill:rgb("+hiscolorR+", "+hiscolorG+", "+hiscolorB+")'/>;`

1536



添加图注数字

127.0.0.1:8080/viz/histogram.htm

1536

- var histText=new Array(6);

- histText[i]=document.createElement("text");

- hist.appendChild(histText[i]);

- histText[i].outerHTML="<text x="+ (i*histWidth)+" y="+ (histHeight-rand)+" style='font-family: 微软雅黑 ;fill:rgb(0,0,255);font-size:20'> "+rand+" </text>";

100

272

289

174

550

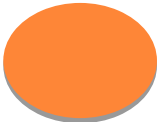
475



加载各种数据

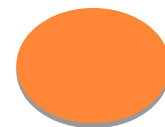
- 2016 年中国人口最多的省份排名、人口 GDP 及人均排名

序号	省区市	人口数量	人口排名	GDP(亿 元)	人均 G (元)	人均排名
1	河南省	9613 万人	5	14234	15056	16
2	山东省	9082 万人	2	25326	27148	7
3	四川省	8673 万人	9	9657	11708	25
4	广东省	7859 万人	1	29863	32142	6
5	江苏省	7381 万人	3	24738	32985	5
6	河北省	6735 万人	6	13387	19363	11
7	湖南省	6629 万人	13	8366	13123	20
8	安徽省	6338 万人	15	6906	11180	28
9	湖北省	5988 万人	11	8451	14733	17
10	广西省	4822 万人	18	5386	11417	27
11	浙江省	4647 万人	4	17633	35730	4
12	云南省	4333 万人	23	4260	9459	30
13	江西省	4222 万人	19	5323	12204	24
14	辽宁省	4203 万人	8	10418	24645	9
15	贵州省	3837 万人	26	2543	6742	31
16	黑龙江省	3813 万人	14	7081	18463	12
17	陕西省	3674 万人	21	4806	12843	21
18	福建省	3466 万人	12	8440	23663	10
19	山西省	3294 万人	17	5465	16143	15
20	重庆市	3107 万人	24	3938	14011	18
21	吉林省	2699 万人	22	4693	17211	13
22	甘肃省	2593 万人	27	2494	9527	29
23	内蒙古	2379 万人	16	6140	25558	8
24	新疆区	1905 万人	25	3305	16164	14
25	上海市	1625 万人	7	11658	65473	1
26	北京市	1423 万人	10	8879	57431	2
27	天津市	1007 万人	20	5014	7972	3



加载数据

- `Array(15056,27148,11708,32142,32985,19363,13123,11180,14733,11417,35730,9459,12204,24645,6742,18463,12843,23663,16143,14011,17211,9527,25558,16164,65473,57431,7972,13361,12695,12809,11567);`
- `var gdpName=new Array(" 河南省 "," 山东省 "," 四川省 "," 广东省 "," 江苏省 "," 河北省 "," 湖南省 "," 安徽省 "," 湖北省 "," 广西
省 "," 浙江省 "," 云南省 "," 江西省 "," 辽宁省 "," 贵州省 "," 黑龙江 "," 陕
西省 "," 福建省 "," 山西省 "," 重庆市 "," 吉林省 "," 甘肃省 "," 内蒙
古 "," 新疆区 "," 上海市 "," 北京市 "," 天津市 "," 海南省 "," 宁夏区 "," 青
海省 "," 西藏区 ");`
- <http://www.chyxx.com/industry/201701/486625.html>

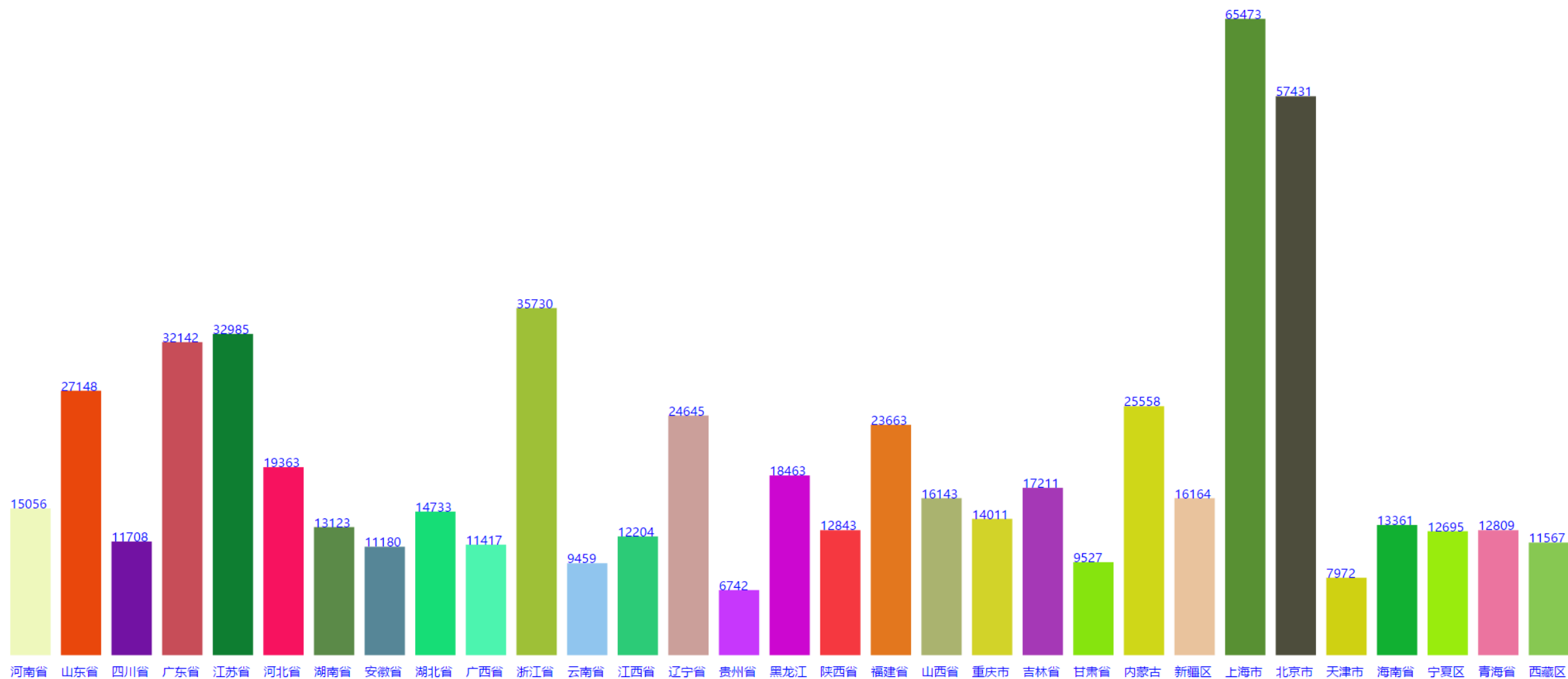


2016 人均 GDP

← → ↺ ⓘ 127.0.0.1:8080/viz/gdp2016.htm

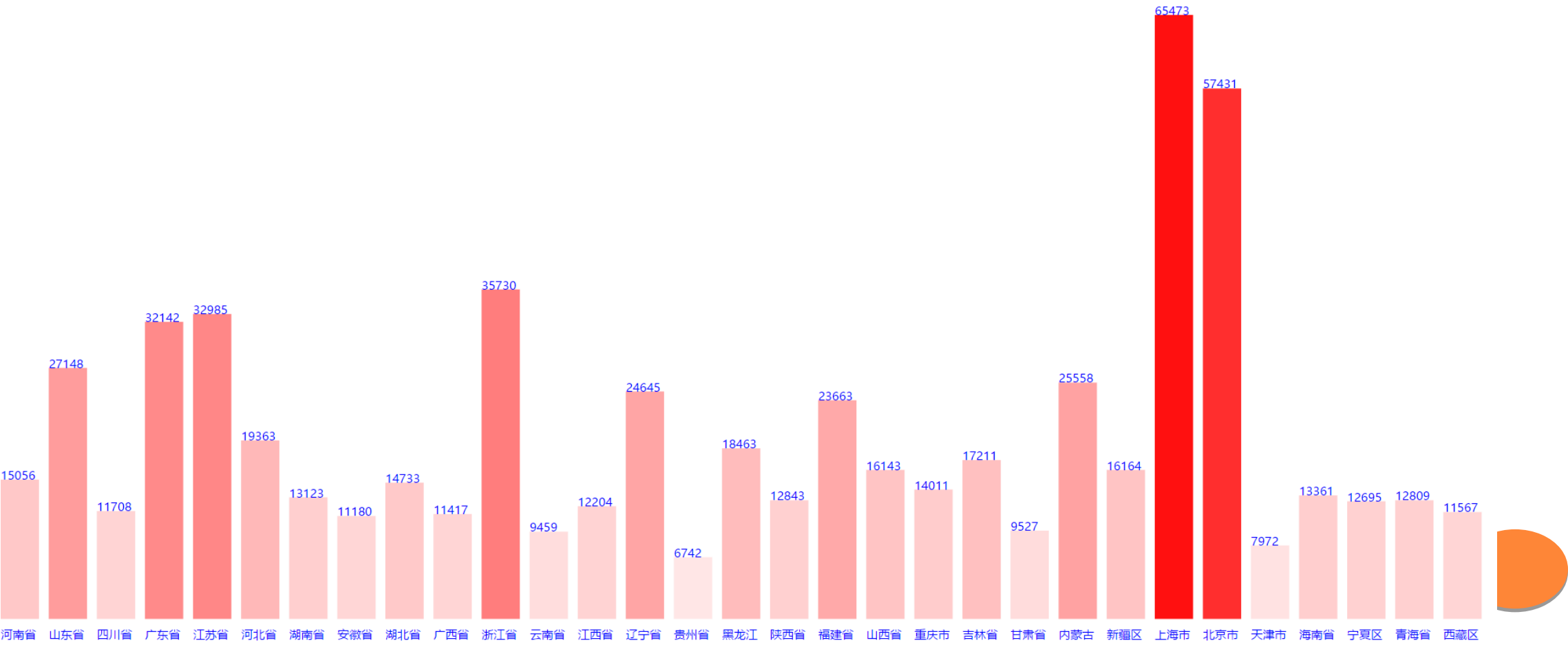


1536



用色彩编码 GDP 的值

1536



JSBIN.COM

○ 左边是代码，右边是结果

The screenshot shows the JS Bin website interface. The browser address bar displays `jsbin.com/?html,output`. The top navigation bar includes links for **JS Bin features**, **Pro features**, **Blog**, **Donate to JS Bin**, and a quote by Steve Jobs. Below the navigation bar is a toolbar with tabs for **HTML**, **CSS**, **JavaScript**, **Console**, and **Output**. The **JavaScript** tab is active, showing the following code:

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

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

w=w*0.98;
h=h*0.9;
svg.setAttribute("width",w);
svg.setAttribute("height",h);

rw=w/10;

var rect=new Array();

for(var i=0;i<10;i++){
  rect[i]=document.createElement('rect');
  svg.appendChild(rect[i]);
  var rh=Math.random()*255;
  rect[i].innerHTML="<rect x="+(rw*i);
}
</script>
</body>
```

The **Output** tab shows a bar chart with 10 bars of varying heights and colors (red and dark red). The bars are arranged in a sequence that peaks in the middle. The **Run with JS** button is visible, and the **Auto-run JS** checkbox is checked.