

## ax5. dom

Refer to this by [ax5](#).

### Methods

**static** `ax5.dom.after` (elements, val) `{Elements|Element}`

 [samples.js, line 2807](#)

엘리먼트에 다음에 노드를 추가합니다.

Name	Type	Description
elements	<code>Elements   Element</code>	
val	<code>String   Element</code>	

#### Example

```
1. var el = ax5.dom.get("[data-list-item='0']");
2. ax5.dom.after(el, "after");
```

**static** `ax5.dom.append` (elements, val) `{Elements|Element}`

 [samples.js, line 2755](#)

엘리먼트에 자식노드를 추가 합니다. (추가되는 위치는 맨 아래 입니다.)

Name	Type	Description
elements	<code>Elements   Element</code>	
val	<code>String   Element</code>	

#### Example

```
1. var el = ax5.dom.get("[data-list-item='0']");
2. ax5.dom.append(el, "자식노드");
3. ax5.dom.append(el, "<div>장기영<a href='#ABCDE'>이건 어렵다</a></div>");
4. ax5.dom.append(ax5.dom.get("[data-list-item='2']"), ax5.dom.get("#move-item"));
```

**static** `ax5.dom.attr` (elements, O) `{String|Elements}`

 [samples.js, line 2373](#)

elements에 attribute를 추가, 제거, 확인 합니다.

Name	Type	Description
elements	<code>Array</code>	대상의 엘리먼트 리스트 혹은 엘리먼트
O	<code>Object   String</code>	json타입또는 문자열

#### Returns:

- elements 또는 속성값

#### Example

```
1. ax5.dom.attr(ax5.dom.get("[data-ax-grid=A]"), {"data-ax-spt":"ABCD"}); // set attribute
2. ax5.dom.attr(ax5.dom.get("[data-ax-grid=A]"), {"data-ax-spt":"9999", "data-next":"next"}); // set attribute
3. ax5.dom.attr(ax5.dom.get("[data-ax-grid=A]"), "data-ax-spt"); // get attribute
4. ax5.dom.attr(ax5.dom.get("[data-ax-grid=A]"), {"data-next":null}); // remove attribute
```

**static** `ax5.dom.before` (elements, val) `{Elements|Element}`

 [samples.js, line 2791](#)

엘리먼트에 앞에 노드를 추가합니다.

Name	Type	Description
------	------	-------------

Name	Type	Description
elements val	Elements   Element String   Element	

#### Example

```
1. var el = ax5.dom.get("[data-list-item='0']");
2. ax5.dom.before(el, "before");
```

**static** `ax5.dom.box_model`(elements, cond) **{Object}**

[samples.js, line 2950](#)

엘리먼트의 box model 속성을 반환합니다.

Name	Type	Description
elements	Elements   Element	
cond	String	optional 원하는 박스 속성

#### Example

```
1. var axd = ax5.dom;
2. axd.box_model(el);
3. // {"offset": {"top": 101, "left": 110}, "position": {"top": 101, "left": 110}, "width": 181.71875, "height": 153, "padding": [5,4,3,2], "margin": [1,10,1,10], "border": ["2px double rgb(34, 34, 34)", "2px double rgb(34, 34, 34)", "2px double rgb(34, 34, 34)", "2px double rgb(34, 34, 34)"], "borderWidth": ["2", "2", "2", "2"], "boxSizing": "content-box"}
4.
5. axd.box_model(el, "offset");
6. axd.box_model(el, "position");
7. axd.box_model(el, "width");
8. axd.box_model(el, "height");
9. axd.box_model(el, "padding");
10. axd.box_model(el, "margin");
11. axd.box_model(el, "border");
12. axd.box_model(el, "borderWidth");
13. axd.box_model(el, "border-width");
14. axd.box_model(el, "boxSizing");
15. axd.box_model(el, "box-sizing");
16. // 각각의 박스모델 속성을 지정하여 호출 할 수 있습니다. borderWidth, border-width 중 하나의 방법으로 사용 가능합니다.
```

**static** `ax5.dom.child`(elements) **{Elements}**

[samples.js, line 2486](#)

타겟 엘리먼트의 자식을 반환합니다.

Name	Type	Description
elements	Element   Elements	

#### Example

```
1. <ul id="list-container">
2.   <li data-list-item="0">
3.     <div>child>child</div>
4.   </li>
5.   <li data-list-item="1"></li>
6.   <li data-list-item="2"></li>
7.   <li data-list-item="3"></li>
8.   <li data-list-item="4"></li>
9.   <li data-list-item="5"></li>
10. </ul>
11. <script>
12.   var el = ax5.dom.get("#list-container");
13.   console.log(
14.     ax5.dom.child(el)[1].tagName,
15.     ax5.dom.attr(ax5.dom.child(el)[1], "data-list-item")
16.   );
17. // LI 1
18. </script>
```

**static** `ax5.dom.class_name` (elements, command, O) `{String|Elements}`

 [samples.js, line 2317](#)

elements에 className 를 추가, 제거, 확인, 토글합니다.

Name	Type	Default	Description
elements	<code>Array</code>		대상의 엘리먼트 리스트 혹은 엘리먼트
command	<code>String</code>	has	<small>optional</small> add,remove,toggle,has
O	<code>String</code>		클래스명

Returns:

- elements 또는 클래스이름

Example

```
1. ax5.dom.class_name(ax5.dom.get("#abcd"), "class-text"); // has 와 동일
2. ax5.dom.class_name(ax5.dom.get("#abcd"), "add", "class-text");
3. ax5.dom.class_name(ax5.dom.get("#abcd"), "remove", "class-text");
4. ax5.dom.class_name(ax5.dom.get("#abcd"), "has", "class-text");
5. ax5.dom.class_name(ax5.dom.get("#abcd"), "toggle", "class-text");
```

**static** `ax5.dom.create` (node\_nm, attr, val) `{Element}`

 [samples.js, line 2257](#)

createElement 구문을 효과적으로 수행합니다.

Name	Type	Description
node_nm	<code>String</code>	엘리먼트 이름
attr	<code>Object</code>	엘리먼트 속성정보
val	<code>String</code>	innerHTML 값

Example

```
1. ax5.dom.create("script", {type:"text/javascript", src:"../ax5.js"});
2. ax5.dom.create("div", {id:"createEleId", "class":"createEleClass"}, "<a>내가만든</a>");
```

**static** `ax5.dom.css` (elements, CSS) `{String|Object|Elements}`

 [samples.js, line 2288](#)

Elements에 CSS속성을 읽고 씁니다.

Name	Type	Description
elements	<code>Array</code>	대상의 엘리먼트 리스트 혹은 엘리먼트
CSS	<code>Object   Array   String</code>	

Example

```
1. ax5.dom.css(ax5.dom.get("#abcd"), {"color":"#ff3300"});
2. ax5.dom.css(ax5.dom.get("#abcd"), {"width":100});
```

**static** `ax5.dom.data` (elements, command, O) `{Elements|Element|String}`

 [samples.js, line 3020](#)

엘리먼트에 data를 속성을 추가하거나 반환합니다.

Name	Type	Default	Description
elements	<code>Elements   Element</code>		
command	<code>String   Object</code>	get	<small>optional</small> 명령어 "get set remove"
O	<code>Object   String</code>		json타입 또는 문자열

Example

```

1. var axd = ax5.dom, axu = ax5.util, res = "";
2.
3. var el = axd.get("#list-container");
4. axd.data(el, {a:1}); // set data
5. axd.data(el, "set", {a:1100, b:[0,1,2,3]}); // set data
6.
7. res += axd.data(el, "a"); // get data
8. res += axd.data(el, "get", "b"); // get data
9. console.log(res);
10. // 1100,1,2,3
11.
12. console.log(axd.data(el, "b"));
13. // [0, 1, 2, 3]
14. axd.data(el, "remove", "b"); // remove data
15. console.log(axd.data(el, "b"));
16. // ""
17.
18. axd.data(el, "remove"); // remove all

```

**static** **ax5.dom.empty** (elements) **{Elements}**

 [samples.js, line 2690](#)

엘리먼트의 자식을 모두 지워줍니다. 내용을 깨끗히 비워 냅니다.

Name	Type	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>	

#### Example

```

1. var el = ax5.dom.get("#list-container");
2. ax5.dom.empty(el);

```

**static** **ax5.dom.get** (query, sub\_query) **{Array}**

 [samples.js, line 2205](#)

CSS Selector를 이용하여 HTML Elements를 찾습니다.

Name	Type	Description
query	<a href="#">String</a>   <a href="#">Element</a>   <a href="#">ax5.dom.0</a>	CSS Selector   Element
sub_query	<a href="#">String</a>	CSS Selector

#### Example

```

1. ax5.dom.get("#element01");
2. ax5.dom.get("input[type='text']");
3. ax5.dom.get( ax5.dom.get("input[type='text']") );

```

**static** **ax5.dom.get\_one** (query, sub\_query) **{Element}**

 [samples.js, line 2241](#)

CSS Selector를 이용하여 HTML Element를 찾습니다.

Name	Type	Description
query	<a href="#">String</a>   <a href="#">Element</a>	CSS Selector   Element
sub_query	<a href="#">String</a>	CSS Selector

#### Example

```

1. ax5.dom.get_one("#element01");
2. ax5.dom.get_one("input[type='text']");

```

**static** **ax5.dom.height** (elements) **{Number}**

 [samples.js, line 2675](#)

엘리먼트의 너비를 반환합니다.

Name	Type	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>	

Example

```
1. var el = ax5.dom.get("#list-container");
2. ax5.dom.height(el);
```

**static** `ax5.dom.html`(elements, htmlcode) [{Elements|String}](#)

 [samples.js, line 2717](#)

엘리먼트안에 HTML코드를 바꿔치기 합니다.

Name	Type	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>	
htmlcode	<a href="#">String</a>	<div>optional</div>

Example

```
1. var el = ax5.dom.get("#list-container");
2. console.log( ax5.dom.html(el) );
3. ax5.dom.html(el, "<a href='#1234'>링크");
```

**static** `ax5.dom.next`(elements, times) [{Element|null}](#)

 [samples.js, line 2624](#)

형제 엘리먼트중에 다음에 위치한 엘리먼트를 반환합니다.

Name	Type	Default	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>		
times	<a href="#">Number</a>	1	<div>optional</div> 횟수

Returns:

- 원하는 위치에 아이템이 없으면 `null` 을 반환합니다.

Example

```
1. <div>
2.   <ul id="list-container">
3.     <li data-list-item="0" id="li0">
4.       <div>child>child</div>
5.     </li>
6.     <li data-list-item="1" id="li1"></li>
7.     <li data-list-item="2" id="li2"></li>
8.     <li data-list-item="3" id="li3"></li>
9.     <li data-list-item="4" id="li4"></li>
10.    <li data-list-item="5" id="li5"></li>
11.  </ul>
12. </div>
13. <script>
14. var el = ax5.dom.get("#list-container");
15. var li = ax5.dom.child(el) [0];
16. var c_li;
17.
18. console.log(
19.   (c_li = ax5.dom.next(li, 2)).id,
20.   (c_li = ax5.dom.prev(c_li)).id
21. );
22. </script>
```

**static** `ax5.dom.off`(elements, type, \_fn)

 [samples.js, line 2453](#)

elements에 이벤트를 언바인드 합니다.

Name	Type	Description
elements	Array	
type	String	이벤트 타입
_fn	function	<small>optional</small> 이벤트 콜백함수

#### Example

```
1. var mydom = ax5.dom.get("[data-event-test=text-box]");
2. ax5.dom.off(mydom, "click");
3. ax5.dom.off(mydom, "click.fnb");
```

**static** `ax5.dom.offset` (elements) `{Object}`

 [samples.js, line 2879](#)

엘리먼트의 offset 값을 반환합니다.

Name	Type	Description
elements	Elements   Element	

#### Example

```
1. console.log(
2.   ax5.util.to_json(ax5.dom.offset(el))
3. );
4. // {"top": 8, "left": 8}
```

**static** `ax5.dom.on` (elements, type, \_fn)

 [samples.js, line 2407](#)

elements에 이벤트를 바인드 합니다.

Name	Type	Description
elements	Array	
type	String	이벤트 타입
_fn	function	이벤트 콜백함수

#### Example

```
1. var fna = function() {console.log("fna")};
2. var fnb = function() {console.log("fnb")};
3. var fnc = function() {console.log("fnc")};
4.
5. var mydom = ax5.dom.get("[data-event-test=text-box]"), remove_dom = ax5.dom.get("[data-event-test=remove]");
6.
7. ax5.dom.on(mydom, "click", window.fna);
8. ax5.dom.on(mydom, "click", window.fnb);
9. ax5.dom.on(mydom, "click", window.fnc);
10.
11. ax5.dom.on(remove_dom, "click", function(){
12.   ax5.dom.off(mydom, "click", window.fna);
13.   ax5.dom.off(remove_dom, "click");
14.   alert("이벤트 제거");
15. });
16.
17. // 핸들방식
18. ax5.dom.on(mydom, "click.fna", window.fna);
19. ax5.dom.on(mydom, "click.fnb", window.fnb);
20. ax5.dom.on(mydom, "click.fnc", window.fnc);
```

**static** `ax5.dom.parent` (elements, cond) `{Element}`

 [samples.js, line 2525](#)

타겟엘리먼트의 부모 엘리먼트 트리에서 원하는 조건의 엘리먼트를 얻습니다.

Name	Type	Description
elements	<a href="#">Element</a>	target element
cond	<a href="#">Object</a>	원하는 element를 찾을 조건

#### Example

```
// cond 속성정의
var cond = {
  tagname: {String} - 태그명 (ex. a, div, span..),
  clazz: {String} - 클래스명
  [, 그 외 찾고 싶은 attribute명들]
};
console.log(
  ax5.dom.parent(e.target, {tagname:"a", clazz:"ax-menu-handel", "data-custom-attr":"attr_value"})
);
```

**static** `ax5.dom.position` (elements) [{Object}](#)

[samples.js, line 2912](#)

엘리먼트의 상대 offset 값을 반환합니다.

Name	Type	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>	

#### Example

```
console.log(
  ax5.util.to_json(ax5.dom.position(el))
);
// {"top": 8, "left": 8}
```

**static** `ax5.dom.prepend` (elements, val) [{Elements|Element}](#)

[samples.js, line 2773](#)

엘리먼트에 자식노드를 추가 합니다. (추가되는 위치는 맨 처음 입니다.)

Name	Type	Description
elements	<a href="#">els</a>   <a href="#">Element</a>	
val	<a href="#">String</a>   <a href="#">Element</a>	

#### Example

```
var el = ax5.dom.get("[data-list-item='0']");
ax5.dom.prepend(el, "자식노드");
ax5.dom.prepend(el, "<div>장기영<a href='#ABCDE'>이건 어렵다</a></div>");
ax5.dom.prepend(ax5.dom.get("[data-list-item='2']"), ax5.dom.get("#move-item"));
```

**static** `ax5.dom.prev` (elements, times) [{Element|null}](#)

[samples.js, line 2588](#)

형제 엘리먼트중에 앞서 위치한 엘리먼트를 반환합니다.

Name	Type	Default	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>		
times	<a href="#">Number</a>	1	<small>optional</small> 횟수

#### Returns:

- 원하는 위치에 아이템이 없으면 null 을 반환합니다.

#### Example

```
<div>
  <ul id="list-container">
    <li data-list-item="0" id="li0">
      <div>child>child</div>
    </li>
```

```

        <li data-list-item="1" id="li1"></li>
        <li data-list-item="2" id="li2"></li>
        <li data-list-item="3" id="li3"></li>
        <li data-list-item="4" id="li4"></li>
        <li data-list-item="5" id="li5"></li>
    </ul>
</div>
<script>
var el = ax5.dom.get("#list-container");
var li = ax5.dom.child(el)[0];
var c_li;

console.log(
    (c_li = ax5.dom.next(li, 2)).id,
    (c_li = ax5.dom.prev(c_li)).id
);
</script>

```

**static** **ax5.dom.ready** (\_fn)

 [samples.js, line 2108](#)

document 로드 완료를 체크 합니다.

Name	Type	Description
_fn	<a href="#">function</a>	로드완료시 호출함수

#### Example

```

var a = 1;
setTimeout(function(){
    ax5.dom.ready(function(){
        console.log("test" + a);
        console.log(ax5.util.left("axisj-ax5", "-"));
    });
}, 1000);

```

**static** **ax5.dom.remove** (elements)

 [samples.js, line 2858](#)

엘리먼트를 제거 합니다.

Name	Type	Description
elements	<a href="#">Elements</a>   <a href="#">Element</a>	

#### Example

```

var el = ax5.dom.get("[data-list-item='0']");
ax5.dom.remove(el);

```

**static** **ax5.dom.resize** (\_fn)

 [samples.js, line 2168](#)

브라우저 resize 이벤트를 캐치합니다.

Name	Type	Description
_fn	<a href="#">function</a>	캐치후 호출될 함수

#### Example

```

ax5.dom.resize(function(){
    console.log( 1, document.body.clientWidth );
});

```

**static** **ax5.dom.scroll** (\_fn)

 [samples.js, line 2182](#)



브라우저 `scroll` 이벤트를 캐치하여 사용자 함수를 호출 하거나 스크롤 포지션을 리턴합니다.

Name	Type	Description
<code>_fn</code>	<code>function</code>	<small>optional</small> 캐치후 호출될 함수

#### Example

```
ax5.dom.scroll(function() {  
    console.log( 1, ax5.dom.scroll().top );  
});
```

---

**static** `ax5.dom.width` (elements) `{Number}`

 [samples.js, line 2660](#)

엘리먼트의 너비를 반환합니다.

Name	Type	Description
<code>elements</code>	<code>Elements   Element</code>	

#### Example

```
var el = ax5.dom.get("#list-container")  
ax5.dom.width(el);
```