↳ Traversing

→ It means moving through/over the HTML elements to find/filter/select particular or entire element.

↳ •children():

→ It returns all direct children of selected element.

↳ finds all child element related to selected element.

↳ This method only traverse single level down the DOM tree

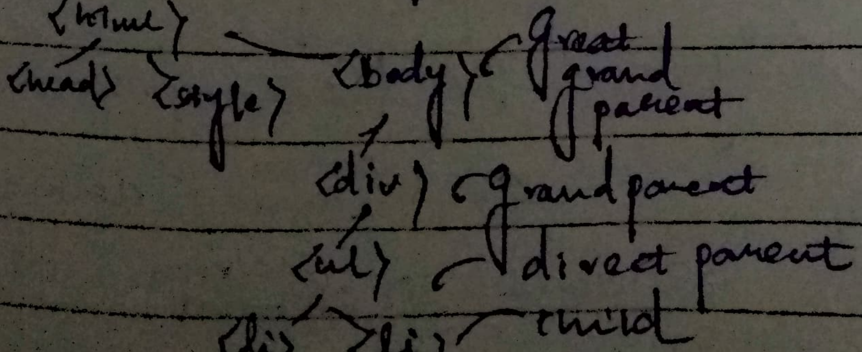
↳ Multiple levels ↳ use ↳ find()

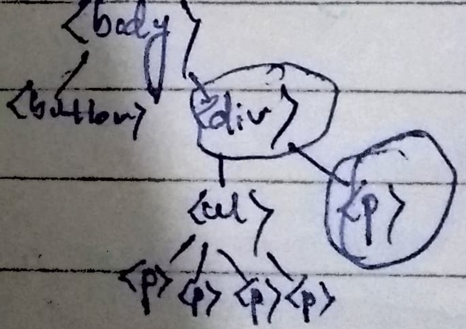
↳ Single level up DOM tree ↳ parent()/parents()

↳ Return all children including text nodes ↳ contents()

↳ Syntax: `$(selector).children(filter)`

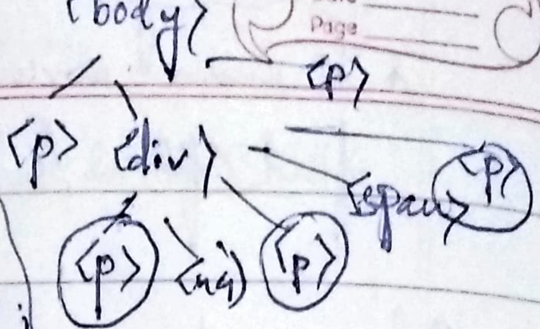
- filter ↳ optional/specifies selector expression to narrow down search for children.





eg :-

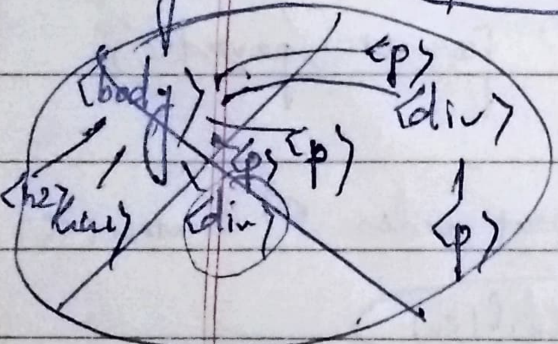
```
$(document).ready(function(){
        $("button").click(function(){
    $("div").children("p")
                    .css("bac—", "red");
    });
});
```



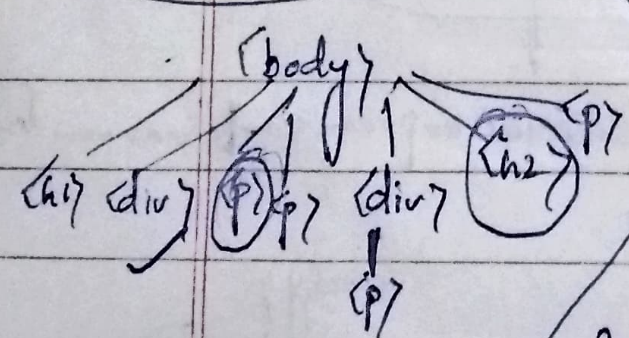
→ .next() :- It returns next sibling element of selected element.

→ Sibling element → element sharing same parent.

→ Syntax → $(selector).next(filter) → optional



eg :- $("div").next().css("—", "—");



→ .closest()

→ It returns the 1st ancestor of selected element.

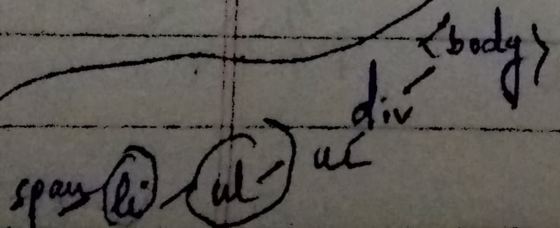
Parent, grand, great grand

→ Traverses upwards

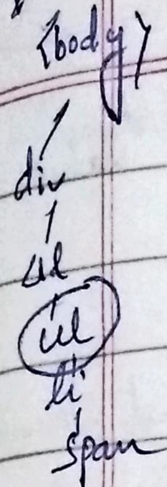
Syntax :- 1) $(selector).closest(filter)

2) $(selector).closest(filter, content)

→ returns 1st ~~element~~ ancestor of selected element in DOM tree



filter → required

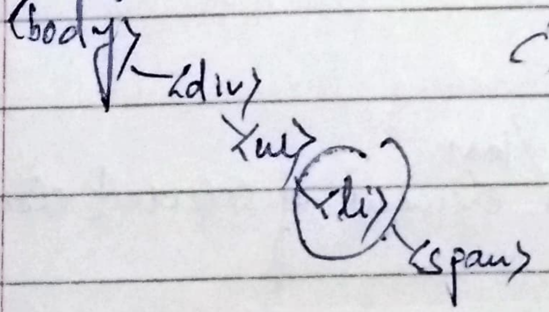


```
eg:- $(document).ready(function(){
        $("button").click(function(){
            $("span").closest("ul").css({"--});
        });
    });
```

→ .parent() → return direct parent element of selected ~~find parent~~ element.

→ Traverse single level up DOM Tree

→ Syntax → $(selector).parent(filter) → optional



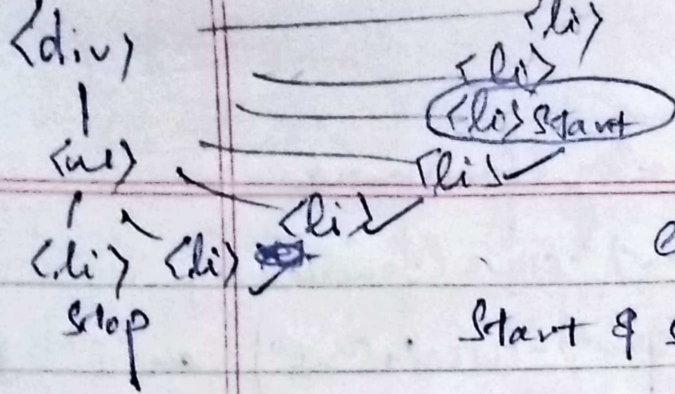
→ Eg: $("span").parent().css(——);

→ .prevUntil() → returns all previous sibling elements b/w selector & stop.

→ Traverses backward along siblings of DOM elements

→ prev() → returns all previous sibling element of selected element

→ prevAll() → return all previous sibling elements of selected element

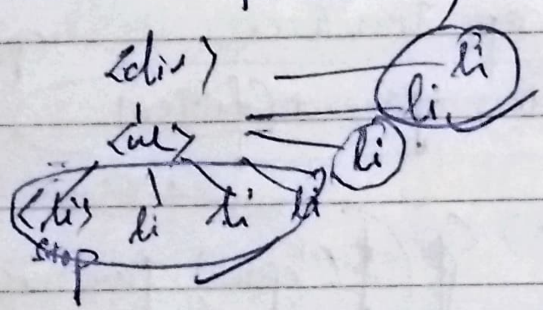


→ It returns all previous siblings element (b/w) li element with class name Start & stop.

→ • siblings → Returns all siblings element of selected element.

→ It traverse forward & backward along siblings of DOM elements.

→ Syntax → $(selector).siblings(filter) → optional



→ • first() → return the first /last element of selected element

& • last()

→ Syntax :

```
$(selector).first()

$(selector).last()
```

→ .is() → Checks if one of selected element matches the selector element.

Syntax :

```
$(selector).is(selectorElement/function(index, element))
```

→ Selector element → required
→ returns → T → If atleast one match
→ F → If no match

→ function(index, element)
→ Optional → Specify a function to run for group of selected element.
→ index → Index pos of element
→ element → Current element.

Eg: <div>
|
<p>

```
$(document).ready(function(){
    $("button").click(function(){
        if($("p").parent().is("div"))
        {
            let p = $(".result");
            p.text("Yes; " ");
        }
    });
});
```

→ map() :- Translates all items in an array or object to a new array.

Syntax → jQuery.map(array/object, callback);

→ array/object → It holds arrays/object to translate
→ callback → It holds function to process each item against.

Eg:-
```
$(document).ready(function(){
  $("button").click(function(){
     Var array = [55, 12, 49, 89, 34, 88];
    Var newArray = jQuery.map(array, function(val){
        return val+5;
    });

  document.getelementById("view").innerHTML
                = "<b>New :</b>" + Json stringfy(newArray);
    });
  });
```

→ .filter() → It is used to filter out all elements that do not match selected criteria & those matches will be returned.

Syntax: $(selector).filter(criteria, function(index))

eg
```
$("p").filter(".p").css("- - -");
```

→ .not() → Return all element which do not match with selected element.

Syntax → $(selector).not(criteria, function(index))

→ opposite of filters

→ • andSelf() → addBack()

→ Adds the previous set of elements to current set.

→ Syntax → $(selector).andSelf() → No arguments

→ Eg: The error because andSelf() → deprecated → jQuery 1.8 and removed in jQuery 3.0

→ Use .addBack()

```
$("div").find("p").addBack().addClass("border");
```

When we call → selects all p elements inside div>

addBack → include <div> → itself (allowing both <p> & <div> modified together.

→ .each() → Specify function to run for each matched element

→ Return false can be used to stop loop early.

→ Syntax

```
$(selector).each(function(index, element))
```

index → Index position

element → Current element

```
$("li").each(function(){
    alert($(this).text());
});
```

- find() → Finds all descendant elements of selected element.

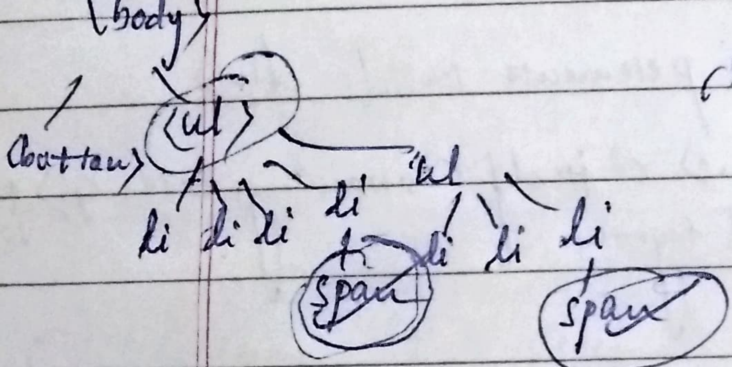
→ Descendant → child, grandchild, great-grandchild

→ Traverse downwards

→ Syntax → $(selector).find(filter) → required

→ To return all descendant element use → *

→ To return multiple descendants → seperate each expression with a comma.



→ eg: $("ul").find("span").css("-")