

## # jQuery

→ In very short it simplify javascript programming.

→ It is Javascript library.

→ It make Dom manipulations easy.

→ powerful, easy to use, functional.

⇒ use jQuery minimum from CDN easy to use & lightweight

⇒ use google based CDN jQuery

visit  
api.jquery.com & Read  
DOCS

④ Use jQuery in console to check that is jQuery added successfully or not

• \$ ⇒ selector in jQuery you don't have to write commands like  
document.querySelector("") in jQuery  
You just have to use

`$("li").css('color', value)`

selector      key      value

only this small code is enough to manipulate text

↳ if multiple entity exist it will return a whole list for it

⇒ you don't have to choose each element you just have app  
it

⇒ Basically \$( ), \$ in jQuery means `document.querySelectorAll()`

⇒ so for class ⇒ ".className"

for id ⇒ "#idName"

## # Styling in jQuery

⑤ If you going to use some library and a external JS file so always keep in mind that script src should be down the position that of library

⇒ Also with jQuery you can send multiple property at once using object Ex

`$("li").css({ color: "blue", background: "blue", border: "1px solid black", fontSize: "26px" })`

Such like that

⇒ It also made it very simple to use jQuery

② change content in html using jquery

`$(".li").text()` ⇒ So it will provide text from all elements in list.

⇒ If you apply directly `$('li')` ⇒ all will change (avoid loops)

⇒ If you want to change single you can go for index that will be easy

· `text()` this only accept text & all thing inside it take them as text

⇒ `$(".li").text("hello <br> world")` → get printed

⇒ If you want to pass html

⇒ `$(".li"); html("hello <br> world")`

⇒ `$(".li").append("</>")` → to treat as HTML  
↑ add content at last

⇒ `$(".li").prepend("<content>")` ⇒ prepend content treat internal content as html

⇒ append and prepend done on each element which we get from list i.e. `($(".li"))` ⇒ output is [ ]

- (#) attributes.

functions: ① addClass() ② attr() ③ hasClass()

④ html() ⑤ prop() → get value

↑  
insert html

text in code.

of prop → get value

of attribute → get value

Prepend &  
append &  
apply  
on each list element

→ boolean  
o/p

## ② CSS functions

(camel case)

`.css()` ⇒ add style you can give object also

# You can also chain jquery

`$(“h1”).addClass(“hell”).hasClass(“hell”).css(“color”: red”)`

→ multiple functions used like a chain in jquery

⇒ If you fetch single element in `$(“h1”)[0]` so now you can also use dom operations too

# Image & form attribute

Say you have input bar. `val()` function is used to get value from input box.

`$(“input”).val()` ⇒ to get value

⇒ so take image from placeholder, take input & change content of image.

# <https://via.placeholder.com/150x900.png?text=Some+text>

⇒ `$(“input”).val(“hello world”)`

↑ this will change value

⇒ props this will also give value of input

Ex `v.prop(“name”)` ⇒ ‘txt’

`v.prop(“placeholder”)` ⇒ ‘enter text’

⇒ Prop & attr nearly work same

## # JQuery click event

In jquery you have direct function which you can use click

```
$(“hi”).click(function () {  
    console.log (“clicked”);  
});
```

that it

- In jquery you don't have to run loop explicitly to each element, you just have to jquery function
- ⇒ also inside functions you can do many things
- ⇒ as we use this in java jquery you have to wrap it in  $\$(this)$  and you can use this easily → this function automatically
- ⇒ You can addClass("red") work like toggle

### ④ document ready event listener

```
 $\$(document).ready(function() {$ 
    console.log("loaded & ready to go")
 $}$ 
```

- ⇒ If document is loaded & dom is ready to use at that time "document ready event will get invoked"
- ⇒ do search on (jquery.gpi) & get info regarding it

$\$(\text{"selector"}).hover(function() {})$

→ event invoked when mouse is hovering on tag.

### ⑤ More mouse Events

① mouseenter() ⇒ apply when mouse click on etc

② mouseleave ⇒ opposite to hover

hover + mouseleave → good visual feel.

⇒ Search event in jquery

## # Keyboard Events

→ keydown() ⇒ when key is being pressed

→ keyup() ⇒ when key is being unpressed

→ keypress() ⇒ key interaction

order of execution    keydown → keypress → keyup

## # On Event

this will provide a extra block to specify event

`$(selector).on ("event keyword", function())`

| - code - |

});

⇒ you have see event keywords online

Ex : click , keypress , mouseenter , mouseover , hover etc

⇒ with on keyword we can create any type of event listener

## # Apply various effect with jQuery (show, hide, toggle)

`$(selector).on ("click", function())` → lets apply effect after triggering some animation effect

`$("selector").on("click")`

`$(this).show()` ⇒ do not do anything

`$(this).hide()` → hide that tag & not reserve space

`$(this).toggle()` → hide & show element

## # fade effects

functions = [ fadeIn()    fadeInOut()    fadeToggle() ]

→ three main effects related to fade or use

in bracket you can pass time in ms  
keywords like "slow" fast

Ex : fadeIn(3000) , fadeOut(3000) , fadeToggle("slow")

fadeOut  $\Rightarrow$  fade and go to hidden state.

also  $\Rightarrow$  fadeIn, fade out, fadeToggle has functions

so after seeing effect we can do some task also

Ex `fadeIn(3000, function () {`

such like that `}); console.log("Fade In Complete");`

$\Rightarrow$  If element is already visible fadeIn, show() not do anything.

$\Rightarrow$  While fading we can able to adjust opacity value  
fadeTo function used for this

`* fadeTo(time, opacityValue)`

### # Slide in jquery

slideUp(), slideDown(), slideToggle()  $\Rightarrow$  can use anything

$\Rightarrow$  slide function also came with additional function parameters which we can use to do some action

$\Rightarrow$  You pass time parameters to slide

`$("#th").slideUp(3000, function(e) {`

`console(e)`

`console("slideup")`

`}`

You can manage speed of slideup, slidetdown & slideToggle

### # JQuery Animate

$\rightarrow$  position should be absolute

`bit{`

`position: absolute`

`|| Styling ||`

(by default static but in animation movement is important)

$\Rightarrow$  Such like that

## # Animate options

animate()  $\Rightarrow$  this function provides custom animations

```
$("h1").animate({  
    left: '300px'  
    width: '100px'  
})
```

If you do :- width : "+=20"  $\rightarrow$  at each time event happens size of width increase by 20.

$\Rightarrow$  search for animate at jquery website

## [#] AJAX jquery

$\rightarrow$  with jquery we can pass AJAX requests.

$\rightarrow$  these AJAX requests are asynchronous, so we need to wait for response

```
Ex: $get(url, function(response){  
    console.log(response)  
})
```

$\Rightarrow$  and after getting request we can able to do some actions on it and make it useful.

## (#) # load & cors errors (#)

load("")  $\Rightarrow$  it is used to load local files and you will get some error.

$\Rightarrow$  so use localhost to load it.  $\Rightarrow$  load so

$\Rightarrow$  run file with some web server and you will have http:// address for a file

(#) So you can load another file in a file using load

```
| $("selector").load("address of file to be  
loaded") |
```