

Content and Form: How one manipulates the other

1

DAY 2:
HTML
CSS
DOM & JQUERY

HNU

Andrea Kohlhase
Neu-Ulm University of Applied Sciences
Andrea.Kohlhase@hs-neu-ulm.de

FAU

Michael Kohlhase
Friedrich-Alexander-University Nürnberg-Erlangen
Michael.Kohlhase@fau.de



Content and Form: How one manipulates the other

2

HTML

HNU

Andrea Kohlhase
Neu-Ulm University of Applied Sciences
Andrea.Kohlhase@hs-neu-ulm.de

FAU

Michael Kohlhase
Friedrich-Alexander-University Nürnberg-Erlangen
Michael.Kohlhase@fau.de



A First Prototype

3



“Sketch“

HTML

4

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

Document

Contact
Please type in a message:

Your contact eMail address:
xx @ xx .de
Send message



Web-Browser



HTML

5

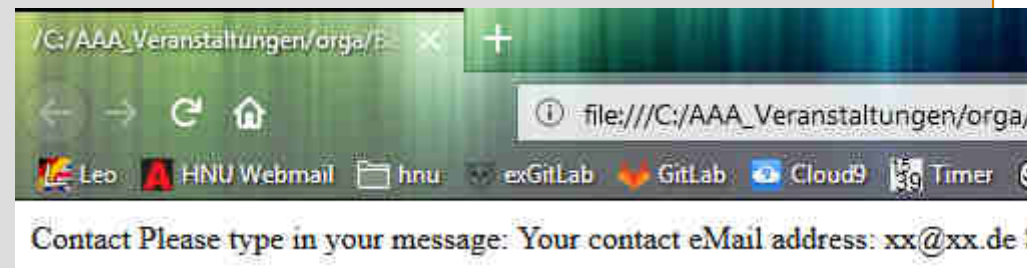
- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

Document

Contact
Please type in a message:

Your contact eMail address:
xx @ xx .de
Send message

Web-Browser



HTML

6

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

title tag
(or title node)

Document

```
<title>Contact</title>
```

Please type in a message:

Your contact eMail address:
xx @ xx.de

Send message

Web-Browser

Contact

file:///C:/AAA_Veranstaltungen/orga

Leo HNU Webmail hnu exGitLab GitLab Cloud9 Timer

Please type in your message: Your contact eMail address: xx@xx.de Send me

HTML

7

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

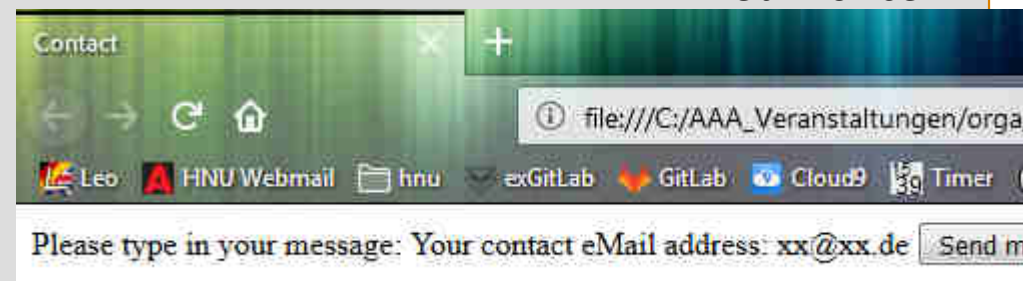
Document

```
<title>Contact</title>
Please type in a message:

Your contact eMail address:
xx @ xx .de
<button>Send message</button>
```

button tag (or button node)

Web-Browser



HTML

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

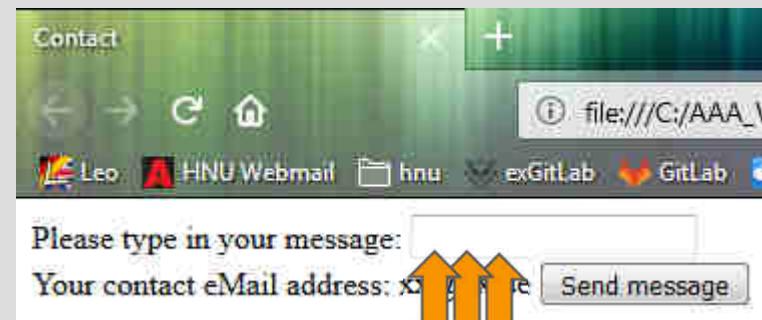
break
tag

input tag

Document

```
<title>Contact</title>
Please type in a message:
<input type="text" />
<br />
Your contact eMail address:
xx @ xx.de
<button>Send message</button>
```

Web-Browser



HTML

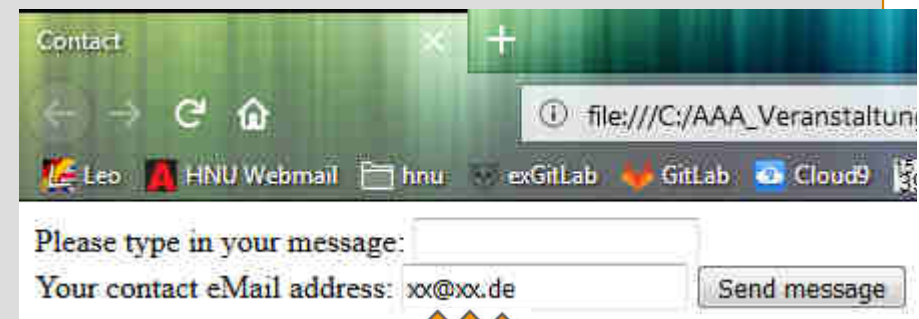
9

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

Document

```
<title>Contact</title>
Please type in a message:
<input type="text"/>
<br/>
Your contact eMail address:
<input type="text"
      value="xx @ xx . de" />
<button>Send message</button>
```

Web-Browser



HTML

10

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

header tags

Document

```
<title>Contact</title>
<h2>Please type in a message:</h2>
<input type="text"/>
<br/>
<h3>Your contact eMail address:</h3>
<input type="text"
      value="xx @ xx.de"/>
<button>Send message</button>
```

Web-Browser



HTML

11

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup

form tag

```
<title>Contact</title>
<form action="contactAfterClick.html">
<h2>Please type in a message:</h2>
<input type="text"/>
<br/>
<h3>Your contact eMail address:</h3>
<input type="text"
      value="xx @ xx.de"/>
<button>Send message</button>
</form>
```

[contactAfterClick.html]



Web-Browser



[CSSExercise-contactForm.html]

HTML

12

- Hypertext Markup Language
 - Markup language to format web pages
 - Logical Markup
- HTML Document Structure
 - Hierarchical nesting of tags

```
1 <html>  
2   <head>  
3     <title>Contact</title>  
4   </head>  
5   <body>  
6     ...  
7   </body>  
8 </html>
```

Configuration
of a webpage

Content
of a webpage

Structure of a Tag

13

- Opening Tag

<name attributeName="value" ... >

- Closing Tag

</name>

Attributes are properties of a tag

- Content of a Tag

<name attributeName="value" ... >

...

</name>

- Abbreviation for an empty tag (one without content)

<name attributeName="value" ... />

Tags, we already used

14

- Title tag (for Browser tab name)

```
<title>...</title>
```

- Header tags

```
<h1>...</h1>, <h2>...</h2>, ... , <h6>...</h6>
```

largest smallest

- Button tag

```
<button>...</button>
```

- Input tags (for input fields)

```
<input type="text" />
```

```
<input type="text" value="..." />
```

- Break tag (for a line break)

```
<br />
```

More Standard Tags

15

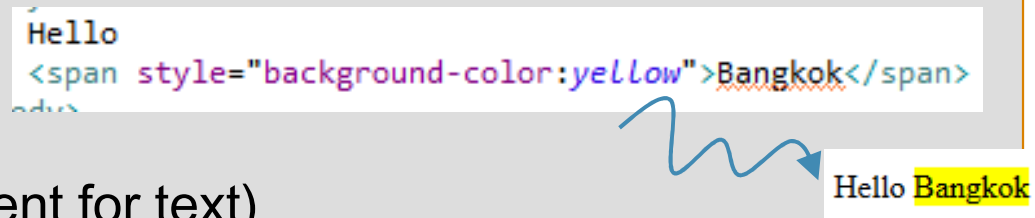
- div tag (a box element, that is, a grouping element with a line break)

```
<div>...</div>
```



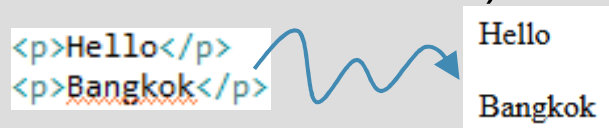
- span tag (an inline markup element)

```
<span>...</span>
```



- Paragraph tag (a box element for text)

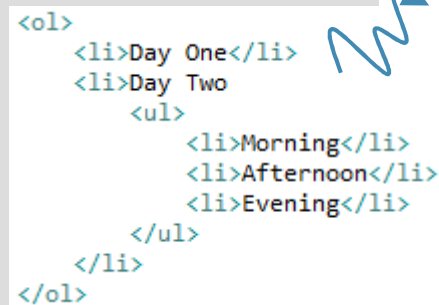
```
<p>...</p>
```



- List tags

ordered list

```
<ol>  
<li>...</li>  
</ol>
```



unordered list

```
<ul>  
<li>...</li>  
</ul>
```

Yet More Standard Tags

16

- Anchor tag (for a hyperlink)

```
<a href="..." >...</a>
```

```
<a href="other.html">some link</a>
```

[some link](#)

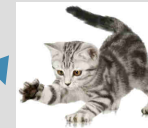
- Image tag

```

```

```

```



- Form tag (for forms)

```
<form action="...">
```

```
<input type="submit" value="..." />
```

```
</form>
```

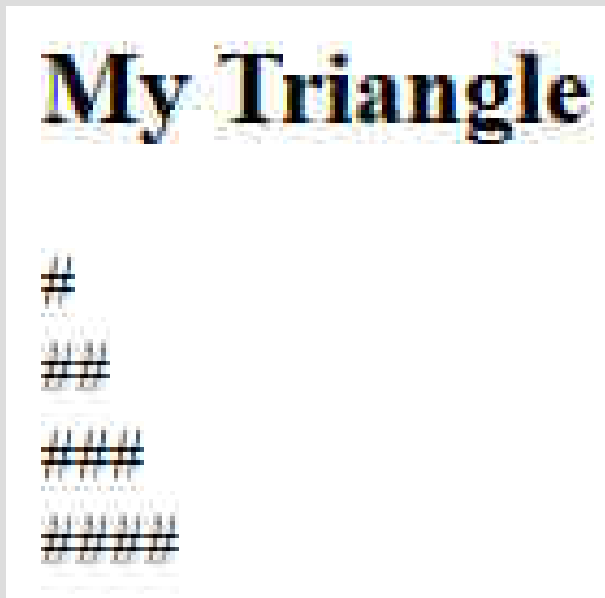
```
<form action="other.html">
  Just click here
  <input type="submit" value="Click me!" />
</form>
```

Just click here

HTML Task 1

17

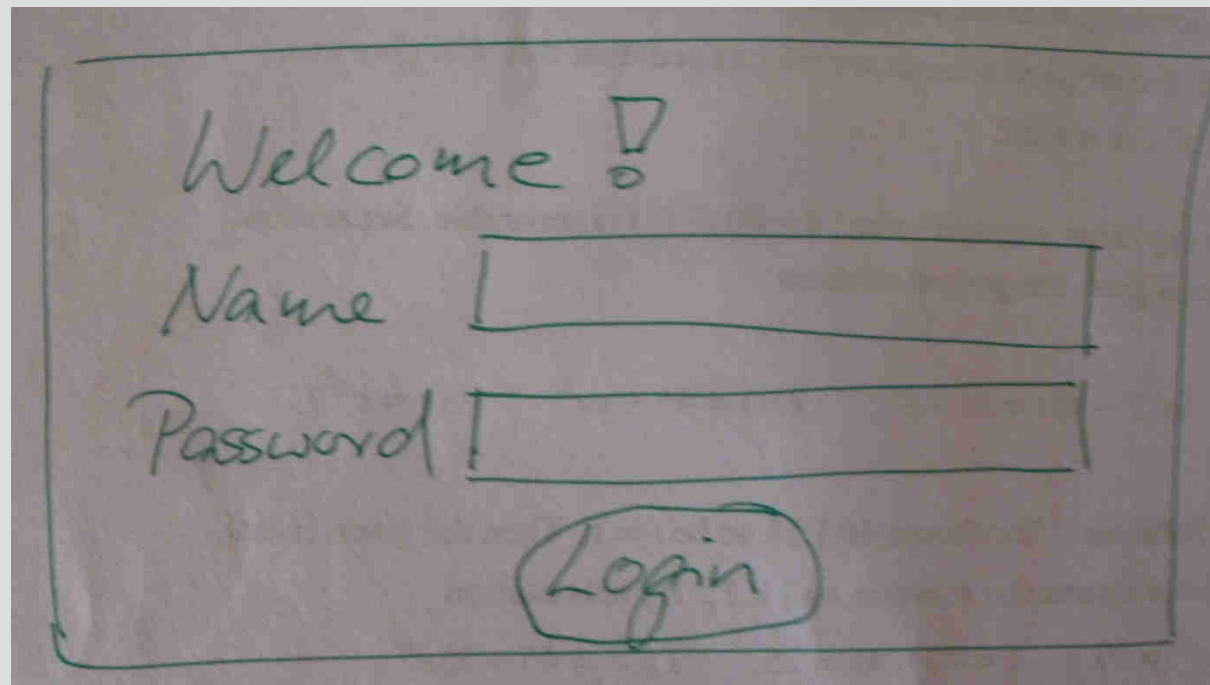
- Create



HTML Task 2

18

- Create a login screen which links to a success message when the login button is clicked



Content and Form: How one manipulates the other

19

CSS

HNU

Andrea Kohlhase
Neu-Ulm University of Applied Sciences
Andrea.Kohlhase@hs-neu-ulm.de

FAU

Michael Kohlhase
Friedrich-Alexander-University Nürnberg-Erlangen
Michael.Kohlhase@fau.de



Cascading Stylesheets (CSS)

20

- Language to markup form properties of specific HTML elements
 - Each HTML-Tag owns special **HTML-Attributes** for this task, to which a value can be assigned. For example,
 - class-Attribute

```
<input class="myClass" .../>
```
 - style-Attribute

```
<input style="color:green" .../>
```
 - **Values** of **CSS-Attributes/Properties** determine position, size and "Layout" of a tag to be presented by the browser

What kinds of CSS attributes are there?

21

- Find it out via World Wide Web! For example <https://www.w3schools.com/cssref/default.asp> :

CSS Properties

Search..

A

align-content	Specifies the alignment between the lines inside a flexible container when the items do not use all available space
align-items	Specifies the alignment for items inside a flexible container
align-self	Specifies the alignment for selected items inside a flexible container
all	Resets all properties (except unicode-bidi and direction)
animation	A shorthand property for all the animation properties
animation-delay	Specifies a delay for the start of an animation

-
-
-

Options for setting CSS attributes?

22

- *In a tag* within the HTML document with the style attribute, e.g.
`<input style="color:green;background-color:red;" />`
- *At a central location in the* HTML document with a style tag, e.g.
`<style type="text/css"> ... </style>`
- *Outside of the* HTML document in a specific *CSS file*, which has to be imported into the HTML document with a link tag, e.g.
`<link type="text/css" rel="stylesheet" href="contentAndForm.css" />`

Every CSS attribute determines the form of at least one HTML tag
→ There has to be a mechanism to reference HTML tags via CSS!

Mechanism to assign CSS attr. to HTML tags?

23

- Structure of a CSS reference

```
Selector {
```

```
    attribute1:value1;
```

```
    attribute2:value2;
```

```
    ...
```

```
}
```

This is an
attribute-
value-list

- **Selectors** specify which HTML tags should carry the CSS attributes

CSS-Selectors

24

How to reference specific HTML tags from within CSS?

- Value for all tags of a specific name (optionally with specific values)

Selector: `tagname[attribute="xyz"]`

- Value for a tag with a known id value :

Selector: `#id`

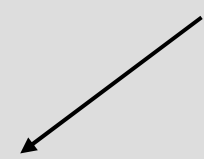
- Value for all tags of a certain class (the classname determines the class of the HTML tag):

Selector: `.classname`

- Value for all tags of the kind tagname of a certain class

Selector: `tagname.classname`

The classname is the value of the class HTML attribute



Find more info on the Web!

CSS Task 1

26

- Create



CSS Task 2

27

- Modify the file “CSSExercise-contactForm.html” so that it looks like



Please type in your message:

Your contact eMail address:

[CSSExercise2a.html]



Please type in your message:

Your contact eMail address:

[CSSExercise2b.html]

Tip: Look on the Web for the CSS attributes width, height, margin-left!

Content and Form: How one manipulates the other

28

DOM AND JQUERY

HNU

Andrea Kohlhase
Neu-Ulm University of Applied Sciences
Andrea.Kohlhase@hs-neu-ulm.de

FAU

Michael Kohlhase
Friedrich-Alexander-University Nürnberg-Erlangen
Michael.Kohlhase@fau.de



Hierarchical Structure = Tree Structure

29

- Idea: A tree has a root, from which is branched out.



A tag branches out into other tags.

Well-Bracketed Structures and Trees

30

- $4x^2+3x+7 = ((4 * (x * x)) + (3 x)) + 7$

Well-Bracketed Structures and Trees

31

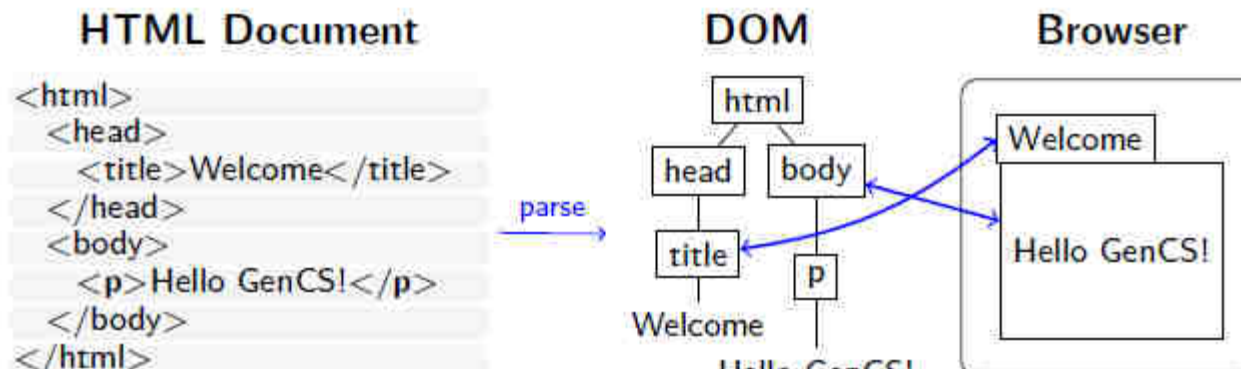
```
1 <html>
2 <head>
3 <title>Contact</title>
4 </head>
5 <body>
6 <form action="contactAfterClick.html">
7 <h2>Please type in your message:</h2>
8 <input type="text"/>
9 <h3>Your contact eMail address:</h3>
10 <input type="text" value="xx@xx.de" />
11 <button>Send message</button>
12 </form>
13 </body>
14 </html>
```

DOM (Document Object Model)

32

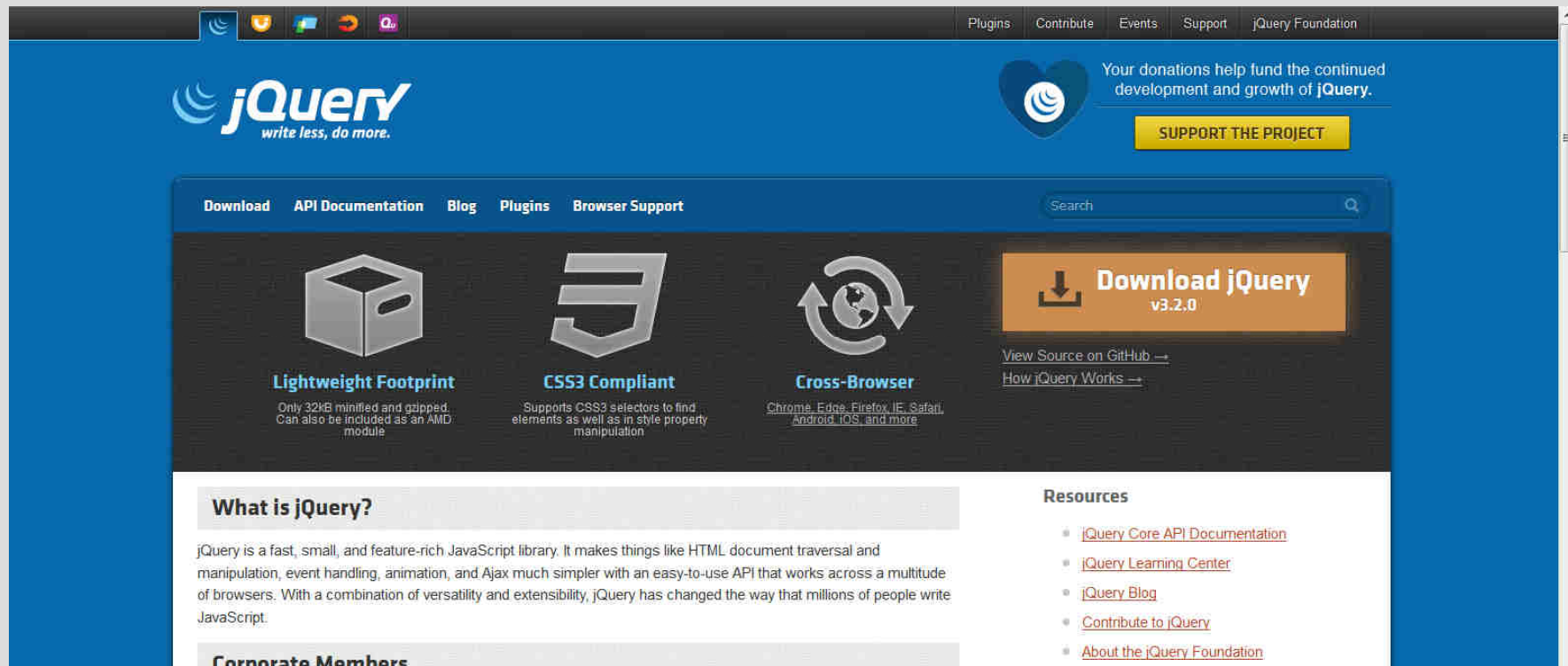
Background: Rendering Pipeline in Browsers

- ▷ **Observation:** The nested, markup codes turn HTML documents into trees.
- ▷ **Definition 0.1** The **document object model (DOM)** is a data structure for the HTML document tree together with a standardized set of access methods.
- ▷ **Rendering Pipeline:** Rendering a web page proceeds in three steps
 1. the browser receives a HTML document,
 2. parses it into an internal data structure, the DOM,
 3. which is then painted to the screen. (**repaint whenever DOM changes**)



jQuery: <https://jquery.com/>

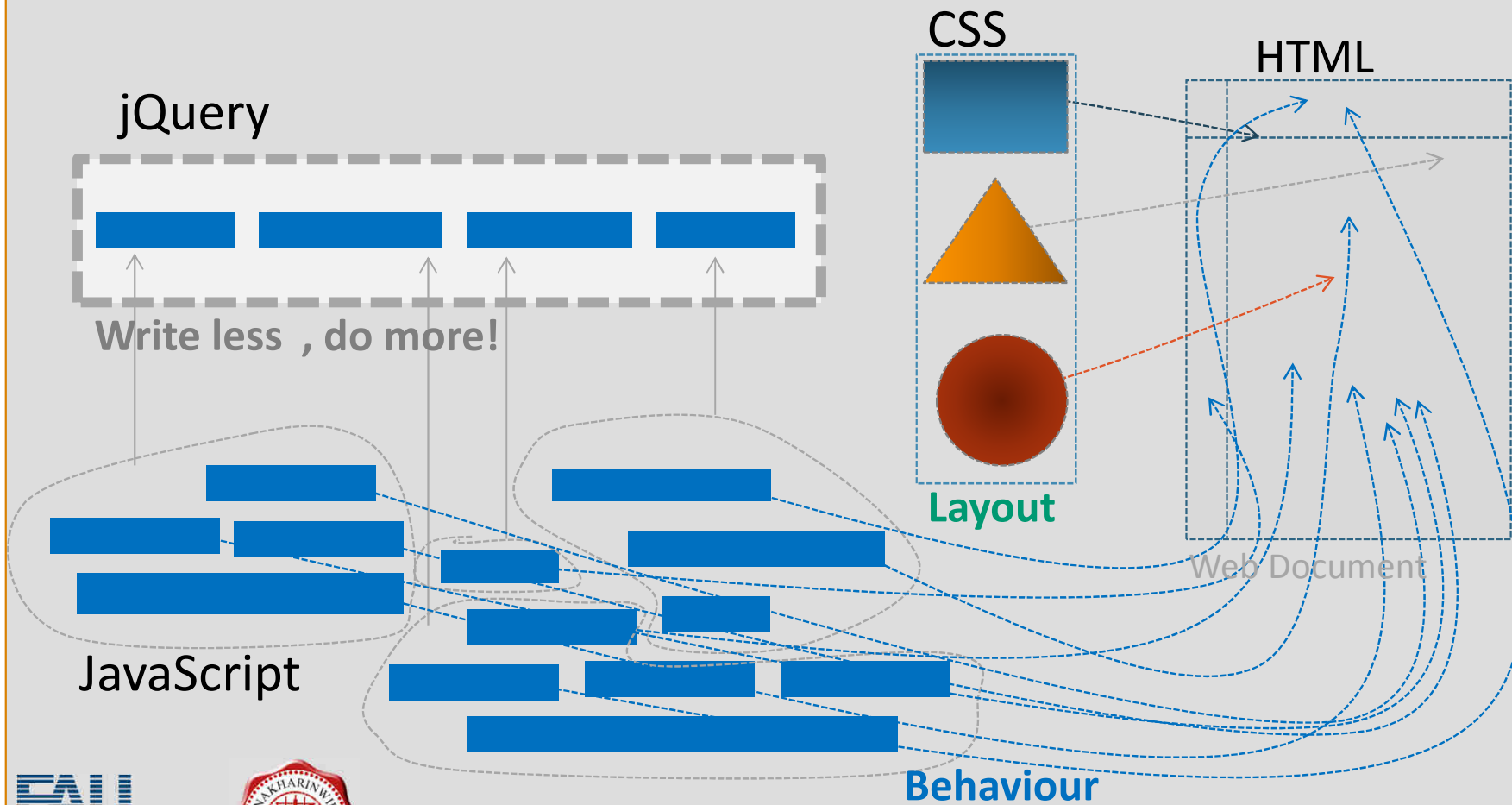
33



The screenshot shows the jQuery website homepage. At the top, there is a navigation bar with links for Plugins, Contribute, Events, Support, and jQuery Foundation. The main header features the jQuery logo with the tagline "write less, do more." and a call to action "SUPPORT THE PROJECT" with a heart icon. Below the header, there is a search bar and a navigation menu with links for Download, API Documentation, Blog, Plugins, and Browser Support. The main content area is divided into three columns: "Lightweight Footprint" (32kB minified and gzipped), "CSS3 Compliant" (supports CSS3 selectors), and "Cross-Browser" (supports Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more). A large orange button labeled "Download jQuery v3.2.0" is prominently displayed, with links to "View Source on GitHub" and "How jQuery Works". Below this, there is a "What is jQuery?" section and a "Resources" section with links to jQuery Core API Documentation, jQuery Learning Center, jQuery Blog, Contribute to jQuery, and About the jQuery Foundation.

jQuery: Framework or Javascript Library

34



jQuery: Philosophy

35

- Find tags to do something with them
- Chaining
- Wrapping, to be able to do more at once

```
$("#myId").show().css("color", "green").slideDown();
```

Select

Modify DOM

Modify Layout

Modify Behaviour

Browser independent

jQuery

Use online jQuery version:

```
<script src=  
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js" />
```

- Javascript library for DOM management
 - Open source and easily accessible
 - Use this way: `$('selector1, selector2, ...').test(...);`
 - ➔ The function `test()` will be applied to all tags which are selected by the given selectors
 - jQuery selectors = CSS selectors
 - `$("#myId")` → Selection of tag with id attribute value = "myId"
 - `$(".important")` → Selection of all tags with class attribute value = "important"
 - `$("li")` → Selection of all "li" tags
 - `$("li[name='myName']")` → Selection of all "li" tags that have a name attribute with the value "myName"

Selector Quiz with jQuery

38

Lecture: jQuery-Quiz

Which clicks trigger what actions?

- First Object
- Second Object
- Third Object
- Fourth Object
- Fifth Object

Blue

Green

Red

Yellow

Reset

Selector Quiz with jQuery

```
<h1 id="myHeader">Lecture: jQuery-Quiz</h1>
<h2>Which clicks trigger what actions?</h2>
<ul id="liste">
  <li color="yes" class="important">First Object</li>
  <li color="no" class="important">Second Object</li>
  <li color="yes" class="important">Third Object</li>
  <li class="important correct">Fourth Object</li>
  <li color="no" class="correct">Fifth Object</li>
</ul>
<input type="button" onclick="writeBlue();" value="Blue"/>
<input type="button" onclick="writeGreen();" value="Green"/>
<input type="button" onclick="writeRed();" value="Red"/>
<input type="button" onclick="writeYellow();" value="Yellow"/>
<input type="button" onclick="writeBlack();" value="Reset"/>
```

HTML

```
function writeGreen(){
  $('.important').css('color','green');
}
function writeRed(){
  $('li[color]').css('color','red');
}
function writeBlue(){
  $('li[color="yes"]').css('color','blue');
}
function writeYellow(){
  $('li[color="no"], .correct').css('background-color','yellow');
}
function writeBlack(){
  $('*').css('color','black').css('background-color','white');
}
```

jQuery

What you can easily do now

40

- Attributes of HTML tags

read: `var v = $('a.menu').attr('href');`

modify: `$('a.menu').attr('href', myPage.html);`

- Value attribute of input fields

read: `var v = $('#myInput').val();`

modify: `$('#myInput').val('Erna');`

- Containing HTML

read: `var content = $('#myOutput').html();`

write: `$('#myOutput').html('<p>This is a test</p>');`

- HTML class attributes

create: `$('#a').addClass('important');`

remove: `$('#a').removeClass('important');`

- CSS attributes

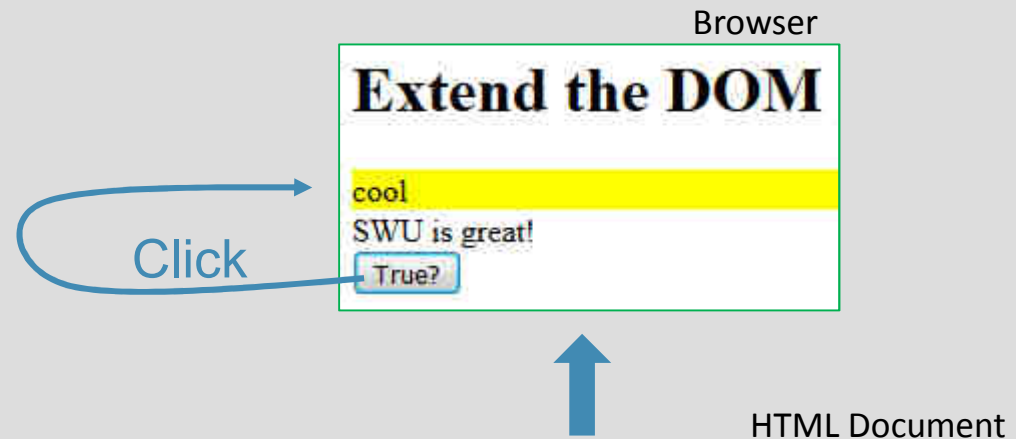
modify: `$('#div').css('color', 'red');`

Extending the DOM

41

- Add HTML code

```
$('#content').prepend($('
```



```
<h1>Extend the DOM</h1>
<div id="content">SWU is great!</div>
<button onclick="sayCool()">True?</button>

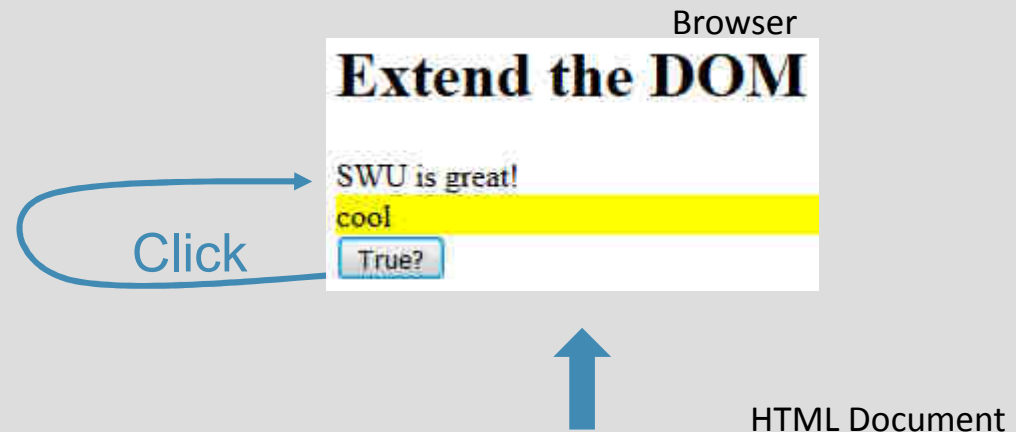
<script>
  function sayCool(){
    $("#content").prepend($('
```


Extending the DOM

42

- Add HTML code

```
$('#content').append($('
```



```
<h1>Extend the DOM</h1>
<div id="content">SWU is great!</div>
<button onclick="sayCool()">True?</button>

<script>
  function sayCool(){
    $("#content").append($('

[jQueryExtendDOM.html]


```

Task

Exercise 1

All red

All blue

All green



Exercise 1

All red

All blue

All green

- Create a screen with the following properties:
 - Input tag of type “button“ with the label “All red“
 - Button tag with the label “All blue“
 - Anchor tag with the textlabel “All green“
- When a button/link is clicked, the font color changes to the promised color.

Tip: The href attribute of an anchor tag can also contain javascript functions, e.g. `href="javascript:setColor('green')"`