jQuery

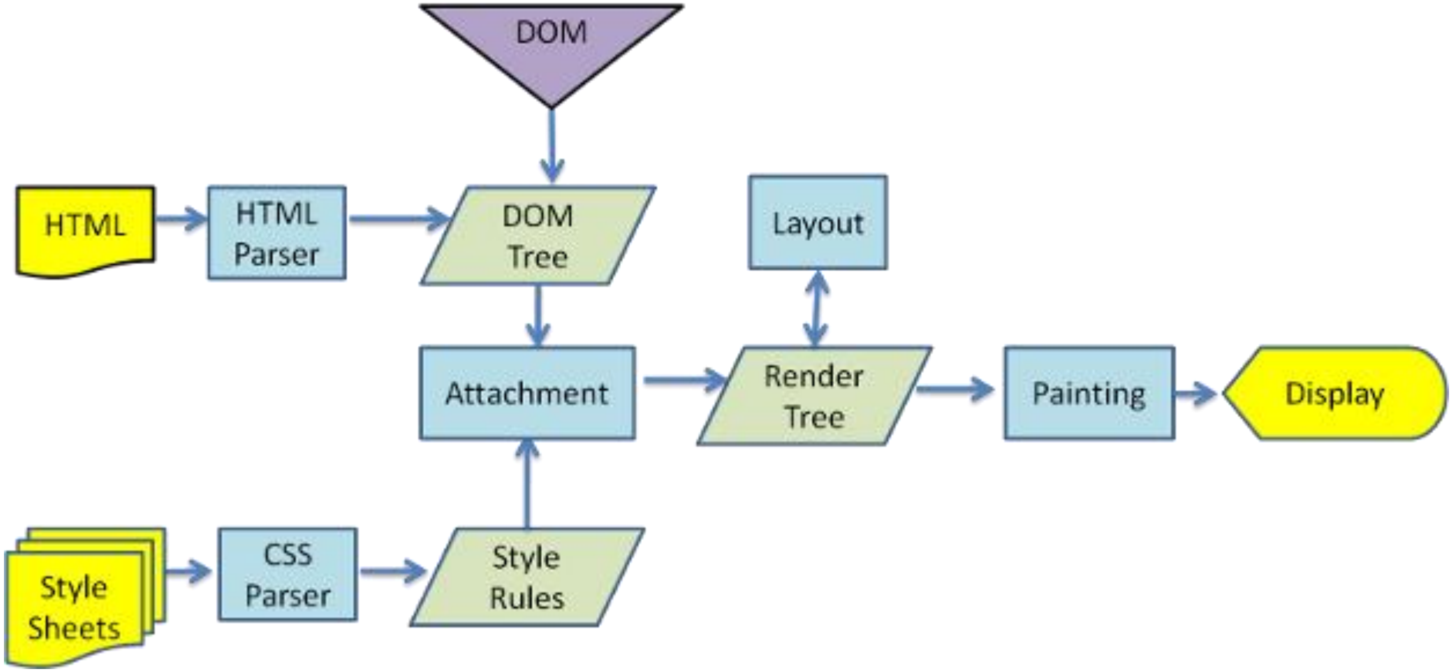
-Write Less, Do More

By morriswmz

Some knowledge about DOM & JavaScript

PREREQUISITES

A Glimpse of the Workflow



- A typical workflow of webkit. Learn more:
  - http://www.iefans.net/liulanqi-ruhe-gongzuo-yuanli/

What is DOM?

- The Document Object Model (DOM) is a cross-platform and language-independent convention for **representing** and **interacting** with **objects** in HTML, XHTML and XML documents.

- It’s now a common accepted **standard**. So no need to worry about compatibility.

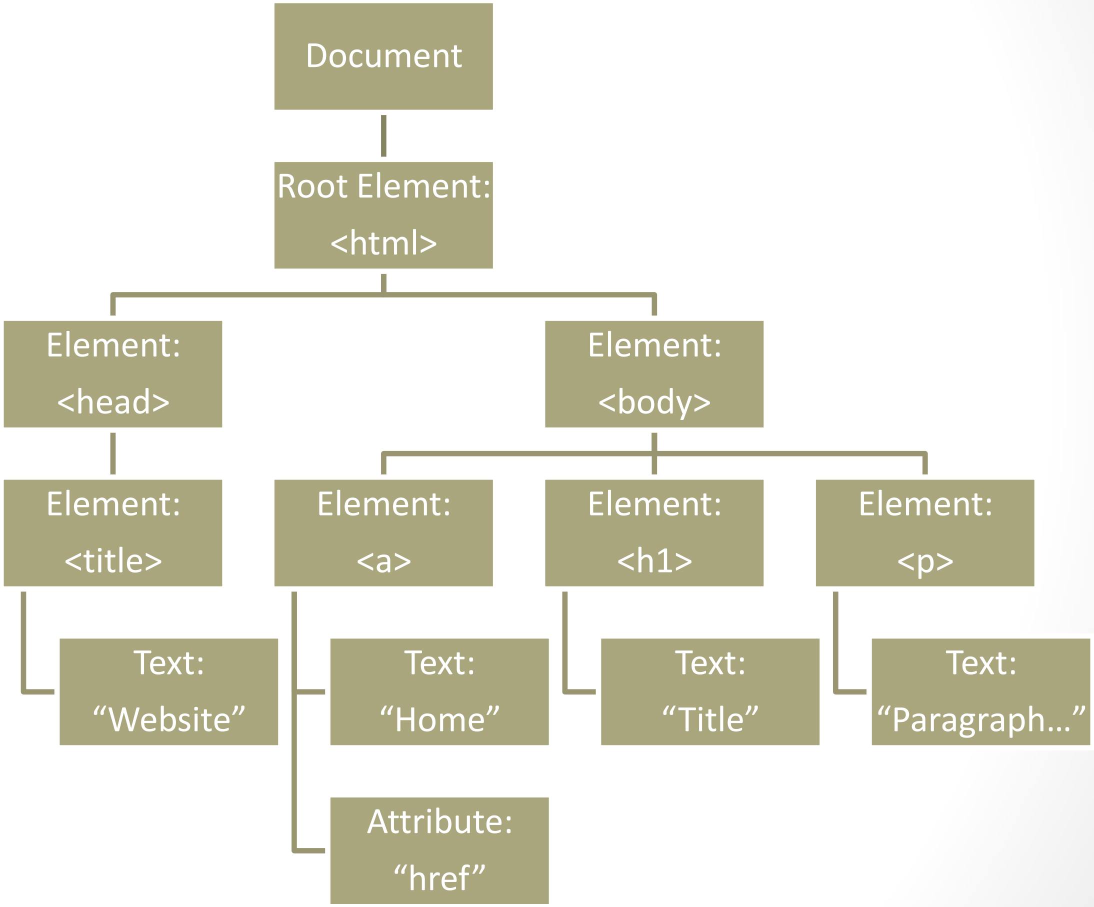
- It provides methods for adding, moving, modifying and deleting HTML elements or changing its attributes.

- DOM is accessible to JavaScript.

HTML DOM Node Tree

- The HTML DOM views an HTML document as a tree-structure. The tree structure is called a node-tree.

- Actually, HTML is structured like a tree.



Document

Root Element: <html>

Element: <head>

Element: <body>

Element: <title>

Element: <a>

Element: <h1>

Element: <p>

Text: “Website”

Text: “Home”

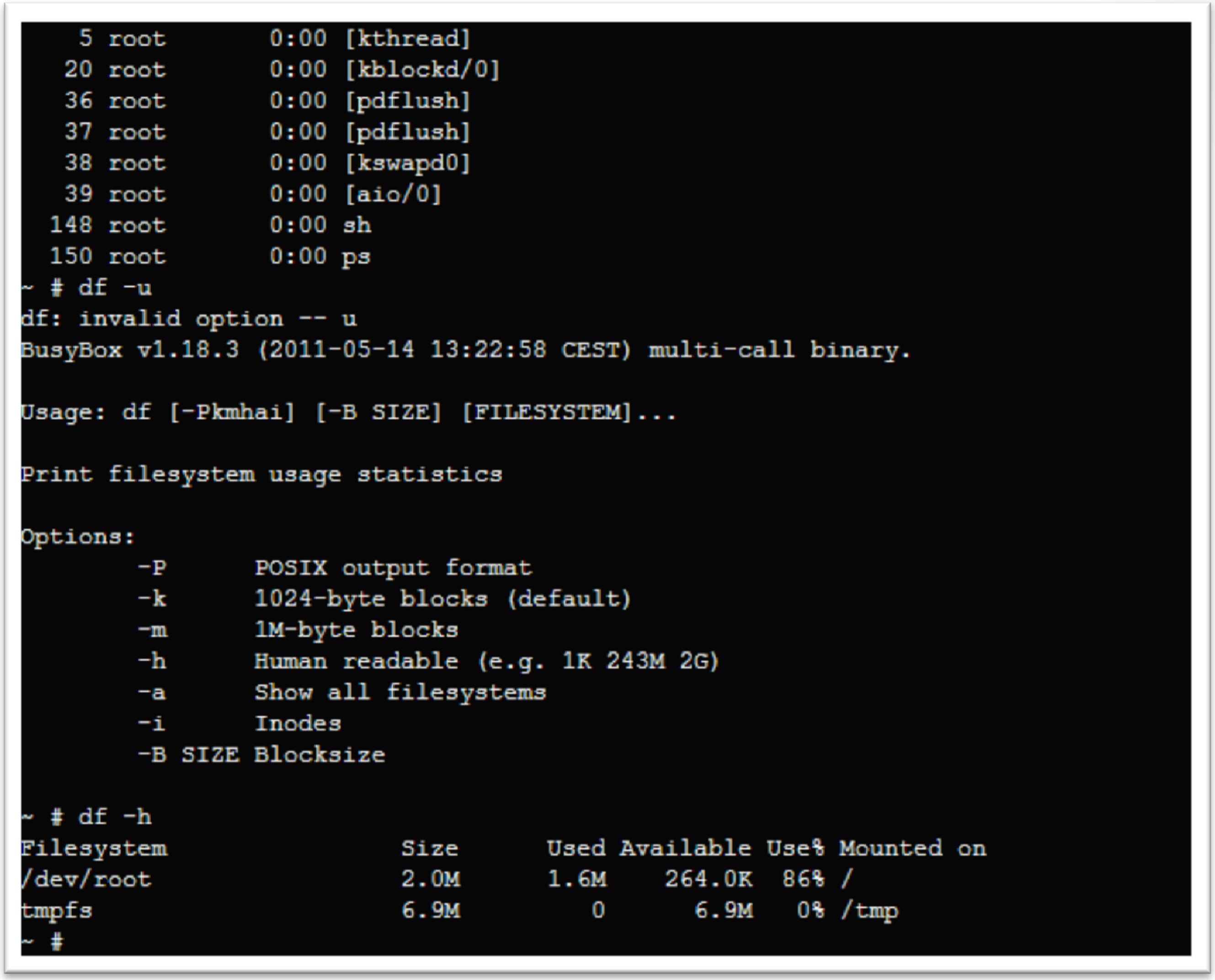
Text: “Title”

Text: “Paragraph…”

Attribute: “href”

So, JavaScript?

- Real Name : ECMAScript
- Running in all major browsers, almost on every website.
- Capable of:
  - Modifying HTML Structure
  - Generating Dynamic Content
  - Animation, Canvas Rendering, WebGL
  - Form Validation
  - Cookies, Local Storage
  - Maintaining Cross-browser Compatibility
  - …
- You can even create a web-based OS using JavaScript.
- JavaScript is CASE SENSITIVE!

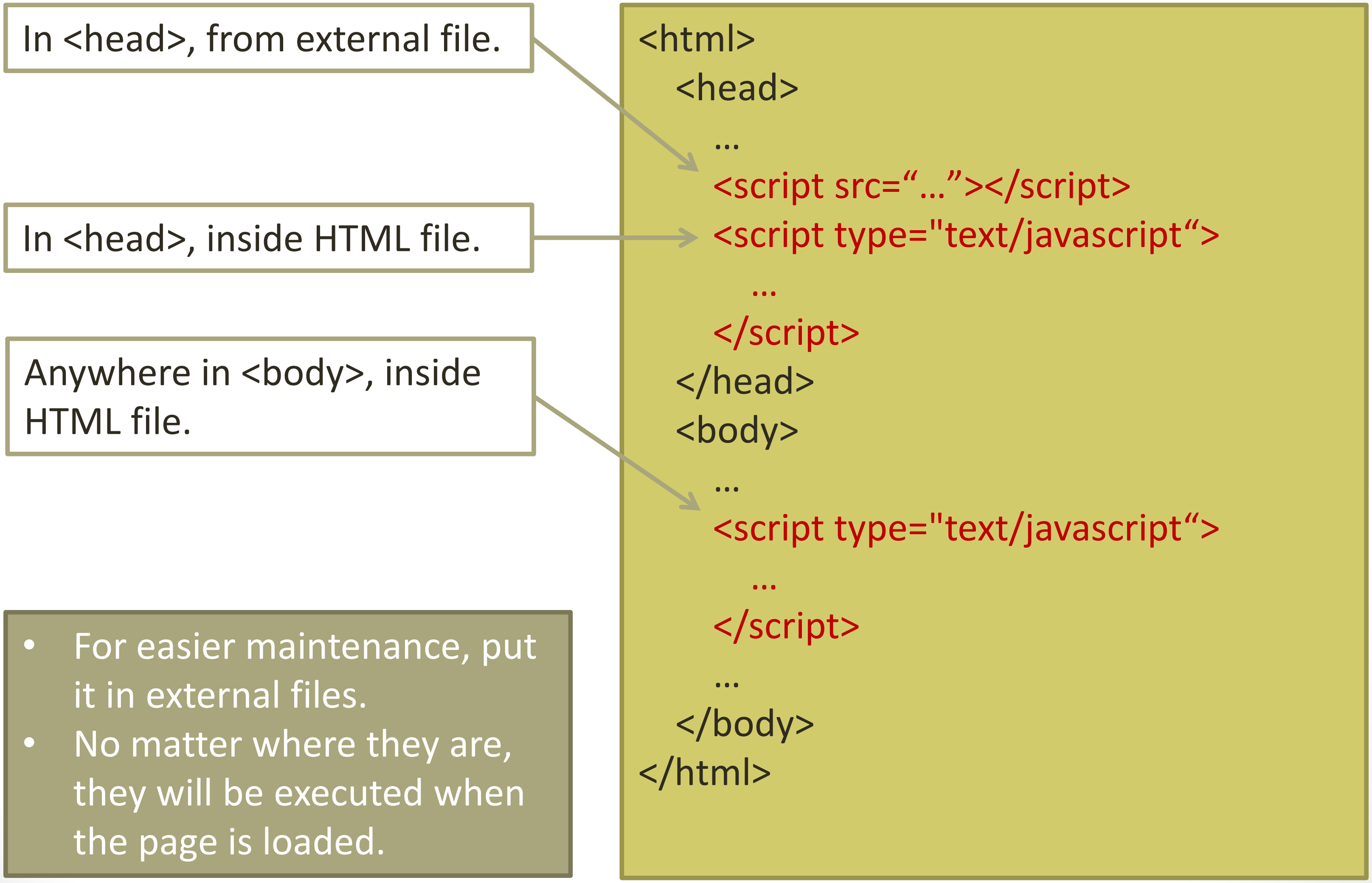


```
    5 root       0:00 [kthread]
   20 root       0:00 [kblockd/0]
   36 root       0:00 [pdflush]
   37 root       0:00 [pdflush]
   38 root       0:00 [kswapd0]
   39 root       0:00 [aio/0]
  148 root       0:00 sh
  150 root       0:00 ps
~ # df -u
df: invalid option -- u
BusyBox v1.18.3 (2011-05-14 13:22:58 CEST) multi-call binary.

Usage: df [-Pkmhai] [-B SIZE] [FILESYSTEM]...

Print filesystem usage statistics

Options:
        -P      POSIX output format
        -k      1024-byte blocks (default)
        -m      1M-byte blocks
        -h      Human readable (e.g. 1K 243M 2G)
        -a      Show all filesystems
        -i      Inodes
        -B SIZE Blocksize

~ # df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                 2.0M      1.6M    264.0K  86% /
tmpfs                     6.9M         0      6.9M   0% /tmp
~ #
```

A pc emulator written in JavaScript

Checkout: http://bellard.org/jslinux/

Where to put it?



In <head>, from external file.

In <head>, inside HTML file.

Anywhere in <body>, inside HTML file.

```
<html>
  <head>
    …
    <script src="…"></script>
    <script type="text/javascript">
      …
    </script>
  </head>
  <body>
    …
    <script type="text/javascript">
      …
    </script>
    …
  </body>
</html>
```

- For easier maintenance, put it in external files.
- No matter where they are, they will be executed when the page is loaded.

Variables

- No strict type definition
  - var a = 12; // 12
  - var b = “12”; // “12”
  - var c = a+b; // “1212”
- Operations
  - + - * / % ++ -- += -= /= *= …
- Comparison
  - > < == != && ||
  - === // “5” == 5 is true but “5” === 5 is false
- Array
  - var list = new Array(“list1”,”list2”,”list3”);
  - var list = [“list1”,”list2”,”list3”];

Somewhat C-like Syntax

```
if (a == b) {
    // do sth
} else if (a > b) {
    // do sth else
} else {
    // do sth
}
```

```
function m(a,b) {
    c = a*b;
    return c;
}
```

```
while (a<10) {
    // do sth
}
```

```
for (i=0;i<10;1++) {
    // do sth
    if (cond1) {
        break;
    }
}
```

```
for (key in dictionary) {
    // do sth
    if (cond2) {
        continue;
    }
}
```

```
switch (n) {
    case 0:
        // do sth
        break;
    case 0:
        // do sth
        break;
    default:
        // do sth
}
```

Events

- A certain event will be triggered under certain condition. JavaScript allows our webpage to be able to respond to these events. Without events, there will be no interaction.

- Event handler can be assigned in HTML:

```
<input type="text" size="30" id="email" onchange="checkEmail()">
```

- checkEmail() function will be called when the text input is changed.

Events

Name	Trigger Condition
onblur	An Element lost focus
onchange	The content of a field changes
onclick	Mouse clicks an object
ondblclick	Mouse double-clicks an object
onerror	An error occurs when loading a document or an image
onfocus	An Element gained focus
onkeydown	A keyboard key is pressed
onkeypress	A keyboard key is pressed or held down
onkeyup	A keyboard key is released
onload	A page or image is finished loading

Events

Name	Trigger Condition
onmousedown	A mouse button is pressed
onmousemove	The mouse is moved
onmouseout	The mouse is moved off an element
onmouseover	The mouse is moved over an element
onmouseup	A mouse button is released
onresize	A window or frame is resized
onselect	Text is selected
onsubmit	Form is submitted
onunload	The user exits the page

A function inside a function?

- This may sound strange. However, you can treat a function like a variable type in JavaScript.
- You can define function inside function.
- Functions can also be arguments.

```
var myfunc = function () { document.write(“abc”); };

$(document).ready(myfunc);

$(“#nav_btn”).click(function () {
    function a() { … }
    …
});
```

You will love it.

JQUERY FOR DUMMIES

The Magic $

- To use jQuery, you should download it from:
  - http://jquery.com/
- And include it in your HTML page like this:
  - <script src=“…/js/jquery/1.4.2/jquery.min.js"></script>
- Then you can use the magic $, the main function of jQuery.

- Actuall $(…) is equal to jQuery(…), if you are not using other JavaScript frameworks, you can simply use $(…) instead of jQuery(…).

Powerful Selectors

- You can select a HTML element or a group of HTML elements using the same selectors as in CSS. Forget things like getElementByID() …
  - $("body")
  - $("#nav a")
  - $(".photo > p")
  - $("#article:first-child")
  - $("#a, #b, #c")
  - …

Get Attributes

- attr(attr_name)
  - Get the value of an attribute for the first element in the set of matched elements.

```
$(“#logo_img”).attr(“src”); // get the src value of #logo_img
```

- attr(attr_name, value)
- attr(properties)
  - Set one or more attributes for the set of matched elements.

```
// get the src value of #logo_img
$(“#logo_img”).attr(“src”,”/images/new_logo.jpg”);
// change attributes of #logo_img
$(“#logo_img”).attr({“src”:”/images/new_logo.jpg”, “alt”:”logo”});
```

Change Style

- css(properties)
  - Set one or more CSS properties.

```
$(“#nav”).css({“margin-left”:”20px”, color:”#ccc”});
```

- addClass(class_name)
- removeClass(class_name)
- toggleClass(class_name)

```
$(“#nav_links”).addClass(“selected”);
$(“#nav_links”).removeClass(“selected cleared”);
$(“#nav_links”).toggleClass(“hightlight”);
```

Modify HTML

- append(content), appendTo(content)
  - $(“#container”).append(“<p>child</p>”);
  - $(“<p>child</p>”).appendTo($(“#container”));
- prepend(content), prependTo(content)
  - $(“#container”).prepend(“<p>child</p>”);
  - $(“<p>child</p>”).prependTo($(“#container”));
- after(content), insertAfter(content)
  - $(“#box”).after(“<p>same level box</p>”);
  - $(“<p>same level box</p>”).insertAfter($(“#box”));
- before(content), insertBefore(content)

  …

Modify HTML

- more to learn:
  - wrap()
    - // wrap an element
    - $(“img").wrap("<div class='wrap'></div>");
  - unwrap()
    - // remove wrapper
    - $(“img").unwrap();
  - remove()
  - replaceAll()
  - …

Filters

- eq(index)
  - // change the second <li>’s text-color to black
  - $(“#nav ul li”).eq(1).css(“color”,”#000”);

- first()
- last()
  - $(“#main p”).last().addClass(“last”);

- has(expr)
  - // add class “first” to <li>s which has <ul>
  - $(“li”).has(“ul”).addClass(“first”);

Respond to Events

- $(document).ready(function () { // do sth });
  - Write like this if you want to do something when the page is loaded (DOM loaded successfully).

- $("#search_btn").click(function() { // do sth });
  - Do sth. when the element matching "#search_bth" is clicked.

- $(".menu_btn").hover(function() { // do sth });
  - Do sth. when mouse over elements matching ".menu_btn".

- Checkout other events yourself.
- Also checkout bind() and unbind().

Simple Effects

- show(speed, callbackFunc)
- hide(speed, callbackFunc)

  - $(“a").show("fast",function() {
  -     $(this).text("Animation Done!");
  - });

- fadeIn(speed, callbackFunc)
- fadeOut(speed, callbackFunc)
- …

Learn more at:

HTTP://JQUERY.ORG.CN/MANUAL/