**Get started**

To embed jQuery in your project, download source code from <https://jquery.com/download/> or include jQuery CDN in the head of html file.

**Selectors $()**

In addition to similar CSS selectors, jQuery selectors include:

* index selectors: :eq(index), :lt(index), :gt(index)

$(“li:lt(3)”) /\*Selects <li> elements with index 0, 1, 2\*/

* content selectors: :contains(text), :has(selector), :empty, :parent

$(“p:contains(hello)”) /\*Selects <p> elements that contain text “hello”\*/

$(“p:has(button)”) /\*Selects all <p> elements that contain <button>\*/

* form selectors type(:button, :file, :input, etc)/status(:selected, :checked, :disabled, :enabled, :focus)

$(":button") /\*Select all <button> elements and elements of type button\*/

**Traversing $(selector).method([param])**

In addition to methods shown in slides, jQuery traversal methods include:

* **Ancestors**
  + .closest(selector [, context])

$(“div#selected”).closest(“div”) /\*The closest <div> ancestor (including itself) of <div> element with id “selected”.\*/

* + .offsetParent()

$("p").offsetParent() /\*The closest ancestor of <p> elements that have a position attribute\*/

* **Filtering**
  + is(selector) returns true, false

$(“#button”).parent().is(“div”) /\*True if the parent of the element with id “button” is a <div> element; otherwise, false.\*/

* + .not(selector)

$(“p”).not(“.test”) /\*All <p> elements that do not have a clast “test”\*/

* + .slice(start\_index [, end\_index])

$(“li”).slice(0, 3) /\*The first three <li> elements\*/

* **.**add(selector [, context])

$(“p”).add(“button”) /\*All <p> and <button> elements\*/

**CSS/HTML Manipulation**

**Structure Manipulation**

|  |  |  |  |
| --- | --- | --- | --- |
| Get | Update | Add | Remove |
| .text() | .text(new\_text) | .append(content) | .remove(filter) |
| .html() | .html(new\_html) | .prepend(content) | .detach(filter) |
| .val() | .val(new\_val) | .after(content) | .empty() |
| .attr(attr\_name) | .attr(attr\_name, new\_value) | .before(content) | .removeAttr(attr\_name) |
|  |  | .wrap(wrapper) | .unwrap() |

**Style Manipulation**

|  |  |  |
| --- | --- | --- |
| Class | Style | Dimension (get or set) |
| .hasClass(class\_name) | .css(property\_name) | .width()  .height(400) |
| .addClass(class\_names)  .addClass(“first second”) | .css(property\_name, value) | .innerWidth()  .innerHeight() |
| .removeClass(class\_names) | .css(property\_object)  .css({backgroundColor:”red”, ... }) | .outerWidth()  .outerHeight() |
| .toggleClass(class\_names) |  | .outerWidth(true)  .outerHeight(true) |

**Event & Animation**

* jQ Event
  + Properties: .target / .pageX / .pageY / .which
  + Create an Event
    - var e = $.Event(“click”); //Same as jQuery.Event(“click”)
    - var e = $.Event(“keydown”, {keyCode: 64} )
  + Trigger an Event: $(“body”).trigger(e);
  + Add/Remove Listener
    - $(“selector”).eventName(function);
    - $("selector").on("eventName", function);
    - $(“selector”).off([“eventName”, handler]);
    - eventName example: click, keypress, mouseenter
* Effect and Animation
  + Effect
    - e.g. $("selector").fadeIn([speed, callback]);
    - Toggle: If element currently shown, execute the effect to hide element; vise-versa
    - effect list: show/hide, fadeIn/fadeOut/fadeToggle/fadeTo, slideDown/slideUp/slideToggle
  + Animation
    - $("selector").animate({CSSparameters} [, speed, callback]);
    - Stop Animation: $("selector").stop([stopAll, goToEnd]);

**AJAX**

|  |  |
| --- | --- |
| Helper | Core |
| $.get(url [,payload, success, dataType]) | $.ajax({  url: “server/location”  }) |
| $.post(url [,payload, success, dataType]) | $.ajax().done(function(data){ })  $.ajax().fail(function(xhr, status, error){ })  $.ajax().always(function(xhr, status){ }) |
| $(selector).load(url [, payload, success]) |  |