

[笔记][LIKE-H5][10]

前端

[笔记][LIKE-H5][10]

- 067. 标签选择器(掌握)
- 068. id选择器(掌握)
- 069. 类选择器(掌握)
- 070. id选择器和类选择器(理解)
- 071. 后代选择器(掌握)
- 072. 子元素选择器(掌握)
- 073. 后代选择器和子元素选择器(理解)
- 074. 交集选择器(理解)
- 075. 并集选择器(掌握)
- 076. 兄弟选择器(掌握)
- 077. 序选择器上(掌握)
- 078. 序选择器下(掌握)
- 079. 属性选择器上(理解)
- 080. 属性选择器下(理解)
- 081. 通配符选择器(理解)
- 082. 选择器练习(理解)

067. 标签选择器(掌握)

什么是标签选择器？

- 作用: 根据指定的标签名称, 在当前界面中找到所有该名称的标签, 然后设置属性
- 格式:

```
标签名称{  
    属性:值;  
}
```

注意点:

- 标签选择器选中的是当前界面中所有的标签, 而不能单独选中某一个标签
- 标签选择器无论标签藏得多深都能选中
- 只要是 `html` 中的标签, 就可以作为标签选择器

068. id选择器(掌握)

什么是id选择器?

- 作用: 根据指定的id名称找到对应的标签, 然后设置属性
- 格式:

```
#id名称{  
    属性:值;  
}
```

- 注意点:
每个HTML标签都有一个属性叫做 `id`, 也就是说每个标签都可以设置 `id`
在同一个界面中 `id` 的名称是不可以重复的
在编写 `id` 选择器时一定要在 `id` 名称前面加上 `#`
`id` 的名称是有一定的规范的
`id` 的名称只能由字母/数字/下划线(`a-z 0-9 _`)
`id` 名称不能以数字开头
`id` 名称不能是HTML标签的名称(即不能是 `a h1 img input ...`)
在企业开发中一般情况下**如果仅仅是为了设置样式, 我们不会使用 `id`**, 因为在前端开发中id一般留给 `js` 使用的

069. 类选择器(掌握)

什么是类选择器?

- 作用: 根据指定的类名称找到对应的标签, 然后设置属性
- 格式:

```
.类名{  
    属性: 值;  
}
```

- 注意点:
每个HTML标签都有一个属性叫做 `class`, 也就是说**每个标签都可以设置类名**
在同一个界面中class的名称是**可以重复的**
在编写class选择器时一定要在class名称前面**加上点**
类名的命名规范和 `id` 名称的命名规范一样
类名就是**专门用来给 `CSS` 设置样式的**
在 `HTML` 中每个标签可以**同时绑定多个类名**
- 格式:
`<标签名称 class="类名1 类名2 ...">`

错误的写法:

```
<p class="para1" class="para2">
```

070. id选择器和类选择器(理解)

id和class的区别?

- id相当于人的身份证不可以重复
class相当于人的名称可以重复
- 一个HTML标签只能绑定一个id名称
一个HTML标签可以绑定多个class名称

id选择器和class选择器区别?

- id选择器是以 # 开头
- class选择器是以 . 开头

在企业开发中到底用id选择器还是用class选择器?

- id一般情况下是给js使用的, 所以除非特殊情况, 否则不要使用id去设置样式
- 在企业开发中一个开发人员对类的使用可以看出这个开发人员的技术水平
- 一般情况下在企业开发中要注重冗余代码的抽取, 可以将一些公共的代码抽取到一个类选择器中, 然后让标签和这个类选择器绑定即可

071. 后代选择器(掌握)

什么是后代选择器?

- **作用:** 找到指定标签的所有特定的后代标签, 设置属性
- **格式:**

```
标签名称1 标签名称2{  
    属性: 值;  
}
```

先找到所有名称叫做“标签名称1”的标签, 然后再在这个标签下面去查找所有名称叫做“标签名称2”的标签, 然后在设置属性

- **注意点:**
后代选择器必须用空格隔开
后代不仅仅是儿子, 也包括孙子/重孙子, 只要最终是放到指定标签中的都是后代
后代选择器**不仅仅可以使用标签名称, 还可以使用其它选择器**
后代选择器可以通过空格一直延续下去

072. 子元素选择器(掌握)

什么是子元素选择器?

- **作用:** 找到指定标签中所有特定的直接子元素, 然后设置属性
- **格式:**

```
标签名称1>标签名称2{  
    属性: 值;  
}
```

先找到所有名称叫做“标签名称1”的标签, 然后在这个标签中查找所有**直接子元素**名称叫做“标签名称2”的元素

- **注意点:**
 - 子元素选择器只会查找儿子, 不会查找其他被嵌套的标签
 - 子元素选择器之间需要用 `>` 符号连接, 并且不能有空格
 - 子元素选择器**不仅仅可以使用标签名称, 还可以使用其它选择器**
 - 子元素选择器可以通过 `>` 符号一直延续下去

073. 后代选择器和子元素选择器(理解)

后代选择器和子元素选择器之间的区别?

- 后代选择器使用空格作为连接符号
- 子元素选择器使用 `>` 作为连接符号
- 后代选择器会选中指定标签中, 所有的特定后代标签, 也就是会选中儿子/孙子..., 只要是被放到指定标签中的特定标签都会被选中
- 子元素选择器只会选中指定标签中, 所有的特定的直接标签, 也就是只会选中特定的儿子标签

后代选择器和子元素选择器之间的共同点

- 后代选择器和子元素选择器都可以使用**标签名称** / **id 名称** / **class 名称**来作为选择器
 - 后代选择器和子元素选择器都可以通过各自的连接符号一直延续下去
- ```
选择器1>选择器2>选择器3>选择器4{}
```

### 在企业开发中如何选择

- 如果想选中指定标签中的所有特定的标签, 那么就使用后代选择器
- 如果只想选中指定标签中的所有特定儿子标签, 那么就使用子元素选择器

## 074. 交集选择器(理解)

什么是交集选择器？

- **作用:** 给所有选择器选中的标签中, 相交的那部分标签设置属性
- **格式:**

```
选择器1选择器2{
 属性: 值;
}
```

**注意点:**

- 选择器和选择器之间**没有任何的连接符号**
- 选择器可以使用标签名称/id名称/class名称
- 交集选择器仅仅作为了解, 企业开发中用的并不多

## 075. 并集选择器(掌握)

什么是并集选择器？

- **作用:** 给所有选择器选中的标签设置属性
- **格式:**

```
选择器1, 选择器2{
 属性: 值;
}
```

- **注意点:**  
并集选择器必须使用 `,` 来连接  
选择器可以使用 **标签名称** / **id 名称** / **class 名称**

## 076. 兄弟选择器(掌握)

**兄弟选择器：**只能选择同级标签。

**相邻兄弟选择器**

**CSS2**

- **作用:** 给指定标签后面**紧跟**的那个标签设置属性
- **格式:**

```
选择器1+选择器2{
 属性:值;
}
```

#### 注意点:

相邻兄弟选择器必须通过 `+` 连接

相邻兄弟选择器只能选中紧跟其后的那个标签, **不能选中被隔开的标签**

### 通用兄弟选择器

CSS3

- **作用:** 给指定选择器后面的所有选择器选中的所有标签设置属性
- **格式:**

```
选择器1~选择器2{
 属性:值;
}
```

- **注意点:**  
通用兄弟选择器必须用 `~` 连接  
通用兄弟选择器选中的是指定选择器后面某个选择器选中的所有标签, 无论有没有被隔开都可以选中

## 077. 序选择器上(掌握)

序选择器是 **CSS3** 中新增的选择器**最具代表性的就是序选择器**

#### 什么是序选择器?

**作用:** 选中指定的任意标签然后设置属性

#### 格式:

- 同级别的第几个
  - `:first-child` 选中同级别中的第一个标签 (jpch89: 选长子)
  - `:last-child` 选中同级别中的最后一个标签
  - `:nth-child(n)` 选中同级别中的第n个标签
  - `:nth-last-child(n)` 选中同级别中的倒数第n个标签
  - `:only-child` 选中父元素中唯一的标签
- 同级同类型的第几个
  - `:first-of-type` 选中同级别中同类型的第一个标签
  - `:last-of-type` 选中同级别中同类型的最后一个标签
  - `:nth-of-type(n)` 选中同级别中同类型的第n个标签

`:nth-last-of-type(n)` 选中同级别中同类型的倒数第n个标签

`:only-of-type` 选中父元素中唯一类型的某个标签

## 078. 序选择器下(掌握)

- 其它用法

`:nth-child(odd)` 选中同级别中的所有奇数

`:nth-child(even)` 选中同级别中的所有偶数

`:nth-child(xn+y)` x和y是用户自定义的, 而n是一个计数器, 从0开始递增 ( y 可以省略 )

`:nth-of-type(odd)` 选中同级别中同类型的所有奇数

`:nth-of-type(even)` 选中同级别中同类型的所有偶数

`:nth-of-type(xn+y)` x和y是用户自定义的, 而n是一个计数器, 从0开始递增 ( y 可以省略 )

## 079. 属性选择器上(理解)

什么是属性选择器?

- 作用: 根据指定的属性名称找到对应的标签, 然后设置属性

- 格式 1:

`[attribute]`

- 作用: 根据指定的属性名称找到对应的标签, 然后设置属性

- 格式 2:

`[attribute=value]` ( 经试验: `[attribute="value"]` 也可以 )

- 作用: 找到有指定属性, 并且属性的取值等于 value 的标签, 然后设置属性
- 最常见的应用场景, 就是用于区分 input 属性

```
input[type=password] {}
<input type="text" name="" id="">
<input type="password" name="" id="">
```

## 080. 属性选择器下(理解)

1. 属性的取值是以什么开头的

`[attribute|=value]` CSS2 ( 了解即可 )

`[attribute^=value]` CSS3 ( value 加不加双引号皆可 )

两者之间的区别:

CSS2中的只能找到 `value` 开头,并且 `value` 是被 `-` 和其它内容隔开的

CSS3中的只要是以value开头的都可以找到, 无论有没有被-隔开

2. 属性的取值是以什么结尾的

`[attribute$=value]` CSS3

3. 属性的取值是否包含某个特定的值的

`[attribute~=value]` CSS2 (了解即可)

`[attribute*=value]` CSS3

两者之间的区别:

CSS2中的只能找到独立的单词, 也就是包含value,并且value是被空格隔开的

CSS3中的只要包含value就可以找到, 无论有没有被隔开

---

## 081. 通配符选择器(理解)

什么是通配符选择器?

- 作用: 给当前界面上所有的标签设置属性
- 格式:

```
*{
 属性: 值;
}
```

**注意点:**

由于通配符选择器是设置界面上所有的标签的属性, 所以在设置之前会遍历所有的标签, 如果当前界面上的标签比较多, 那么**性能就会比较差**, 所以在企业开发中**一般不会使用通配符选择器**

---

## 082. 选择器练习(理解)

**注意**

序选择器选择的是**序数**, 不是索引

即**没有第 0 个**标签这样的说法!

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>选择器练习</title>
 <style>
 /*标签选择器*/
 /*
 p{
 color: red;
```



```
}
*/
/*ID选择器*/
/*
#identity1{
 color: red;
}
#identity2{
 color: red;
}
*/
/*类选择器*/
/*
.para{
 color: red;
}
*/
/*后代选择器*/
/*
div p{
 color: red;
}
*/
/*子元素选择器
div>p{
 color: red;
}*/
/*交集选择器
p.para{
 color: red;
}*/
/*并集选择器
#identity1,#identity2{
 color: red;
}*/
/*通用兄弟选择器
h1~p{
 color: red;
}*/
/*序选择器*/
/*
p:nth-child(n+1){
 color: red;
}
*/
/*
p:nth-of-type(n+1){
 color: red;
}
*/
/*属性选择器*/
p[id]{
```

```
 color: red;
 }

</style>
</head>
<body>
<div>
 <h1>我是标题</h1>
 <p id="identity1" class="para">我是段落1</p>
 <p id="identity2" class="para">我是段落2</p>
</div>
</body>
</html>
```

---

完成于 201811030953