

《Web前端开发》理论考试

初级试卷分析

马庆槐 2021年3月



www.miiteec.org.cn

目录

1. 总体介绍
2. 典型试题分析

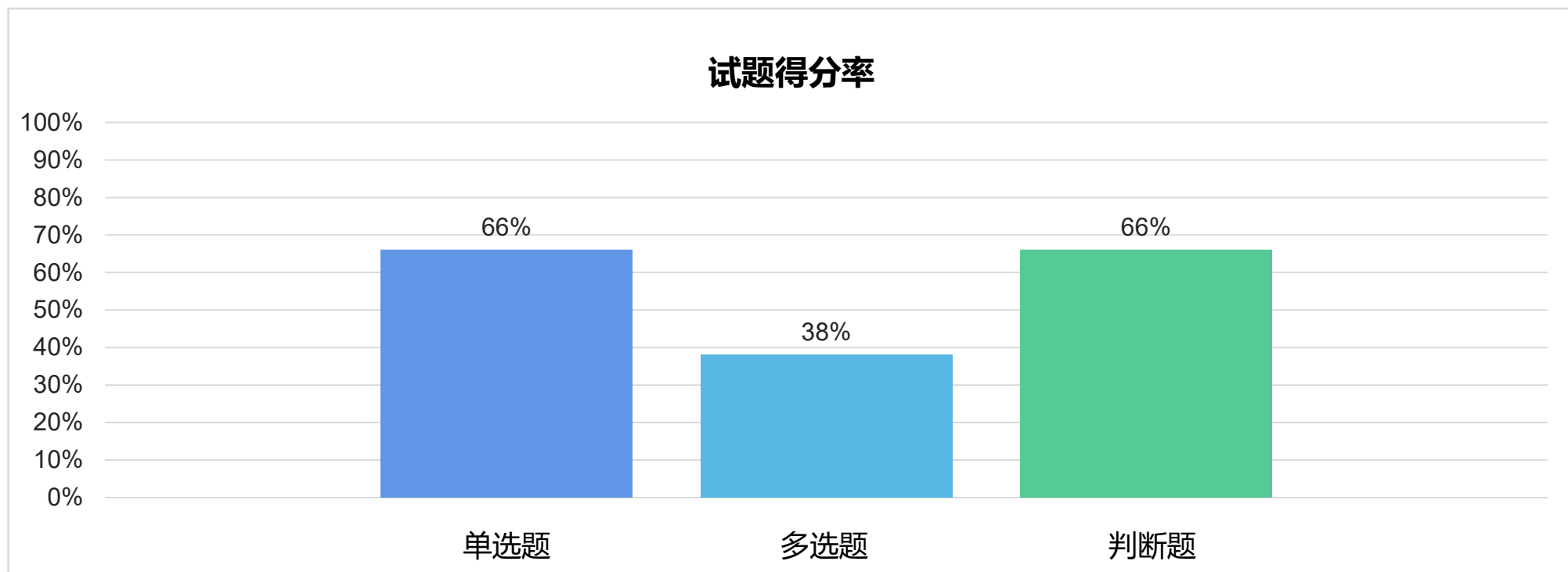


01

总体介绍

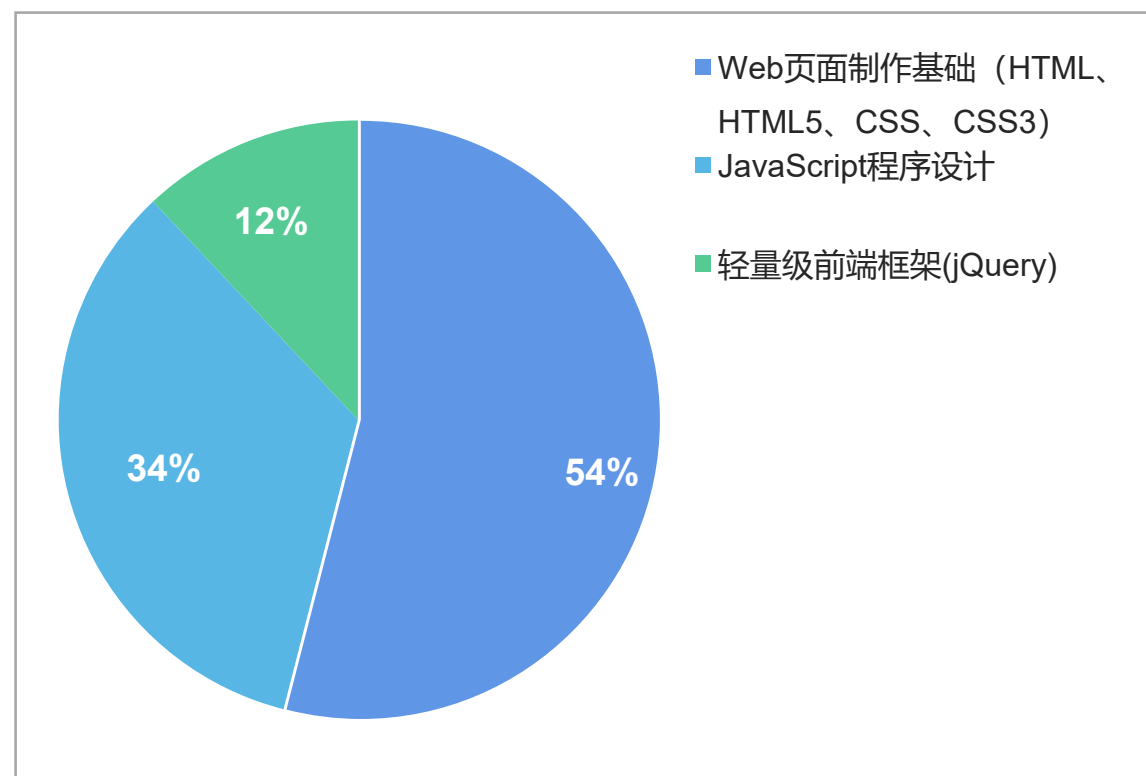
>>> 试卷结构

题型	试题数	得分率
单选题	30	66%
多选题	15	38%
判断题	5	66%



>>> 考试范围：分值分布

知识模块	试题数	占比
Web页面制作基础 (HTML、HTML5、CSS、CSS3)	27	54%
JavaScript程序设计	17	34%
轻量级前端框架(jQuery)	6	12%



>>> 考试范围：考核点分布

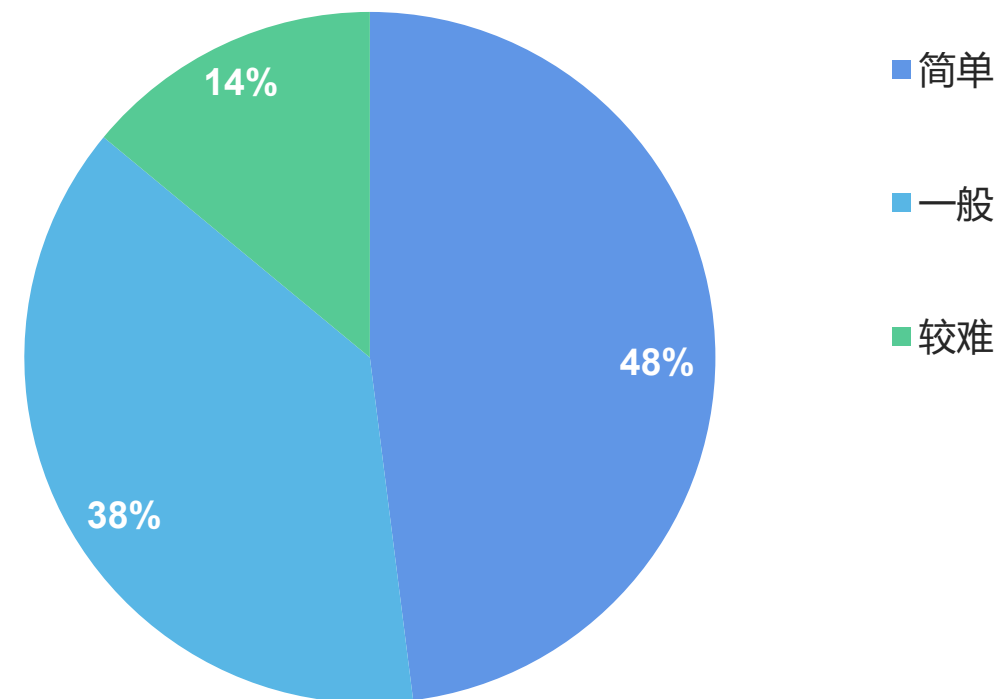


知识模块	考核点(职业技能要求)
Web页面制作基础 (HTML、HTML5、CSS、CSS3)	(1) HTML: 超文本标记语言、HTML基本结构、HTML 属性、列表、表格、表单、内联框架、块级元素、超链接、文件路径、HTML注释
	(2) HTML5: HTML5语义化标签、HTML5标签新特新、HTML5废弃元素、HTML5兼容性
	(3) CSS: 引用、命名规范、CSS语法、选择器、优先级、颜色、单位、字体、文本、背景、区块、边框、盒模型、定位、浮动
	(4) CSS3: 盒阴影、背景、边框新特性、盒模型、弹性布局
JavaScript程序设计	类型语言、变量、语句、数据类型、运算符、流程控制、字符串、函数、数组、内置对象(Date对象)、BOM、DOM、事件、console 对象
轻量级前端框架(jQuery)	基础语法、选择器、DOM操作、动画

难易度分析

难易度	题型	题目数	占比	合计
简单	单选题	15	30%	48%
	多选题	6	12%	
	判断题	3	6%	
一般	单选题	11	22%	38%
	多选题	6	12%	
	判断题	2	4%	
较难	单选题	4	8%	14%
	多选题	3	6%	
	判断题	0	0%	

难易度占比





02

典型试题分析



单选题

01. 盒模型(边框)



单选题

02. 弹性布局



多选题

03. DOM



多选题

04. JavaScript函数



多选题

05. jQuery选择器



单选

设置边框样式的属性是()。

- A、 box-sizing
- B、 box-shadow
- C、 border-radius
- D、 border

考核知识和技能：

- ✓ 盒模型
- ✓ 盒模型内容宽高计算
- ✓ 盒模型边框属性
- ✓ CSS3新特性（圆角边框、盒阴影）

解析：

1、盒模型

- (1) content(内容)
- (2) padding内边距
- (3) border边框
- (4) margin外边距

2、盒模型边框属性 (border)

格式：border: border-width border-style border-color ;



>>> 单选题：盒模型

3、盒模型内容宽高计算

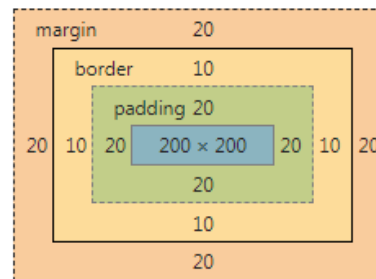
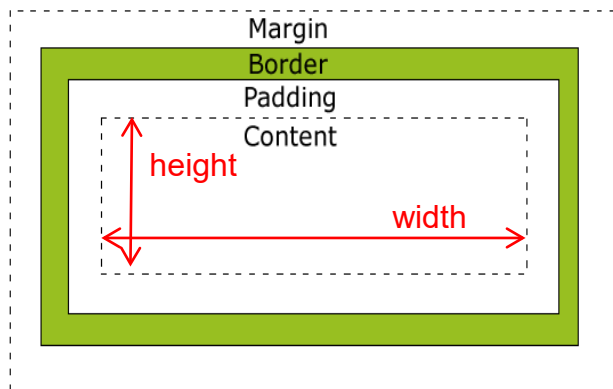
box-sizing 属性：

(1) **content-box**：默认值

width = 内容的宽度

height = 内容的高度

content-box



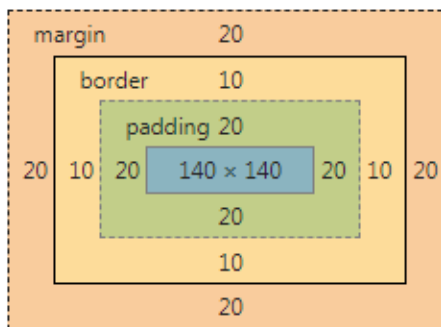
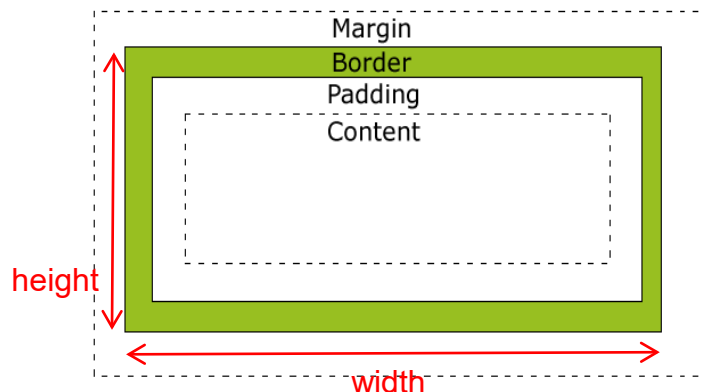
```
div{  
  width: 200px;  
  height: 200px;  
  border: 10px solid yellowgreen;  
  padding: 20px;  
  margin: 20px;  
  box-sizing: content-box;  
}
```

(2) **border-box**：

width = border*2 + padding*2 + 内容的宽度

height = border*2 + padding*2 + 内容的高度

border-box



```
div{  
  width: 200px;  
  height: 200px;  
  border: 10px solid yellowgreen;  
  padding: 20px;  
  margin: 20px;  
  box-sizing: border-box;  
}
```

4、CSS3新特性

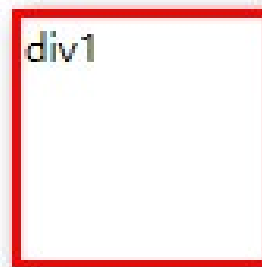
(1) 盒阴影 (box-shadow属性)

格式：

`box-shadow: h-shadow v-shadow blur spread color inset, h-shadow v-shadow blur spread color inset;`

```
<div class="div1">div1</div><br/>
<div class="div2">div2</div>
```

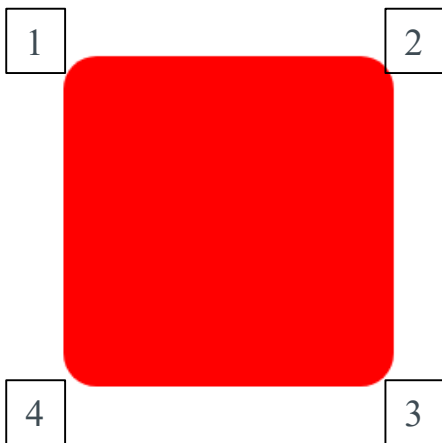
```
div{
    width: 100px;
    height: 100px;
    border: 5px solid red;
}
.div1{ box-shadow: 5px 5px 10px black;}
.div2{ box-shadow: 5px 5px 10px black,5px 5px 10px black inset;}
```



(2) 圆角边框 (border-radius属性)：设置四个方向的圆角边框border-*-radius

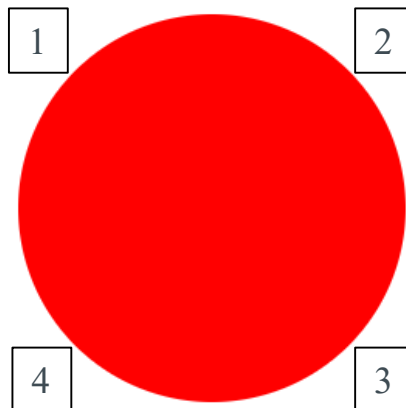
```
<body>
  <div id="div1"></div>
</body>
```

```
<style>
#div1{
  width: 200px;
  height: 200px;
  background-color: red;
  border-radius: 20px 20px 20px 20px;
}
</style>
```



```
<body>
  <div id="div2"></div>
</body>
```

```
<style>
#div2{
  width: 200px;
  height: 200px;
  background-color: red;
  border-radius: 100px;
}
</style>
```



单选题

关于flex说法错误的是()。

- A、设置flex布局以后，子元素的float和clear等样式全部失效
- B、任何一个容器都可以使用flex弹性布局
- C、设置flex:1无意义
- D、flex是弹性布局

考核知识和技能：

- ✓ 弹性布局
- ✓ flex属性
- ✓ float、clear

>>> 单选题:弹性布局

解析:

1、Flex(Flexible Box)弹性布局

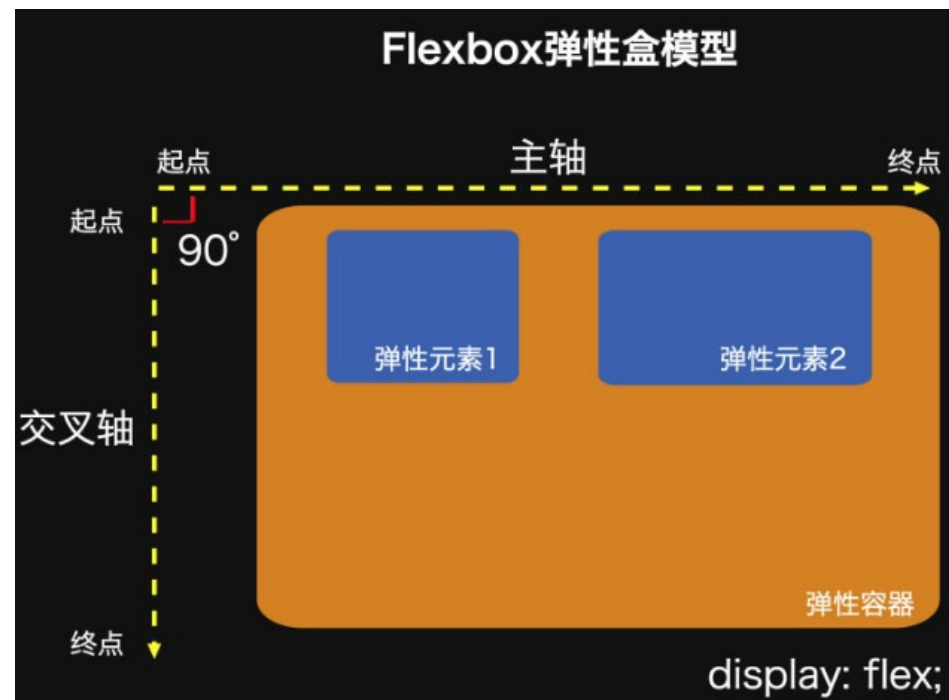
```
div{  
  display: flex;  
}
```

任何一个容器都可以指定为Flex布局

2、弹性布局属性

(1) 弹性容器的属性: `flex-direction`、`flex-wrap`、`justify-content`、`align-items`

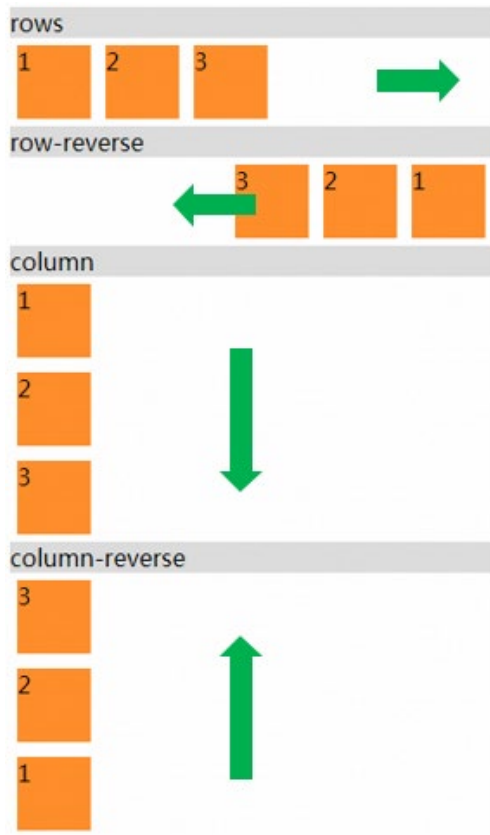
(2) 弹性元素的属性: `flex-grow`、`flex-shrink`、`flex-basis`、`flex`



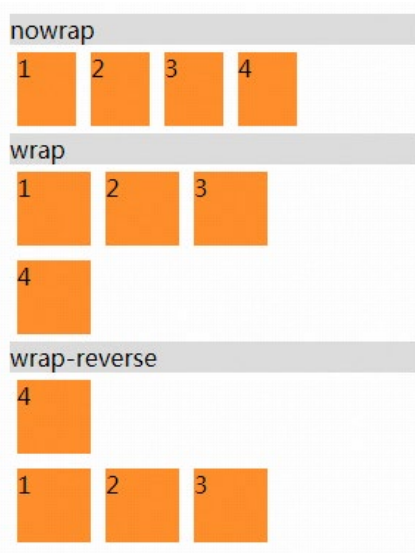
>>> 单选题:弹性布局

2、容器的属性

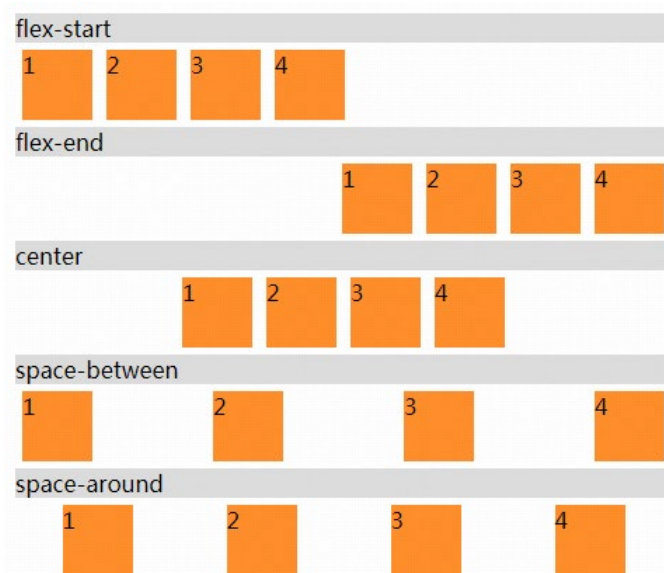
flex-direction主轴方向



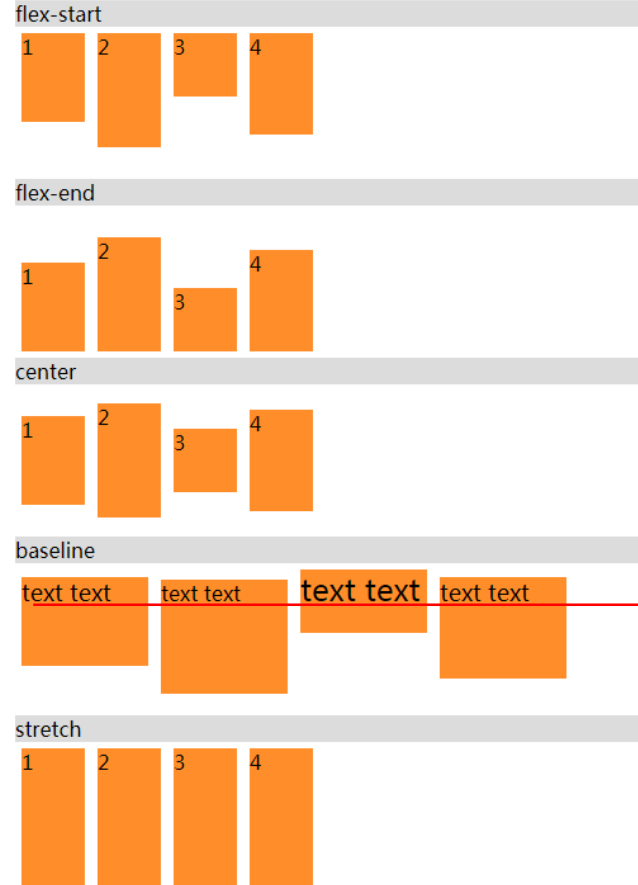
flex-wrap换行



justify-content主轴对齐方式



align-items交叉轴对齐方式



基线

3、弹性元素的属性

(1) flex-grow:

(2) flex-shrink:

(3) flex-basis:

(4) flex属性: flex-grow, flex-shrink和flex-basis的简写

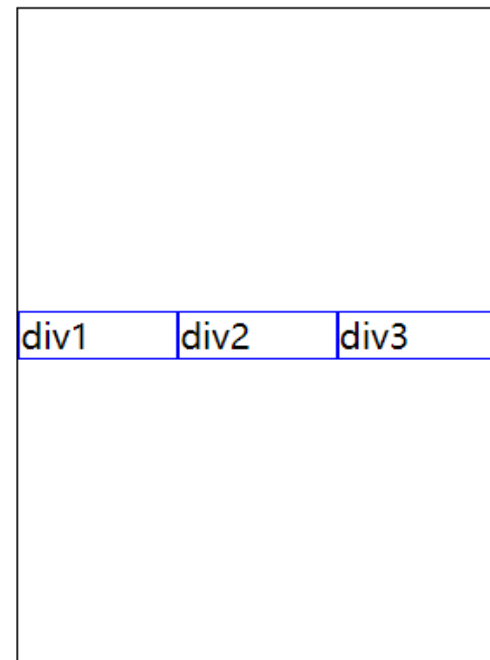
flex: 1

- 计算值为「1 1 0%」

```
Styles Computed Layout E
Filter
element.style {
}
#main div {
  flex: 1;
  flex-grow: 1;
  flex-shrink: 1;
  flex-basis: 0%;
  border: 1px solid blue;
}
```

```
<div id="main">
  <div>div1</div>
  <div>div2</div>
  <div>div3</div>
</div>
```

```
#main{
  width:220px;
  height:300px;
  border:1px solid black;
  display:flex;
  align-items: center;
}
#main div{
  flex:1;
  border: 1px solid blue;
}
```



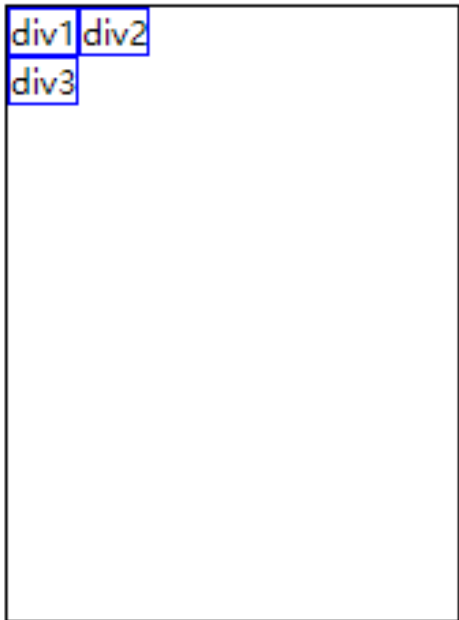
4、float、clear

弹性子元素float和clear全部失效

(1) 容器中子元素设置float、clear属性

```
<div id="main">
  <div>div1</div>
  <div>div2</div>
  <div>div3</div>
</div>
```

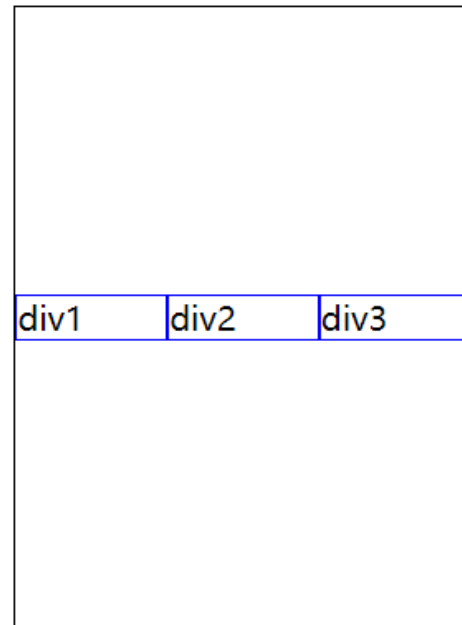
```
#main{
  width:220px;
  height:300px;
  border:1px solid black;
}
#main div{
  float: left;
  border: 1px solid blue;
}
#main div:last-child{
  clear: left;
}
```



(2) 添加弹性布局后子元素float、clear失效

```
<div id="main">
  <div>div1</div>
  <div>div2</div>
  <div>div3</div>
</div>
```

```
#main{
  width:220px;
  height:300px;
  border:1px solid black;
  display:flex;
  align-items: center;
}
#main div{
  flex:1;
  float: left;
  border: 1px solid blue;
}
#main div:last-child{
  clear: left;
}
```



多选题

关于HTML DOM document对象错误的是（ ）。

- A、不是每个载入浏览器的HTML文档都会成为document对象
- B、document对象使我们可以从脚本中对HTML页面中的所有元素进行访问
- C、document对象是Window对象的一部分，可通过window.document属性对其进行访问
- D、document对象中title属性不可以修改网页的标题

考核知识和技能：

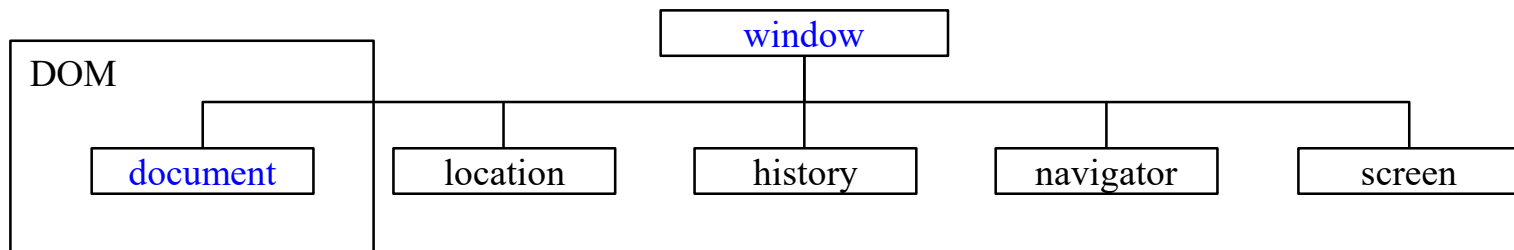
- ✓ BOM (Browser Object Model)
- ✓ DOM (文档对象模型Document Object Model)
- ✓ window对象
- ✓ document对象

>>> 多选题:BOM、DOM

解析:

1、BOM (Browser Object Model)

window对象、document对象、location对象、screen对象、history对象、navigator对象



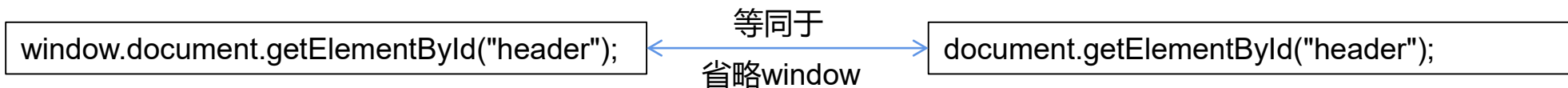
2、window对象

windows对象充当全局对象，所有全局 JavaScript 对象，函数和变量自动成为 window 对象的成员。

全局变量是 window 对象的属性。

全局函数是 window 对象的方法。

document 对象也是 window 对象的属性。

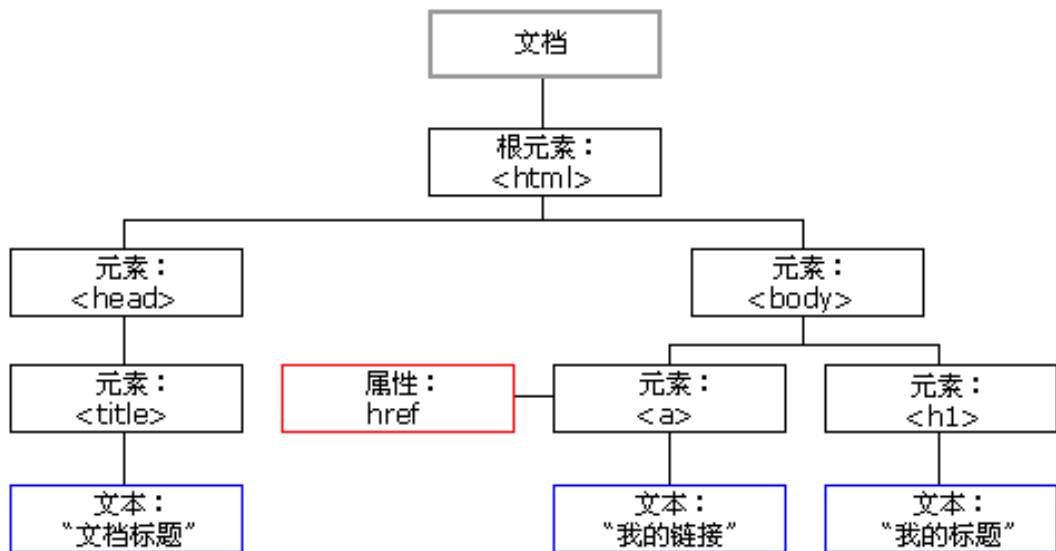


>>> 多选题: BOM、DOM

3、DOM (Document Object Model)

当网页被加载时，浏览器会创建页面的文档对象模型 (Document Object Model)

通过 HTML DOM 能够访问和改变 HTML 文档的所有元素



```
<!DOCTYPE html>
<html>
  <head>
    <title>文档标题</title>
  </head>
  <body>
    <a href="#">我的链接</a>
    <h1>我的标题</h1>
  </body>
</html>
```

4、Document对象(HTML DOM Document)

每个载入浏览器的 HTML 文档都会成为 Document 对象

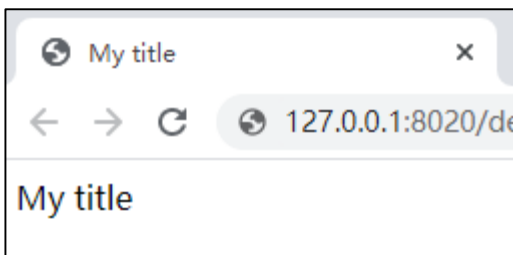
HTMLDocument 接口对 DOM Document 接口进行了扩展，定义 HTML 专用的属性和方法。

属性： body属性、cookie属性、domain属性、lastModified属性、referrer属性、**title属性**、URL属性

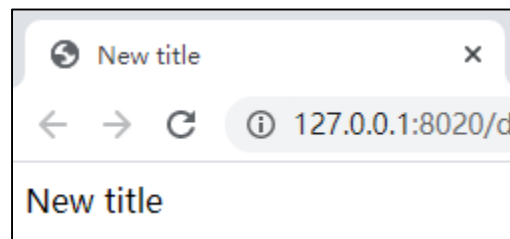
方法： close()、getElementById()、getElementsByName()、getElementsByTagName()、open()、write()、writeln()

document.title属性： 返回和设置当前文档的标题

```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write(document.title);//返回文档标题
    </script>
  </body>
</html>
```



```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <script type="text/javascript">
      document.title="New title";//设置文档标题
      document.write(document.title);
    </script>
  </body>
</html>
```



多选题

下列关于JavaScript中return的含义描述错误的是（ ）。

- A、return可以将函数的结果返回给当前函数名
- B、如果函数中没有return，则返回false
- C、return可以用来结束一个函数
- D、return只能返回一个值

考核知识和技能：

- ✓ 函数定义
- ✓ 函数调用
- ✓ return

多选择题:JavaScript函数

解析：

1、函数定义

通过 function 关键词进行定义，其后是函数名和括号 ()，括号中包括由逗号分隔的参数

```
function m(a, b) {  
    return a * b;  
}
```

2、函数调用

使用函数名调用函数

```
function myFun(a, b) {  
    return a * b;  
}  
myFun(10, 2); // 结果 20
```

3、函数返回

- (1) **return [[expression]]**;表达式的值会被返回，如果忽略，则返回 undefined。
- (2) 没有return语句的函数（或者在没有命中的情况下结束执行的函数）将返回 undefined

正常返回值

```
var x = myFunction(7, 8);  
document.write(x);  
  
function myFunction(a, b) {  
    var c = a*b;  
    return c;  
}
```

输出结果:

56

无返回值

```
var x = myFunction(7, 8);  
document.write(x);  
  
function myFunction(a, b) {  
    var c = a*b;  
    return;  
}
```

输出结果:

undefined

没有return语句

```
var x = myFunction(7, 8);  
document.write(x);  
  
function myFunction(a, b) {  
    var c = a*b;  
}
```

输出结果:

undefined

>>> 多选题:JavaScript函数

4、事件返回

在大多数情况下，为事件处理函数如果让其返回 false，可以防止默认的事件行为。

return false 就相当于终止符，return true 就相当于执行符。在 js 中 return false 的作用一般是用来取消默认动作的。

```
<h2>登录页面</h2>
<form action="index.html" method="get" onsubmit="return login()">
  用户名<input type="text" id="user" /><br/>
  密码<input type="password" id="password" /><br/>
  <input type="submit" value="登录">
</form>
```

//添加点击提交按钮触发onclick事件，调用login()的函数

```
function login(){
  //阻止表单提交，停留在登录页面
  return false;
}
```

← → ↻ 127.0.0.1:8020/debug/login.html

登录页面

用户名

密码

//添加点击提交按钮触发onclick事件，调用login()的函数

```
function login(){
  //表单正常提交，跳转至index.html页面
  return true;
}
```

← → ↻ ⓘ 127.0.0.1:8020/debug/index.html?

首页

多选题

关于jQuery的选择器，下列描述错误的是（ ）。

- A、\$(div span)表示匹配所有后代元素
- B、\$('div>span')表示匹配直接子元素
- C、\$('div + next')表示匹配紧接在div元素后的next元素
- D、无法匹配元素的所有同辈元素

考核知识和技能：

- ✓ jQuery基础语法
- ✓ jQuery选择器

解析:

1、jQuery 基础语法

jQuery基础语法: `$(selector).action();`

- (1) '\$': 美元符号定义 jQuery;
- (2) selector: 选择器, 使用双引号或单引号;
- (3) action(): 执行对元素的操作;

```
//选择所有<p>元素  
$("p").fadeOut()
```

2、jQuery 选择器

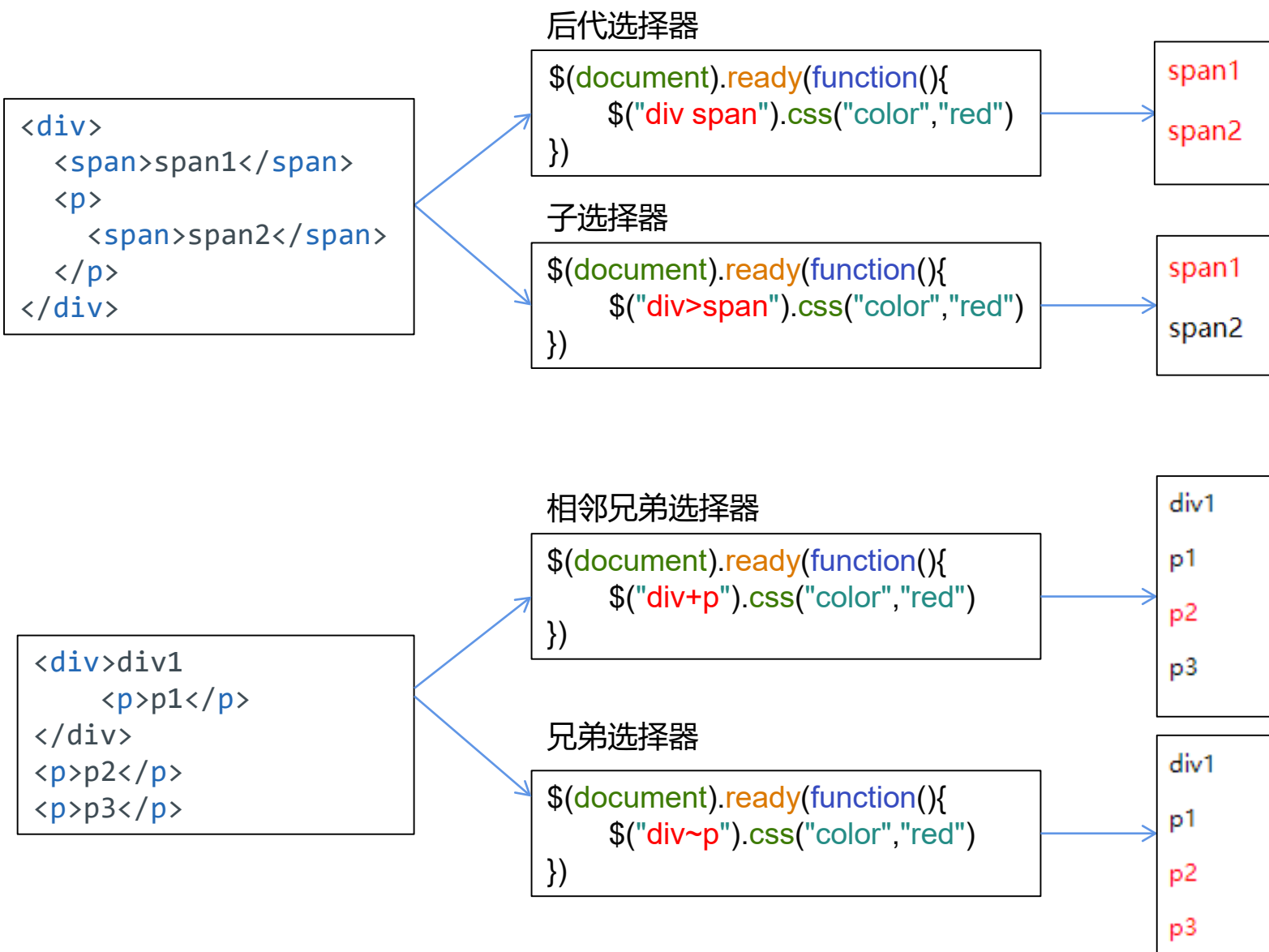
基于已经存在的 CSS 选择器, 除此之外, 它还有一些自定义的选择器。

包括: 元素选择器、id选择器、class类选择器、通配符选择器、后代选择器、子选择器、兄弟选择器、相邻兄弟选择器、属性选择器、伪类选择器等

>>> 多选题:jQuery选择器

3、jQuery层次选择器

- (1) 后代选择器
- (2) 子选择器
- (3) 相邻兄弟选择器
- (4) 兄弟选择器





THANKS