729G26 Interaction Programming

Lecture 4

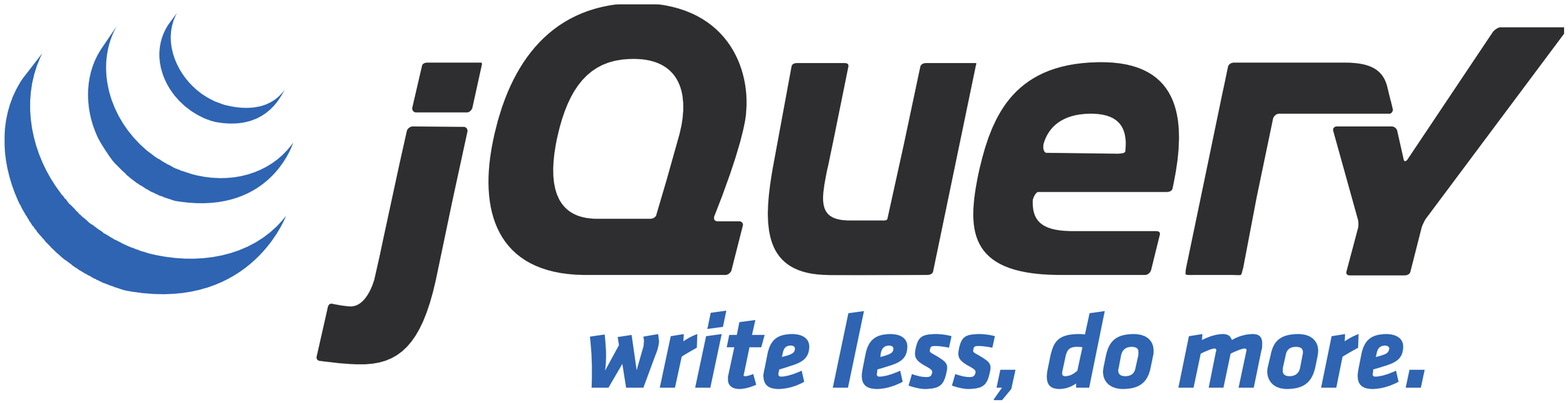
LINKÖPINGS UNIVERSITET

Lecture overview

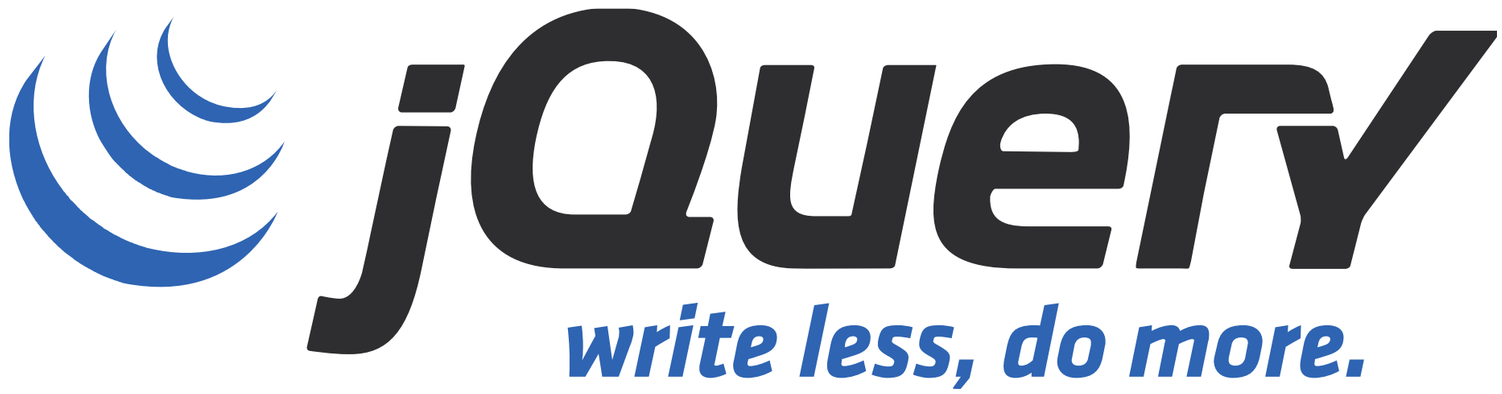
- jQuery - write less, do more
- Capturing events using jQuery
- Manipulating the DOM, attributes and content with jQuery
- Animation with jQuery
- Describing interaction

Todays lecture in the context of the course

- jQuery is the most used JavaScript library. With it you write less, and do more.
- To be able to communicate your interaction ideas, we need some kind of language, e.g. wireframes and transition diagrams
- Next week we will talk about creating UI components using jQuery UI + how to use third party components.

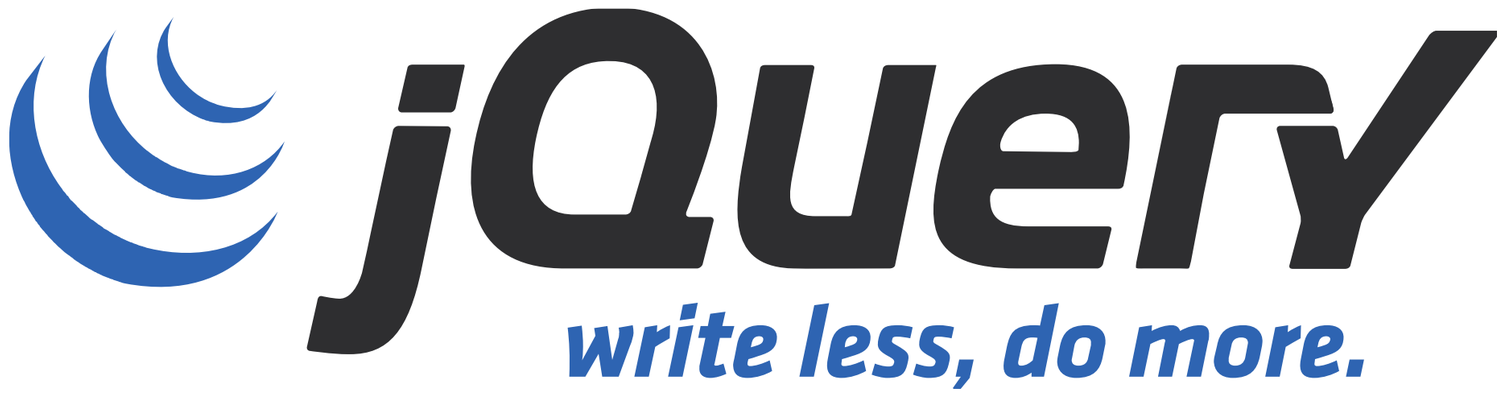


LINKÖPINGS UNIVERSITET



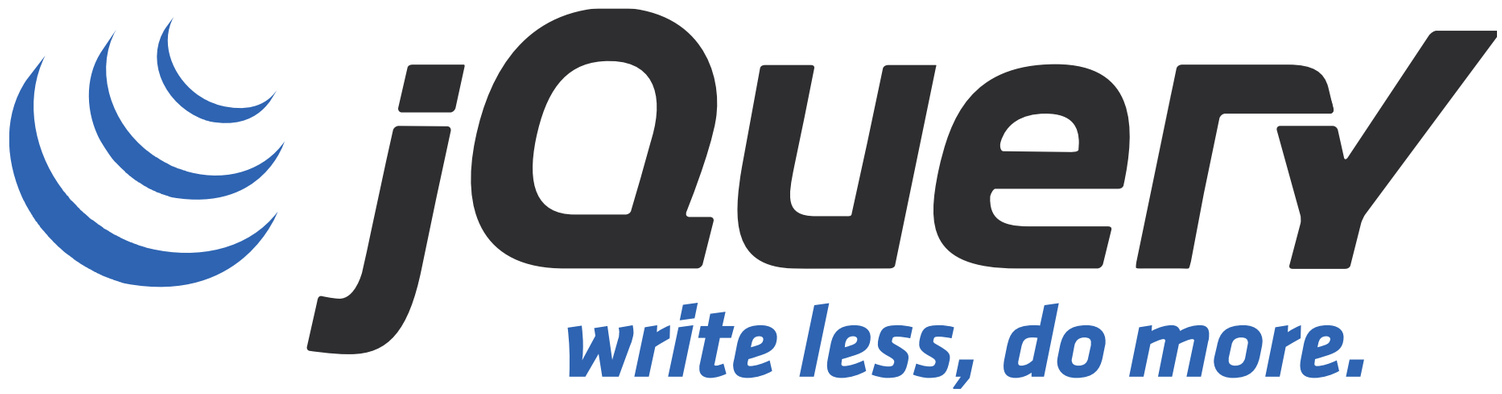
"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."

jQuery.com



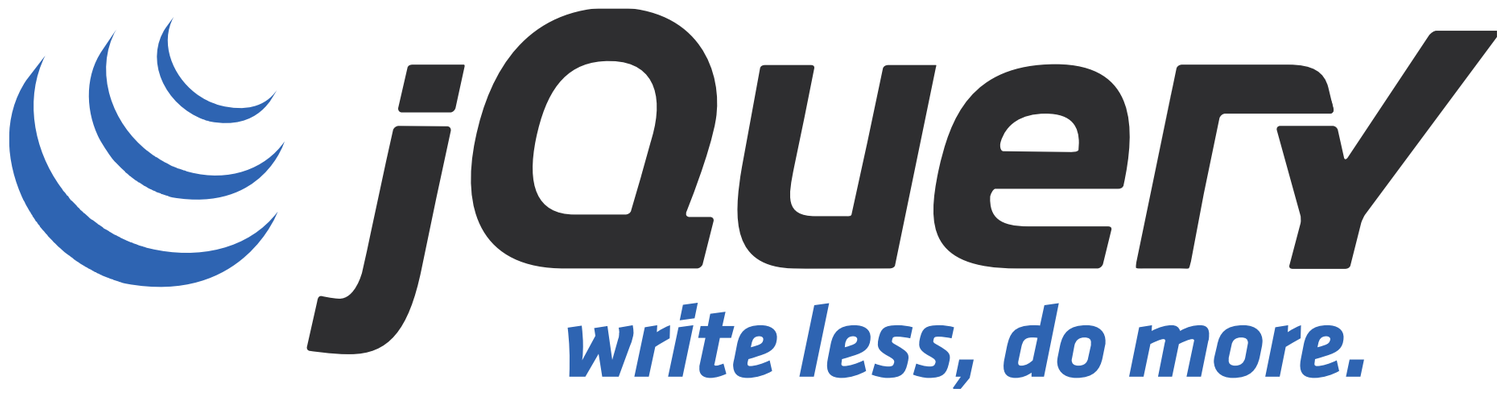
"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."

jQuery.com



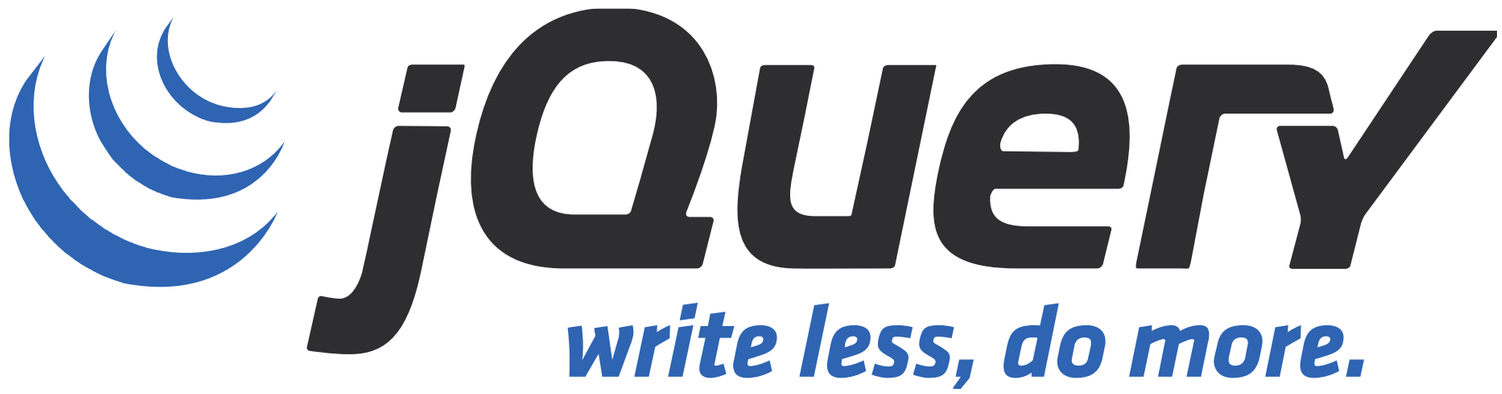
"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."

jQuery.com



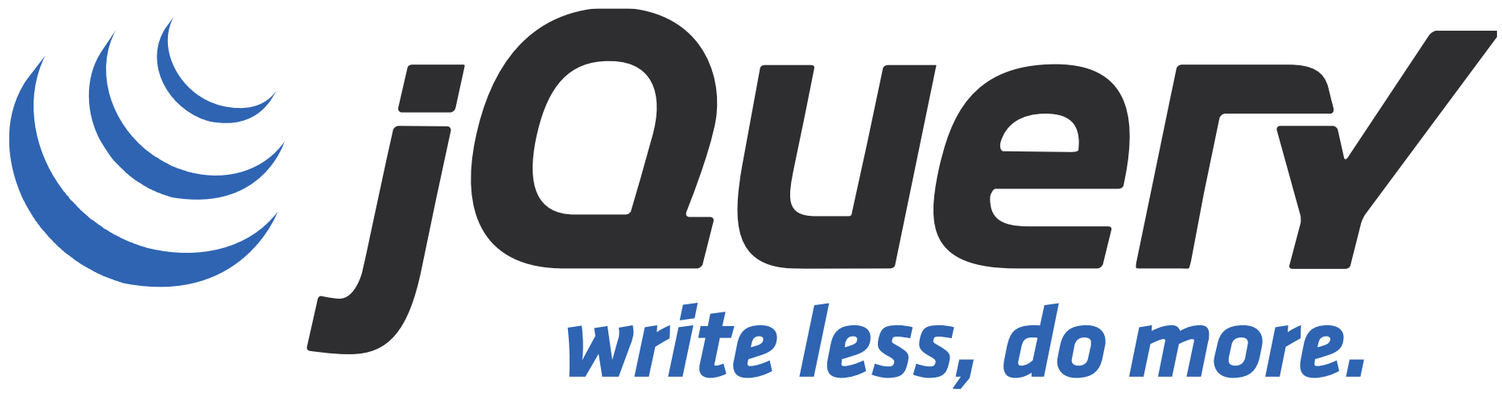
"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."

jQuery.com



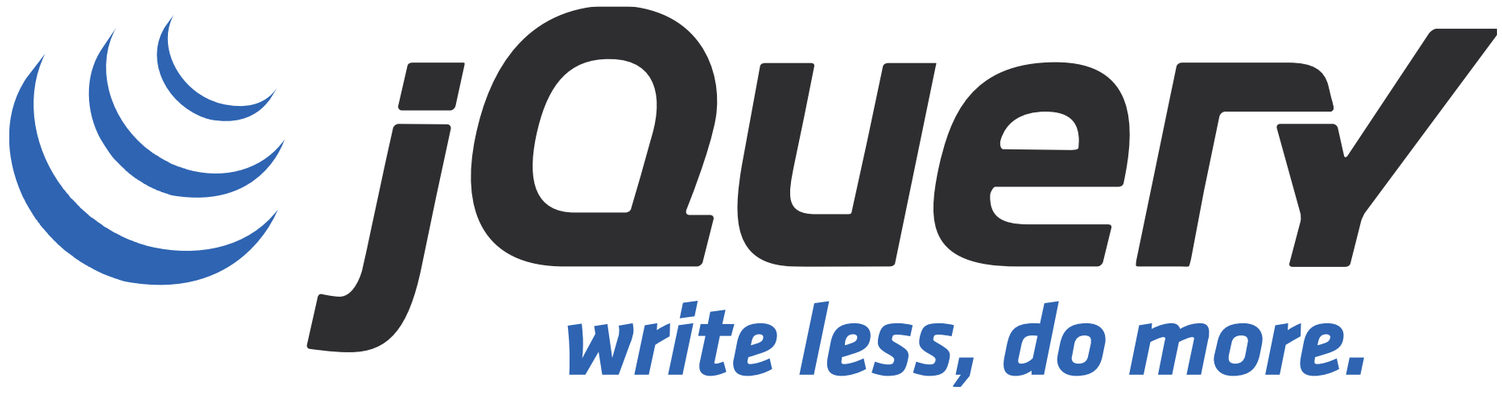
"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."

jQuery.com



"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."

jQuery.com



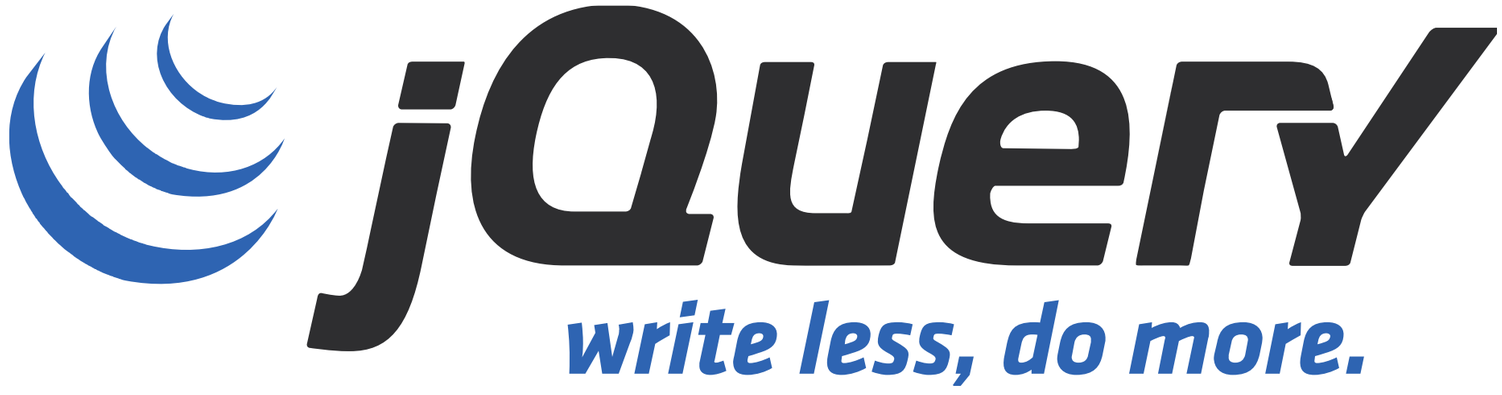
- January 14, 2006: John Resig released the first version of jQuery BarCampNYC
- August 26, 2006: jQuery v 1.0 Released
- April 18, 2013: jQuery 2.0 Released
- July 13, 2015: jQuery 3.0 Alpha/jQuery Compat 3.0 Alpha
- June 9, 2016: jQuery 3.0 Final Released
- September 22, 2016: jQuery 3.1.1 Released

Semantic Versioning 2.0.0

- Version numbering: `MAJOR.MINOR.PATCH`,
- `MAJOR` : incompatible API changes
- `MINOR` : add functionality in a backwards-compatible manner
- PATCH : backwards-compatible bug fixes.
- See http://semver.org/

jQuery 1.11.x vs 2.1.x vs 3.1.x

- jQuery 1.9+ and 2.0+ are the same for newer browsers.
- 2.0+ branch: gets rid of the extra code that provides support for legacy browsers (e.g. IE ⩽ 8)
- 3.0+ branch: code overhaul that contains some breaking changes (i.e. not backward compatible with 1.9+ and 2.0+)



- The most used JavaScript library on the web today. Just search for jQuery in the page source.
- W3 Tech Technology Survey

  Nov 2015: jQuery use on 97.5% of their tracked web domain names using JS, around 67% of all their tracked websites.

  Oct 2016: jQuery use on 96.3% of their tracked web domain names using JS, around 71.2% of all their tracked websites.

  http://w3techs.com/technologies/details/js-jquery/all/all

Learn more about jQuery

http://learn.jquery.com/

LINKÖPINGS UNIVERSITET

jQuery topics for today

- HTML document traversal
- Manipulating elements
- Creating elements
- Events in jQuery
- Animation using jQuery

LINKÖPINGS UNIVERSITET

Setting up jQuery

- Just like linking to a file containing your JavaScript
- Link to the jQuery library, or download and link to the jQuery library (available via several CDNs)
  - Example linking:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
```

Calling jQuery

- jQuery defines a function called `$` (which is also bound to the symbol `jQuery`)
- $ === jQuery
  - The `$` function is used to query the HTML page for a set of jQuery objects.
  - We then call methods associated with jQuery objects.

jQuery script loader

```jquery
$(document).ready(function() {
    // your code here
});

// ordinary JavaScript
window.document.onload = function() {
    // your code here
}
```

Retrieving elements using jQuery

- Use a **CSS selector** to have jQuery fetch a *result set* (a special jQuery array).
- The terms "*matched set*" and "*wrapper set*" are sometimes used instead of result set.
- The result set, technically **contains DOM elements**, each wrapped inside a jQuery object.
- A CSS selector can be combined with a filter.

Examples

```
$("li")                // fetch all the li elements
$("a")                 // fetch all the a elements
$(".thumbnail")        // fetch all element with class thumbnail
$("#container")        // fetch the element with the id container.
$("#menu .menuitem") // fetch all the elements with the class
                       // menuitem that are descendants of the
                       // element with the id="menu".
```

LINKÖPINGS UNIVERSITET

List of CSS selectors

selector	result set
`*`	all elements
`TagName`	all elements of a specific type
`.ClassName`	all elements with a specific class
`#idName`	all elements with a specific id
`TagA TagB`	all elements of type TagB that are descended from a tag of Type a
`TagA > TagB`	all elelemts of type TagB that are direct children of TagA
`TagA+TagB`	all elelemts that are children of TagA
`TagA~TagB`	all elements of type TagB that are proceeded by an element of TabA
`TagA:has(TagB)`	all elements tagA that has a TabB child element
`TagA.Classname`	all elelemts of type TagAs that also have the class Classname.
`TagA[attribute]`	all elements of type TagA that have the attribute
`TagA[attribute=value]`	all elelemts of type TagA that have the attribute with the value value
`TagA[attribute^=value]`	all elements of type TagA that have the attribute with a value that starts with value
`TagA[attribute$=value]`	all elements of type TagA that have the attribute with a value that ends with value
`TagA[attribute*=value]`	all elements of type TagA that have the attribute with a value that contains value

Examples

```
// get all div elements
var divs = $("div");

// get all elements with class "slide"
var slides = $(".slide");

// get all images that have the attribute "data"
var dataImgs = $("img[data]")
```

Example filters

selector	result set
`BaseSelector:first`	first element of result set
`BaseSelector:last`	last element of result set
`BaseSelector:even`	all elements with a specific class
`BaseSelector:odd`	all elements with a specific id
`BaseSelector:not(filterSelector)`	select elements that do not match the filterSelector
`BaseSelector:contains(text)`	select elements that contain the specified text
`BaseSelector:hidden`	select elements that are hidden
`BaseSelector:visible`	select elements that are visible

Events in jQuery

LINKÖPINGS UNIVERSITET

Pattern for handling events in jQuery

```
// select an element, access an event handler and
// set the function used to handle the event

// we can use a reference to a previously defined function
$(selector:filter).eventName(functionName);

// or provide an anonymously defined function
$(selector:filter).eventName(function() {
    // stuff to do
});
```

jQuery mouse event handlers

Event handler	Description
`dblclick`	double click
`mousedown`	mouse button pressed down
`mouseup`	mouse button released
`mouseover`	mouse is over an element
`mouseout`	mouse moves out from an element
`mousemove`	the mouse moves

jQuery document/window events

Event handler	Description
`load`	when the browser finishes downloading all files: HTML, images, CSS, JavaScript etc **Removed in jQuery 3.0**
`unload`	click a link to another page, close a tab, close window **Removed in jQuery 3.0**
`scroll`	drag the scroll bar, keyboard to scroll, mouse scroll wheel
`resize`	e.g. clicking the maximize button, dragging the resize handle

jQuery form events

Event handler	Description
`submit`	when a user presses the submit button
`reset`	when a user presses the form reset button
`change`	when a form element's value changes
`focus`	when a form element receives focus, e.g. when a text field becomes active
`blur`	when a form element looses focus

jQuery form events

Event handler	Description
`keypress`	when the user presses down a key, continues to fire while the key is down
`keydown`	when the user presses down a key, some browser do not fire continually, but some do
`keyup`	when the user releases the key

Special jQuery events

Event handler	Description
hover	handles both mouseover and mouseout
`toggle`	toggle between two functions, triggered by mouse click

Event handlers and the event object

```
bla.event(function(evt) {
    // evt is the event object
    var xPos = evt.pageX;              // dist from window left
    var yPos = evt.pageY;              // dist from window top
    var xPosAbs = evt.screenX;         // dist from screens left
    var yPosAbs = evt.screenY;         // dist from screens top
    var shiftPressed = evt.shiftKey; // true if shift was
                                       // pressed
    var keycode = evt.which;           // the keycode of the key pressed
    var trgElement = event.target;     // the element that was
                                       // the target of the event,
                                       // e.g. click()
});
```

Manipulating elements: attribute values and CSS

LINKÖPINGS UNIVERSITET

Template for demos

https://trinket.io/html/f1920df86c

LINKÖPINGS UNIVERSITET

Adding stuff to a page

```
var endString = "<p>Hello, I am some info, added at the end of the
element content.</p>";
var begString = "<p>Hello, I am some info, added to the beginning of
the element content</p>";

// Append something to the end of the infobox element
$("infobox").append(endString);
$("infobox").prepend(begString);
```

Removing and replacing content

```
// remove an element from the DOM
$("#message1").remove();

// replace an element in the DOM with another element
$("#message2").replaceWith("<p>Added to cart</p>");
```

Reading, changing and adding attributes

```
// read the value of an attribute
var imageSrc = $("#currentImage").attr("src");

// change the value of an attribute
var nextImage = "image2.jpg";
$("#currentImage").attr("src", nextImage);
```

```
// add an attribute to an element
$("#infobox").attr("style", "background-color: #888");

// remove an attribute + value from an element
$("#infobox").removeAttr("style");
```

Adding and removing a class

```
// add the class active to an element
$("li:first").addClass("active");

// remove the class active from an element
$("li:first").removeClass("active");
```

```
// add the class selected if it is not there
// remove the class selected if it is there
$("li:first").toggleClass("selected")
```

Reading and changing CSS properties

```
// read a css property value
var fontSize = $("h1:first").css("font-size");

// set a css property
$("h1:first").css("font-size", "12px");

// set multiple css properties
$("h1:first").css({"padding" : "4px",
                   "margin" : "50px"
                  });
```

Naming variables that will refer to jQuery objects

```
// If you use both regular JavaScript DOM object and
// jQuery objects in your code, it can be helpful to
// prefix you jQuery variable with a $ sign
$listObject = $("li");

// regular DOM element array
listItems = document.getElementsByTagName("li");
```

Implicit iteration over the result set

```
// jQuery implements implicit iteration which means that you can
// e.g. change edit css properties of all the objects in the
// result set without using a loop!

// select all li elements and set their css style (inline)
$("li").css("font-size", "12px");
```

Explicit iteration

```
// We can also iterate over the result set using explicitly using
// the $().each function. The $().each function can optionally
// pass an index and an element to the specified handler.

// select all li elements and set their css style (inline)
$("li").each(function(i, element) {
    $(element).css("font-size", "12px");
    console.log(i);
});
```

Explicit iteration

```
// The handler given to the $().each function does not have to
// use the index or element though. We can instead use the
// $(this) reference.

// select all li elements and set their css style (inline)
$("li").each(function() {
    $(this).css("font-size", "12px");
});
```

jQuery events and css

https://trinket.io/html/abe7a5c1c6

LINKÖPINGS UNIVERSITET

`$(this)` is not `this`

```
// $(this) is the jQuery object
// this - without surrounding $() is the DOM element

// apply css to p element in the result set
$("p").each(function() {
    $(this).css("color","#F00");
});

// using the DOM object
$("p").each(function() {
    this.style.color="#F00";
});
```

Animation in jQuery

Animation

- show, hide, toggle
- fadeIn, fadeOut, fadeToggle, fadeTo
- slideDown, slideUp, slideToggle
- general animate function

show, hide, toggle

```
// hide contentboxes
$("#button1").click(function() {
    $("div.contentbox").hide(1000);
});

// show contentboxes
$("#button2").click(function() {
    $("div.contentbox").show(1000);
});

// toggle visible state of contentboxes
$("#button3").click(function() {
    $("div.contentbox").toggle(1000);
});
```

fadeIn, fadeOut, fadeToggle, fadeTo

```
// fade out contentboxes
$("#button1").click(function() {
    $("div.contentbox").fadeOut(1000);
});

// fade in contentboxes
$("#button2").click(function() {
    $("div.contentbox").fadeIn(1000);
});

// toggle fade state of contentboxes
$("#button3").click(function() {
    $("div.contentbox").fadeToggle(1000);
});

// fade to a .50 opacity
$("#button3").click(function() {
    $("div.contentbox").fadeTo(1000, .50);
});
```

slideUp, slideDown, slideToggle

```
// slide up contentboxes
$("#button1").click(function() {
    $("div.contentbox").slideUp(1000);
});

// slide down contentboxes
$("#button2").click(function() {
    $("div.contentbox").slideDown(1000);
});

// toggle slide state of contentboxes
$("#button3").click(function() {
    $("div.contentbox").slideToggle(1000);
});
```

Chaining jQuery and animation

- Most jQuery functions return the result set.
- This means that is you want to do more than one thing on the same result set, you can chain these function calls together.

Chaining jQuery and animation

```
$("h1").css("font-size", "3em").css("font-size", "2em");
$("h1").animate({"font-size", "3em"}).css("font-size", "2em");
```

General animation of CSS changes using .animate()

```
// animate({css changes}, duration

$(".box").animate({
                    // css stuff
                    "width": '50px'
                  },
                  // duration of animation
                  1500
);
```

The animate() function - relative change

```
//relative change of css using += and -=
$(".box").animate({
                    // css stuff
                    'left': '+=50px'
                  },
                  // duration of animation
                  1500
);
```

Fading elements using a delay – setTimeout()

```
// setTimeout(function, delay)

function fadeAll(selector) {
    $(selector).each(function(i, element) {
        $(element).css({'opacity':1});

        // delay the animation
        setTimeout(function() {
                // animate opacity change
                $(element).animate( { "opacity": 0.0 }, 450);
            },
        // each element is delayed by 500ms + 500ms * position in list
        500 + ( i * 500 ));
    });
}
```

Halting all animation

```
$(selector).stop();
```

jQuery events and animation

https://trinket.io/html/860fb6f806

LINKÖPINGS UNIVERSITET

Describing interaction

- Words are not enough. We need to use visual descriptions.
- Images are not enough. Interaction is movement.

  - Wireframes
  - State transition diagrams

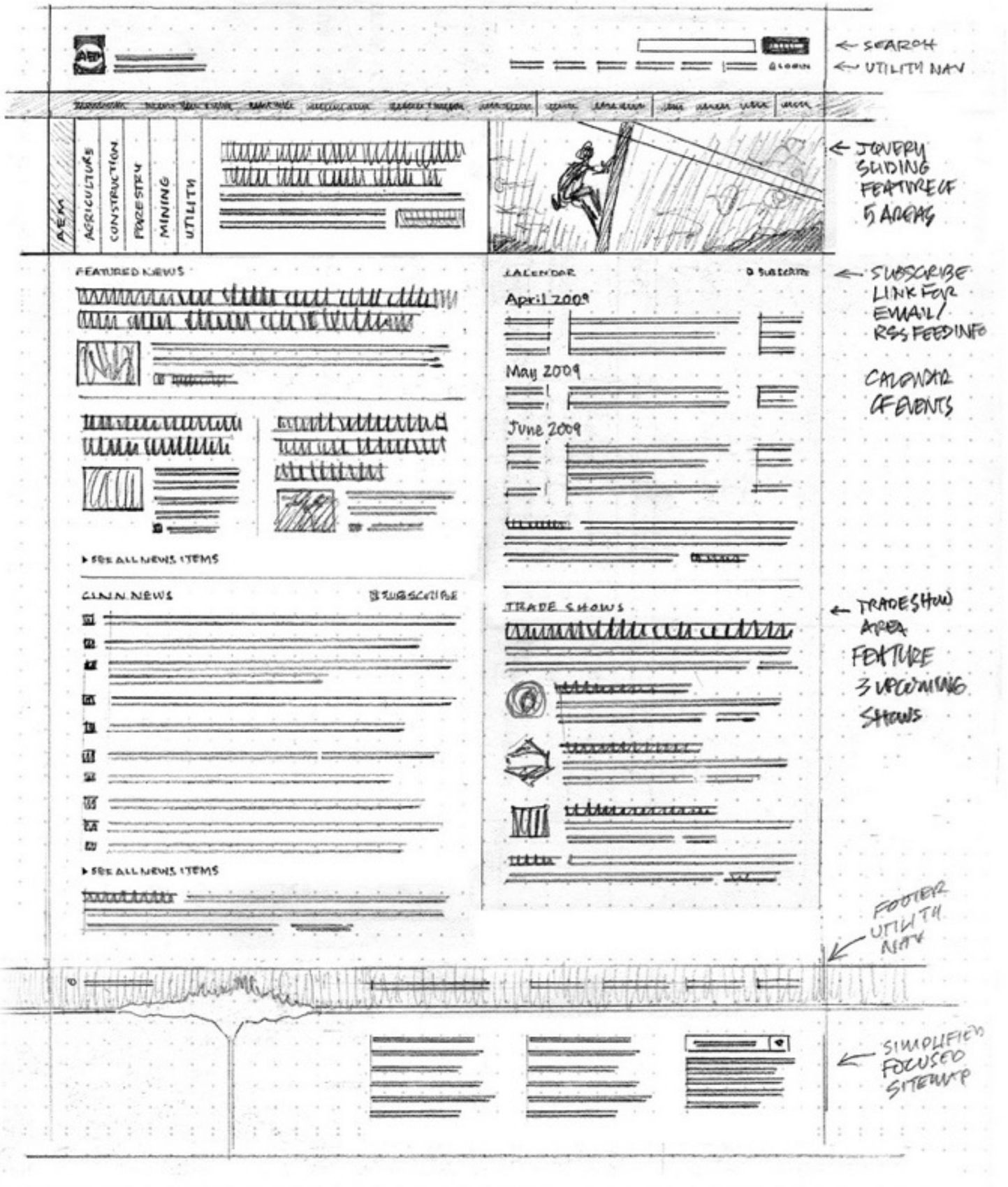
Wireframes

Paper prototyping

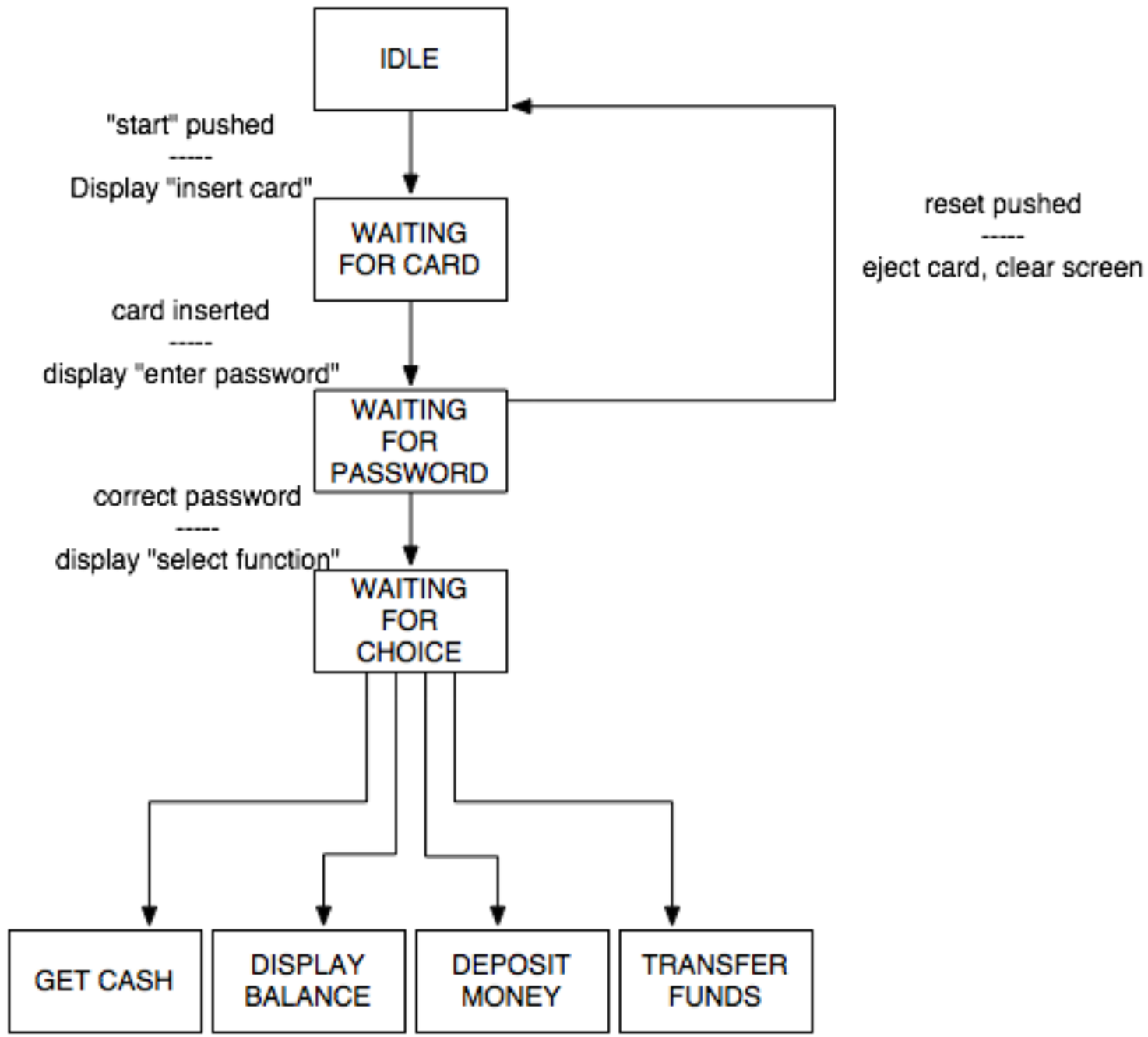
Plan layout for content

LINKÖPINGS UNIVERSITET

AEM MAINPAGE CONCEPT v|C



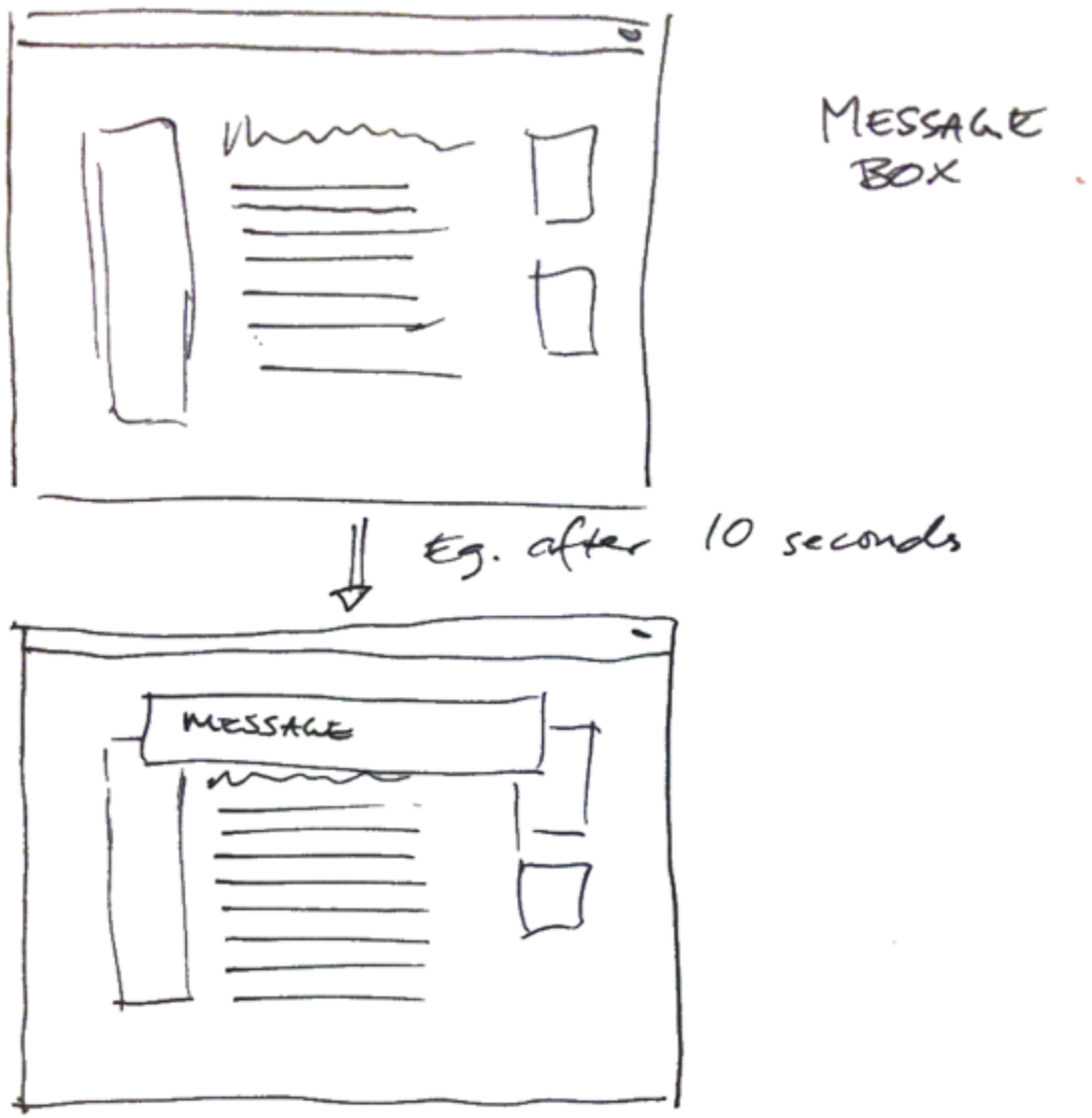
State transition diagrams



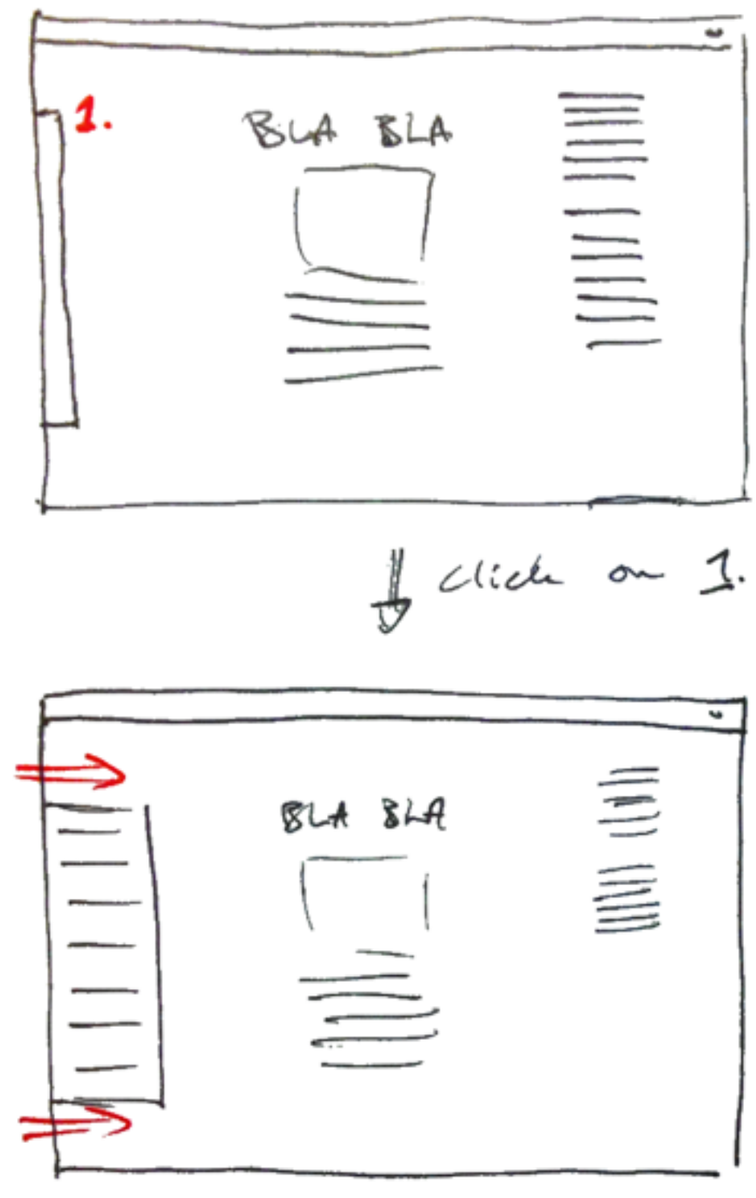
For describing interaction,
we can combine these two

LINKÖPINGS UNIVERSITET

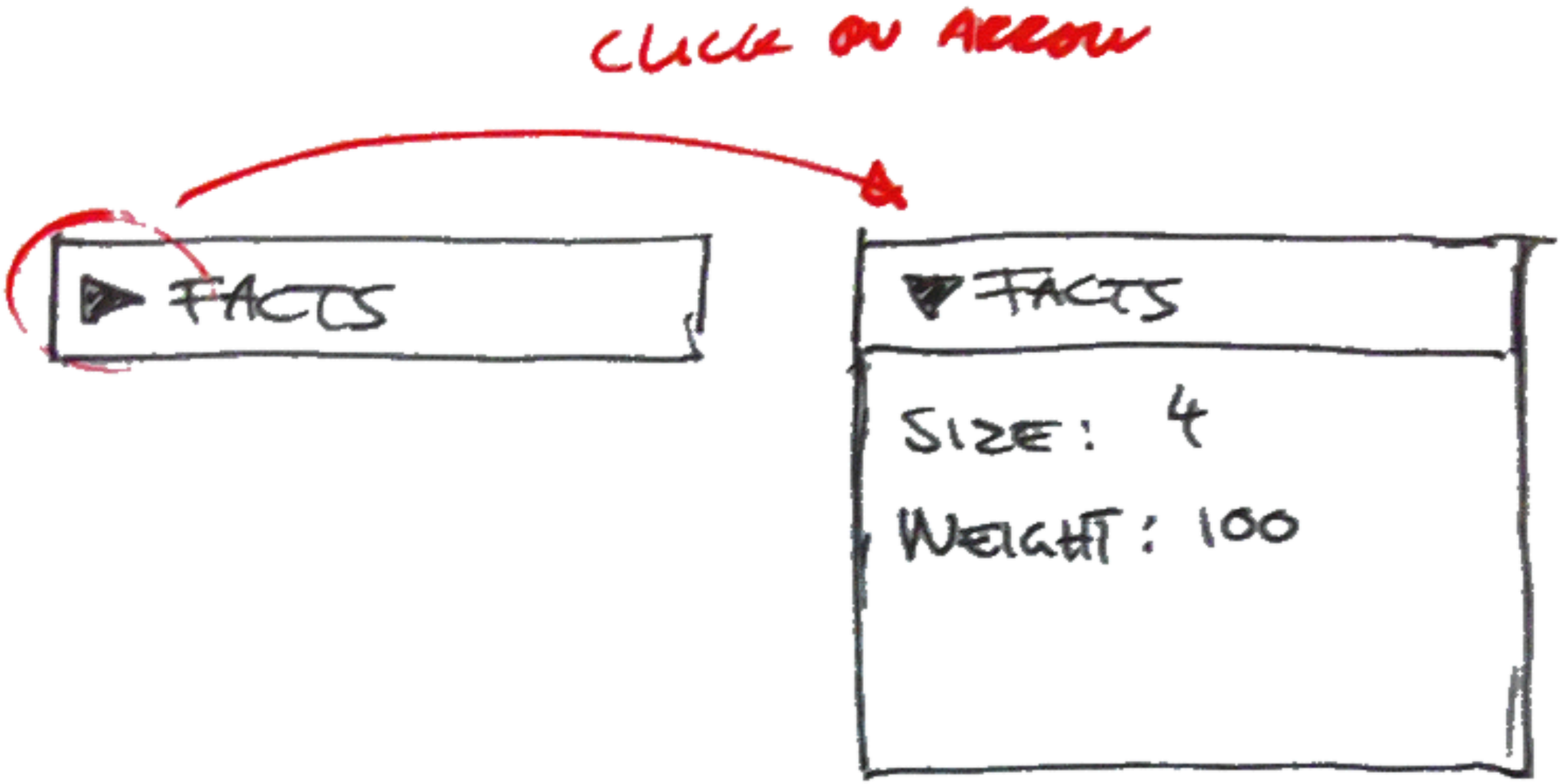
Showing a message box



Popping out a menu bar



Showing and hiding text



Adding things to a shopping list