HTML DOM & JS

HTML

- <tag> </tag> - открывающий и закрывающий тег (<p></p>,<a></a>, ….)
- <tag > - тег не требующий закрытия (<img>, <link>,<meta>…)

- <tag atr1 = “value1” atr2 = “value2” atr3 = “value3” …. ></tag>
- <img src = “pic.png” name = “some_name” id = “some_id” alt = “some_text”>

HTML

```
<html>
        <head>
              <title></title>
              <link   >
        <head>
        <body>
              Текст (текст, <h1>, <p>, …)
              Картинка (<img …  >)
              Ссылка (<a  href="other_page.html"… ></a>)
              Таблица (<table>, <tr>, <th>, <td>)
              Списки (<ul>, <ol>, <li>, <dl>, <dt>,<dd>)
        </body>
</html>
```

CSS

- HTML code:

```
<tag style = “property: value;”> <tag>

<tag class = “class_name”> <tag>

<tag id= “some_id”> <tag>

<tag class = “class1 class2”> <tag>
```

- CSS code:

```
tag {property: value;}

.class_name {property: value;}

#some_id {property: value;}

.class_name :pseudo-classes {property: value;}
```

CSS

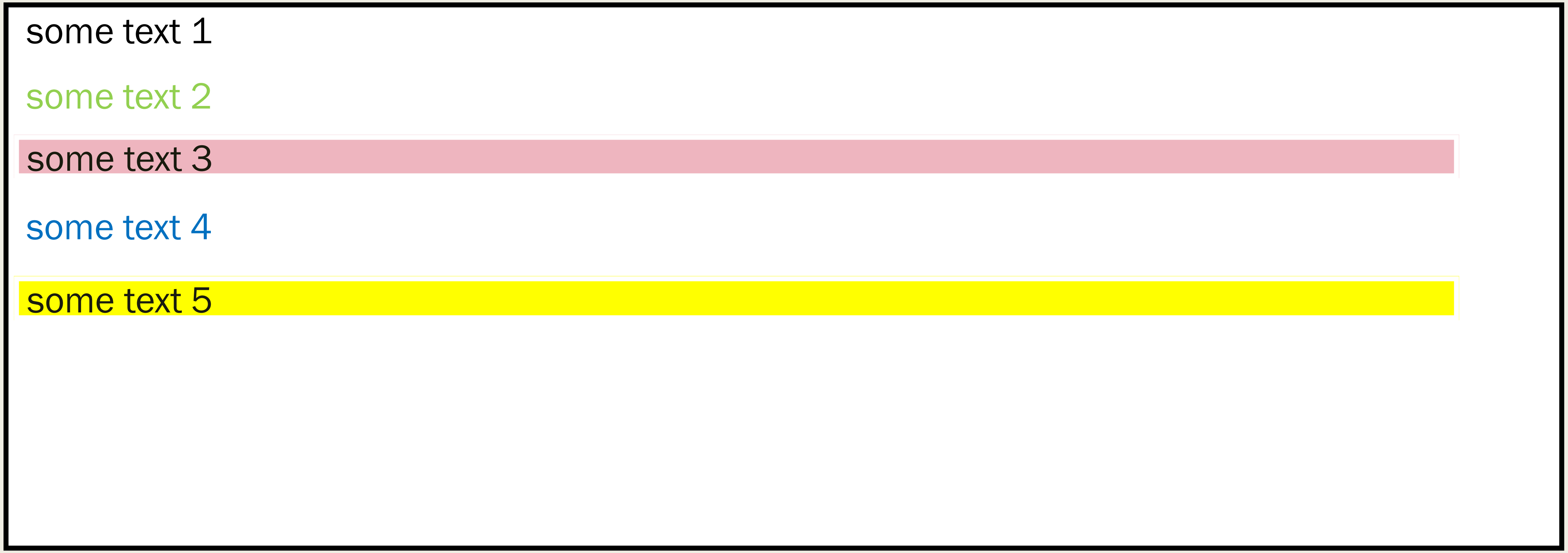
- HTML code:

```
<div class=“div_page”>
   <p>some text 1</p>
   <p class=“cl_name_1”>some text 2</p>
   <p id=“p_id”>some text 3</p>
</div>
<p class=“cl_name_1”>some text 4</p>
<p id=“p_id”>some text 5</p>
```

- CSS code:

```
.div_page{color: red;}
.div_page p {color: black;}
.cl_name_1 {color: blue;}
.div_page p.cl_name_1{color: green;}
#p_id {background-color: yellow;}
div #p_id {background-color: pink;}
```

CSS



some text 1

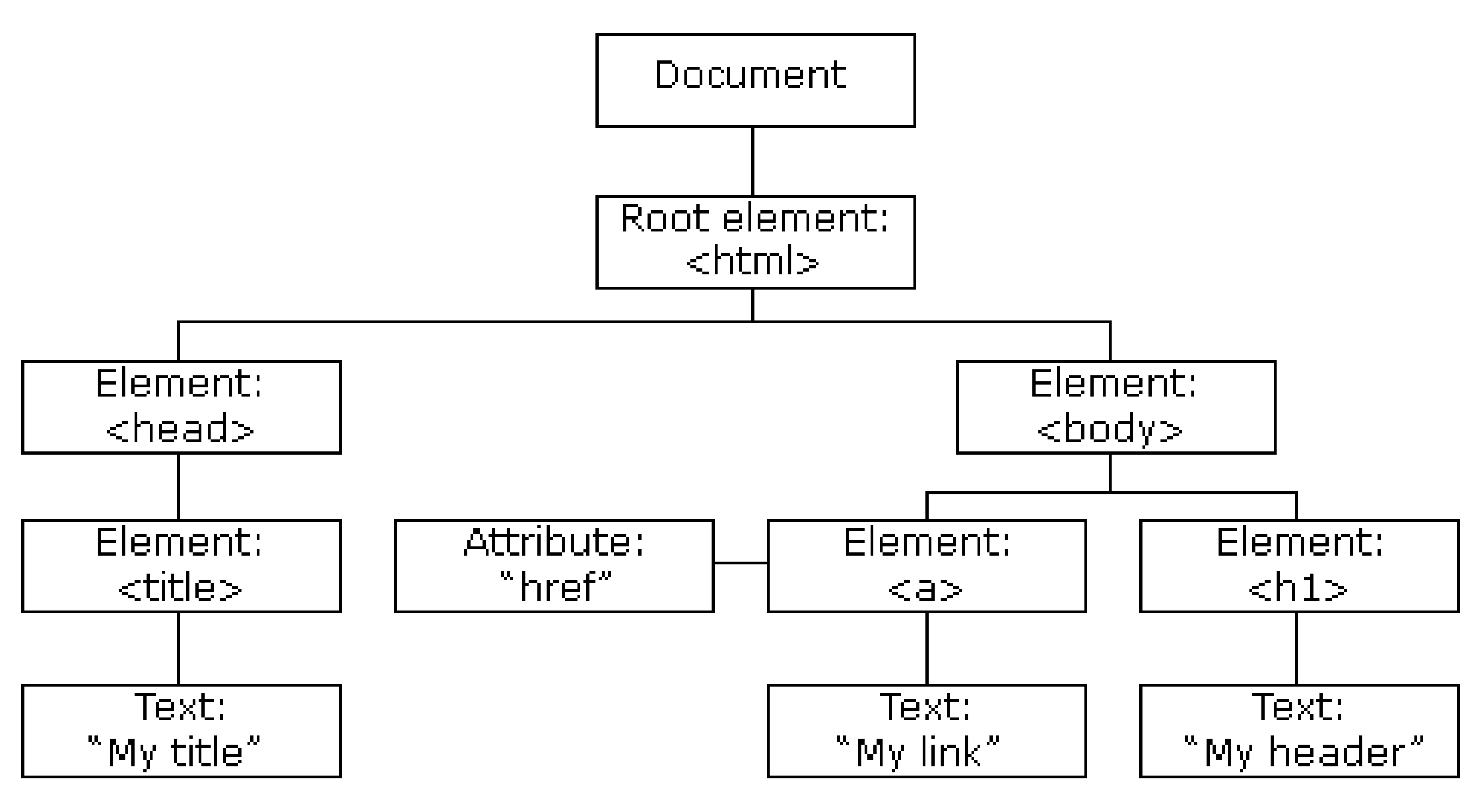
some text 2

some text 3

some text 4

some text 5

HTML DOM (Document Object Model)



HTML

```
<body>
        <div>
                <p><img…><img…> <span></span></p>
                <p><span></span></p>
                <p></p>
        </div>
        <table>
                <tr>  <th></th><th></th><th></th>  </tr>
                <tr>  <td></td><td></td><td></td>  </tr>
                <tr>  <td></td><td></td><td></td>  </tr>
        </table>
</body>
```

HTML DOM

- document.getElementById(*id*)
- document.getElementsByTagName(*name*)
- document.getElementsByClassName(*name*)

var element = document.getElementById(*id*);

- *element*.innerHTML = *“new html content”*
- *element.attribute* = *“new value”*
- *element*.style.*property* = *“new style ”*

HTML DOM

- HTML code:

```
<button type="button“
onclick="document.getElementById('demo').innerHTML = ‘Hello <b>world!</b>’">
Click me</button>
<p id = “demo”> <p>
```

- Page:

Hello **world**!

HTML DOM

- HTML code:

```
<script>
    function myFunc(){
        var tag = document.getElementById(“demo”);
        tag.innerHTML = “Hello world!”;
        tag.style.color = “red”;
    }
<script>
<button type="button“ onclick=“myFunc()”>Click me</button>
<p id = “demo”> <p>
```

- Page:

Hello world!

HTML tags

- <input type=“radio” id=“inp_1” value=“male” > Some text
- document.getElementById(“inp_1“).checked;

- <input type=“checkbox” id=“inp_2”> Some text
- document.getElementById(“inp_2”).checked;

- <input type=“text” id=“inp_3” >
- document.getElementById(“inp_3“).value;

HTML tags

```
<form id="frm1“>
        First name: <input type="text" name="fname" value="Donald"><br>
        Last name: <input type="text" name="lname" value="Duck"><br><br>
</form>
<button onclick="myFunction()">Click me</button>
<p id="demo"></p>
<script>
      function myFunction() {
              var x = document.forms["frm1"];
              var text = "";   var i;
              for (i = 0; i < x.length ;i++) {
                text += x.elements[i].value + "<br>";
              }
              document.getElementById("demo").innerHTML = text;
      }
</script>
```

Лабораторная работа №5.2

- Создать страничку с тестом из работы №5.1. После прохождения теста и при нажатии кнопки пользователь должен увидеть результаты теста.

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page