

DOM Tree Manipulation

Document Object Model (DOM)

- DOM

- https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
-

- Document & element

- <https://developer.mozilla.org/en-US/docs/Web/API/Document>
-
- document.getElementById("ID");
 - document.getElementById("ID").style.color = "blue";
 - <https://developer.mozilla.org/en-US/docs/Web/API/Element/innerHTML>
 -
- document.createElement("type");
-
-

What we need

1. Way to identify a tag
2. Way to modify/add/delete a tag
3. JavaScript
4. Possibility to make web requests in the background

(1) How to identify a Tag?

- Use attribute “id”
 - `<p id="p1">Hello world!</p>`
 - `<button id="b2"> Change color </button>`

(2) Change Tag Content

- Change content of tag
 - `<p id="p1">Hello world!</p>`
 - `document.getElementById("p1").innerHTML="New text!";`

(2) Change Tag Attribute

- Change Style attributes of a tag
 - `document.getElementById("p1").style.color = "blue";`

(2) Create Tag and add

- Add new tag

```
var pic = document.getElementById("pic");
```

```
var tag = document.createElement('img');
```

```
tag.setAttribute('src', 'smile.jpg');
```

```
document.body.insertBefore(tag,pic);
```

(2) Delete Tag

- Remove tag
 - `document.getElementById("p1").remove();`

JavaScript (JS)

- How to add JavaScript (code) in Page?
 - `<script type = "text/javascript"> ... </script>`
- How to call JavaScript?
 - Script -> interpreter, starts executing
 - Script -> Function definitions & statements

```
function ChangeText()  
{ document.getElementById("p1").innerHTML="New text!"; }
```

```
function ChangeColor()  
{ document.getElementById("p1").style.color = "blue"; }
```

```
document.getElementById("b2").addEventListener("click", ChangeColor);
```

JS & HTML

```
<p id="p1">Hello world!</p>
```

```
<button onclick="ChangeText()"> Change text </button>
```

```
<button id="b2"> Change color </button>
```

Python - Functions

Named Function

```
def foo:  
    return 'hello'
```

Unnamed/lambda function

```
greet = lambda : print('Hello World')  
  
greet_user = lambda name : print('Hey there,', name)
```

JS Functions

- Named

- `foo(Args) { };`

- Unnamed

- `function(Args) { };`
 - `let a = function() { };`
 - `a();`
 - `foo(function(Args) { ...};`
 - `app.get('/something', function(req,res) { });`

- Unnamed Arrow functions

- (Args) => { }

```
let a = () => { ..... };
```

```
a();
```

```
app.get('/greeting', (req,res) => {  
    var first_name = req.query.fname;  
    res.send('hello' + first_name);  
});
```

