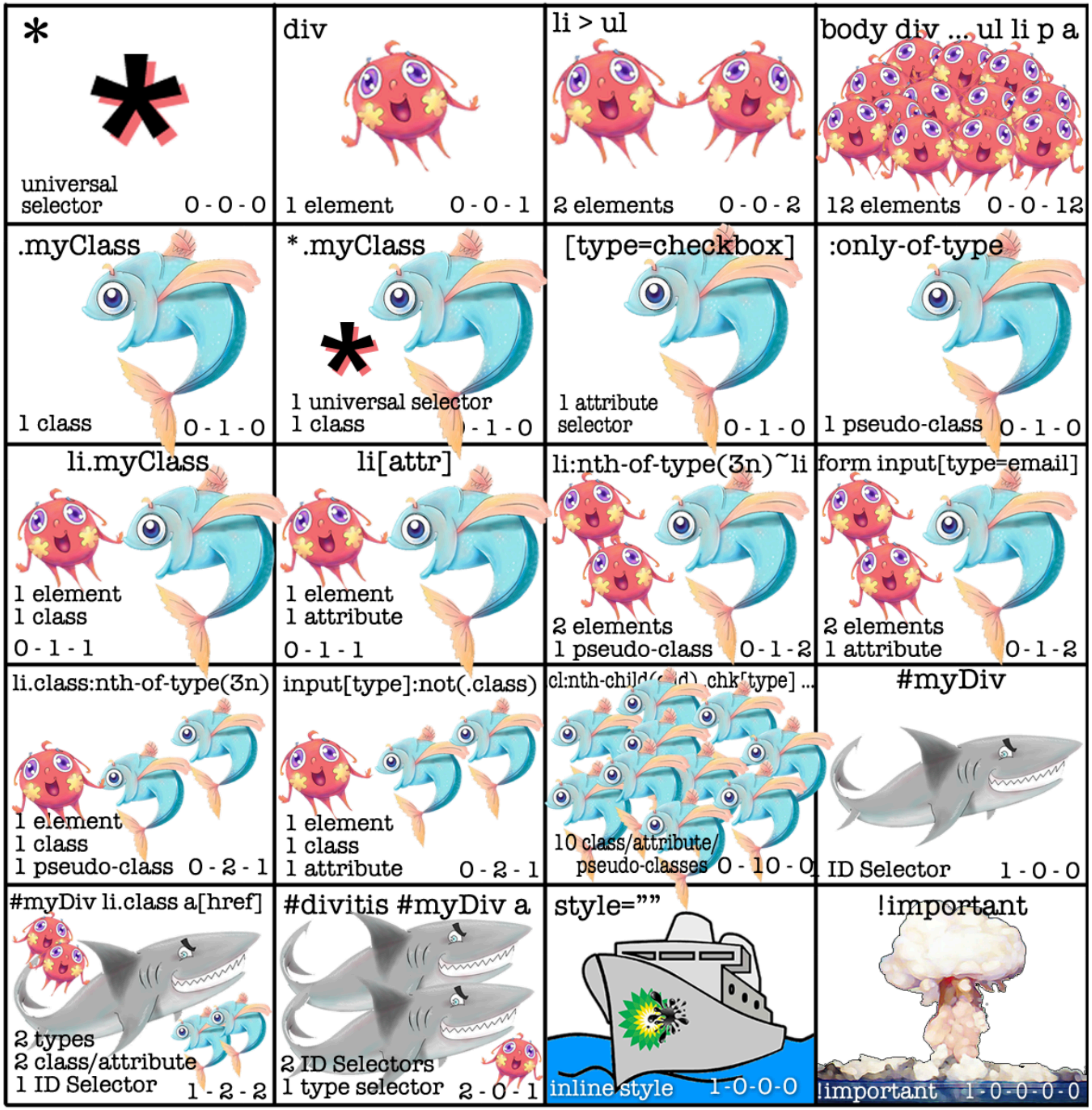


* universal selector 0 - 0 - 0	div 1 element 0 - 0 - 1	li > ul 2 elements 0 - 0 - 2	body div ... ul li p a 12 elements 0 - 0 - 12
.myClass 1 class 0 - 1 - 0	* .myClass 1 universal selector 1 class 0 - 1 - 0	[type=checkbox] 1 attribute selector 0 - 1 - 0	:only-of-type 1 pseudo-class 0 - 1 - 0
li.myClass 1 element 1 class 0 - 1 - 1	li[attr] 1 element 1 attribute 0 - 1 - 1	li:nth-of-type(3n)~li 2 elements 1 pseudo-class 0 - 1 - 2	form input[type=email] 2 elements 1 attribute 0 - 1 - 2
li.class:nth-of-type(3n) 1 element 1 class 1 pseudo-class 0 - 2 - 1	input[type]:not(.class) 1 element 1 class 1 attribute 0 - 2 - 1	cl:nth-child(odd) .chk[type] ... 10 class/attribute/ pseudo-classes 0 - 10 - 0	#myDiv ID Selector 1 - 0 - 0
#myDiv li.class a[href] 2 types 2 class/attribute 1 ID Selector 1 - 2 - 2	#divitis #myDiv a 2 ID Selectors 1 type selector 2 - 0 - 1	style="" inline style 1-0-0-0	!important !important 1-0-0-0-0

```css
* { }

li { }

li:first-line { }

ul li { }

ul ol+li { }

h1 + *[rel=up] { }

ul ol li.red { }
```

* { }	0
li { }	
li:first-line { }	
ul li { }	
ul ol+li { }	
h1 + *[rel=up] { }	
ul ol li.red { }	

* { }	0
li { }	1 (one element)
li:first-line { }	
ul li { }	
ul ol+li { }	
h1 + *[rel=up] { }	
ul ol li.red { }	

* { }	0
li { }	1 (one element)
li:first-line { }	2 (one element, one pseudo-element)
ul li { }	
ul ol+li { }	
h1 + *[rel=up] { }	
ul ol li.red { }	

* { }	0
li { }	1 (one element)
li:first-line { }	2 (one element, one pseudo-element)
ul li { }	2 (two elements)
ul ol+li { }	
h1 + *[rel=up] { }	
ul ol li.red { }	

* { }	0
li { }	1 (one element)
li:first-line { }	2 (one element, one pseudo-element)
ul li { }	2 (two elements)
ul ol+li { }	3 (three elements)
h1 + *[rel=up] { }	
ul ol li.red { }	

* { }	0
li { }	1 (one element)
li:first-line { }	2 (one element, one pseudo-element)
ul li { }	2 (two elements)
ul ol+li { }	3 (three elements)
h1 + *[rel=up] { }	11 (one attribute, one element)
ul ol li.red { }	

* { }	0
li { }	1 (one element)
li:first-line { }	2 (one element, one pseudo-element)
ul li { }	2 (two elements)
ul ol+li { }	3 (three elements)
h1 + *[rel=up] { }	11 (one attribute, one element)
ul ol li.red { }	13 (one class, three elements)

li.red.level { }

style=“”

p { }

div p { }

.sith

div p.sith { }

#sith

body #darkside .sith p { }

li.red.level { }	21 (two classes, one element)
style=“”	
p { }	
div p { }	
.sith	
div p.sith { }	
#sith	
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=“”	1000 (one inline styling)
p { }	
div p { }	
.sith	
div p.sith { }	
#sith	
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=""	1000 (one inline styling)
p { }	1 (one HTML selector)
div p { }	
.sith	
div p.sith { }	
#sith	
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=""	1000 (one inline styling)
p { }	1 (one HTML selector)
div p { }	2 (two HTML selectors)
.sith	
div p.sith { }	
#sith	
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=""	1000 (one inline styling)
p { }	1 (one HTML selector)
div p { }	2 (two HTML selectors)
.sith	10 (one class selector)
div p.sith { }	
#sith	
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=“”	1000 (one inline styling)
p { }	1 (one HTML selector)
div p { }	2 (two HTML selectors)
.sith	10 (one class selector)
div p.sith { }	12 (two HTML selectors and a class selector)
#sith	
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=""	1000 (one inline styling)
p { }	1 (one HTML selector)
div p { }	2 (two HTML selectors)
.sith	10 (one class selector)
div p.sith { }	12 (two HTML selectors and a class selector)
#sith	100 (one id selector)
body #darkside .sith p { }	

li.red.level { }	21 (two classes, one element)
style=“”	1000 (one inline styling)
p { }	1 (one HTML selector)
div p { }	2 (two HTML selectors)
.sith	10 (one class selector)
div p.sith { }	12 (two HTML selectors and a class selector)
#sith	100 (one id selector)
body #darkside .sith p { }	112 (HTML selector, id selector, class selector, HTML selector; 1+100+10+1)

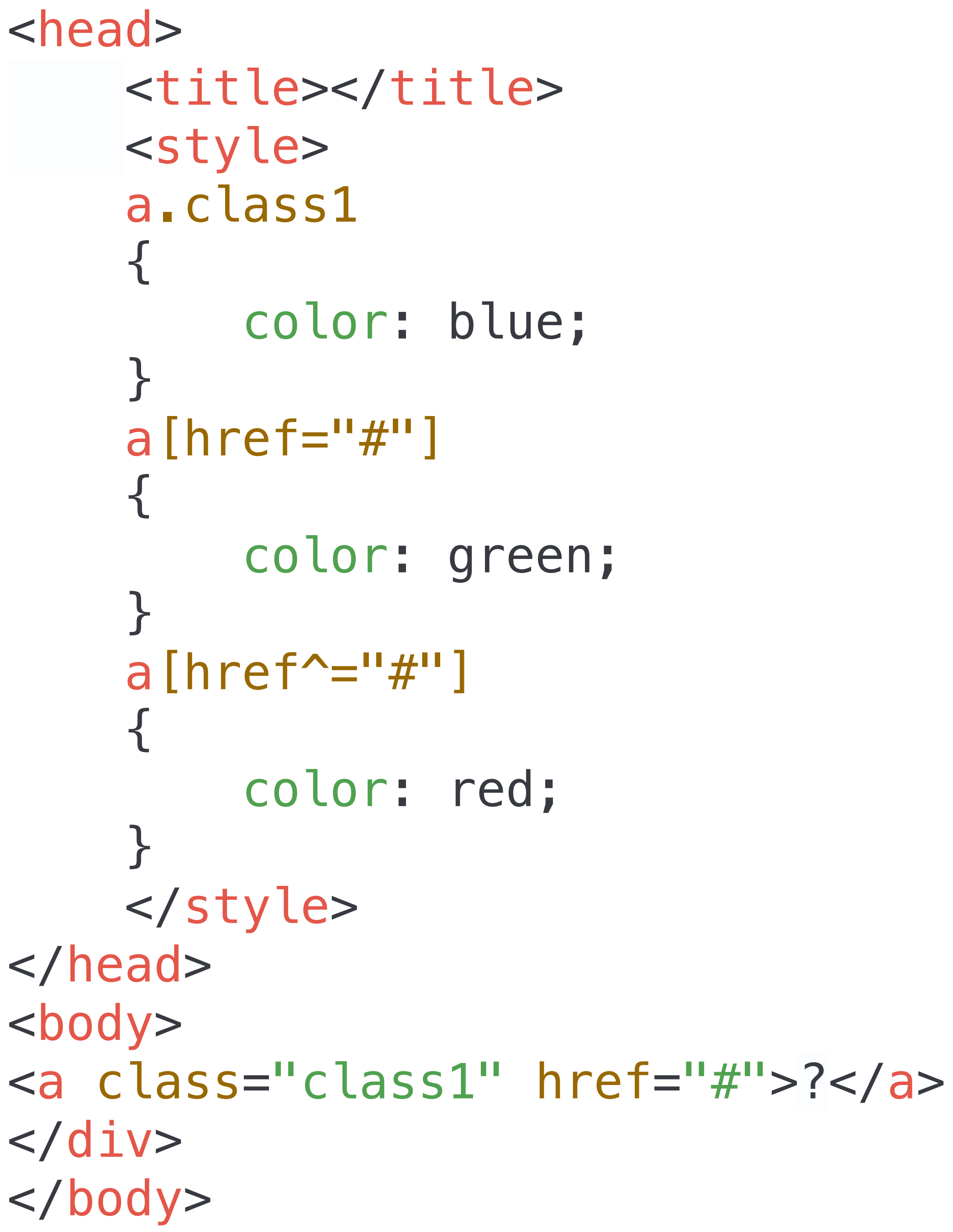
```
<head>
    <title></title>
    <style>
        span p
        {
            background-color: gray;
        }
        html p
        {
            background-color: red;
        }
    </style>
</head>
<body>
    <div>
        <span><p>?</p></span>
    </div>
</body>
```

```
<head>
    <title></title>
    <style>
        div.test1
        {
            background-color: gray;
        }
        div > .test1
        {
            background-color: yellow;
        }

        div .test1
        {
            background-color: red
        }
    </style>
</head>
<body>
<div>
    <div class='test1'>?</div>
</div>
</body>
```

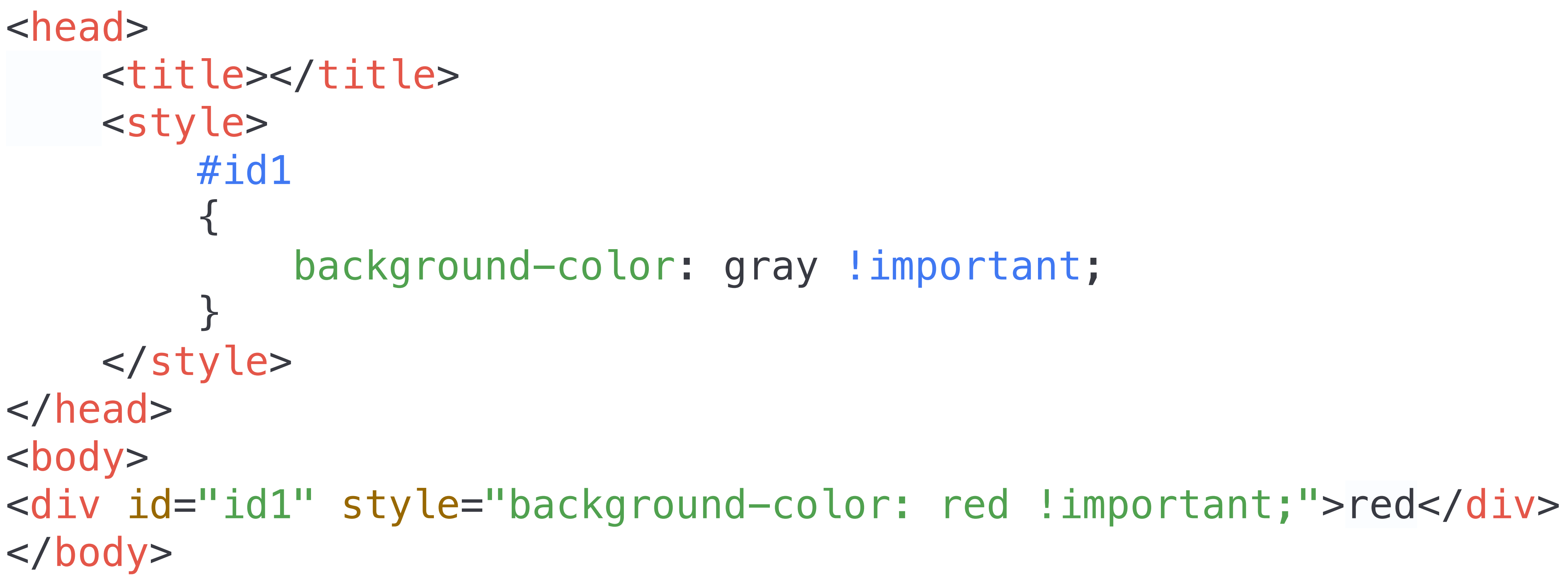
```
<head>
    <title></title>
    <style>
        #id1 div
        {
            background-color: gray;
        }

        #id2
        {
            background-color: red
        }
    </style>
</head>
<body>
<div id="id1">
    <div id="id2">?</div>
</div>
</body>
```



```
<head>
    <title></title>
    <style>
    a.class1
    {
        color: blue;
    }
    a[href="#"]
    {
        color: green;
    }
    a[href^="#"]
    {
        color: red;
    }
    </style>
</head>
<body>
<a class="class1" href="#">?</a>
</div>
</body>
```

```
<head>
    <title></title>
    <style>
        #id1
        {
            background-color: gray !important;
        }
    </style>
</head>
<body>
<div id="id1" style="background-color: red;">?</div>
</body>
```



```
<head>
    <title></title>
    <style>
        #id1
        {
            background-color: gray !important;
        }
    </style>
</head>
<body>
<div id="id1" style="background-color: red !important;">red</div>
</body>
```

Задания

Применить наследуемые и ненаследуемые свойства

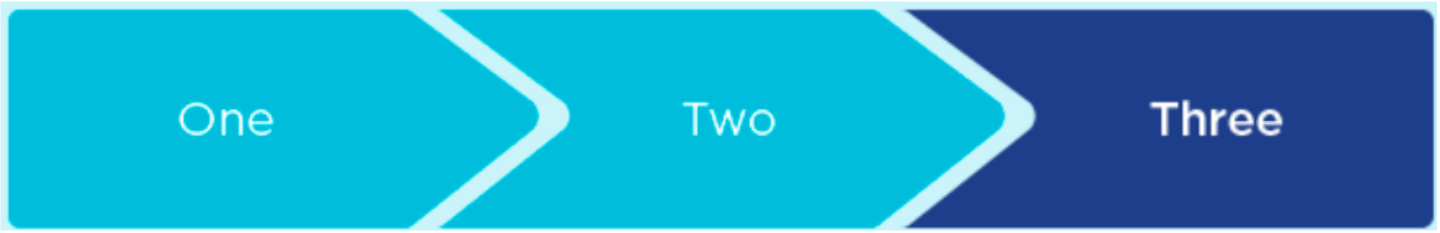
Использовать принудительное наследование

Использовать import/Font-face с подключением 2-3-х шрифтов

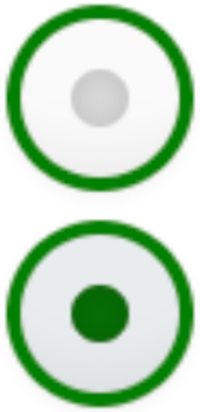
Реализовать хлебные крошки на странице.

Вариант 1:
Цвет всех элементов голубой, последний элемент темно-синий;

Вариант 2:
Цвет всех элементов голубой: при наведении - темно-синий.



Реализовать и стилизовать радиокнопки на странице:



Реализовать `trigger` переключения кнопок с состояниями, как указано на рисунке: