

접근성을 확보하는 가장 좋은 방법은 시맨틱한 태그를 사용하여 마크업 하는 것이지만 시맨틱태그사용이 힘든경우나 커스텀 팝업 사용, AJAX통신을 하여 데이터를 불러오는 경우 처럼 접근성을 확보하기가 힘든경우가 있습니다.

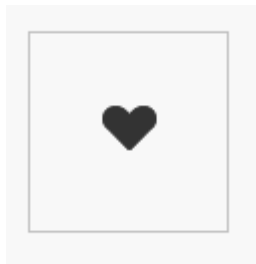
인터파크 UI개발을 하면서 많이 접할 수있는 UI와 간편한 속성으로도 접근성 확보 효과가 좋은 속성을 소개하려고 합니다.

테스트는 PC NVDA프로그램, Chrome 브라우저로 하였습니다.

(다른 스크린리더 프로그램, 설정, 브라우저에 따라 출력결과가 달라집니다)

Toggle 버튼

기존에는 좋아요버튼, 음소거버튼 등등 토글버튼의 상태를 설명하기 위해 DOM조작이 필요했습니다.



아래처럼 진행하는경우가 많았죠

[첫번째방법]

```
btnLike클래스 내 문구 변경해주기 좋아요 활성화, 좋아요 비활성화
<button class="btnLike"> 좋아요 활성화 </button>
```

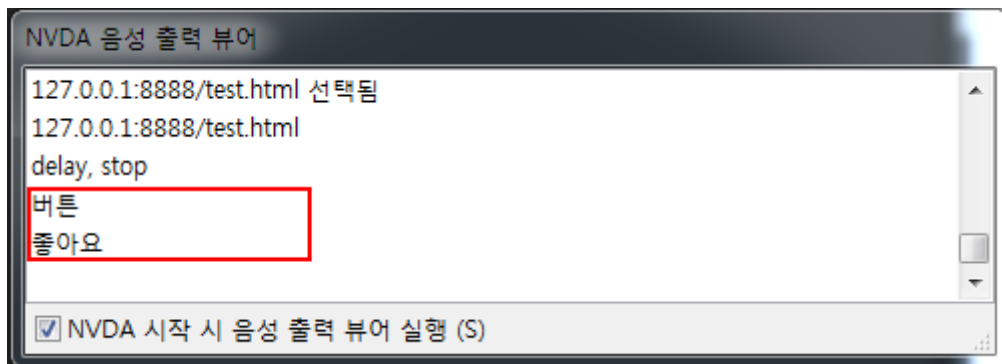
[두번째방법]

```
활성화되는 버튼 display로 조작하기
<button class="btnLike" style="display:none">좋아요 활성화</button>
<button class="btnLike" style="display:block">좋아요 비활성화</button>
```

첫번째방법처럼 DOM을 다시 그려주는 작업이 많으면 성능저하 될 수있고,

두번째방법은 안좋은방법은 아니지만 코드관리하기가 힘든 단점이 있습니다.

두방법 모두 상태가 변경한 후(활성화에서 비활성으로 변경) 다시 스크린리더에서 상태를 읽어주지 않습니다.



aria-pressed

aria-pressed속성을 사용하면 최소한의 DOM조작으로 접근성을 확보할 수 있습니다. 상태가 변경되면 스크린리더에서 상태를 읽어줍니다.

aria-pressed= **"true"** //토글버튼 활성화시
aria-pressed= **"false"** //토글버튼 비활성화시

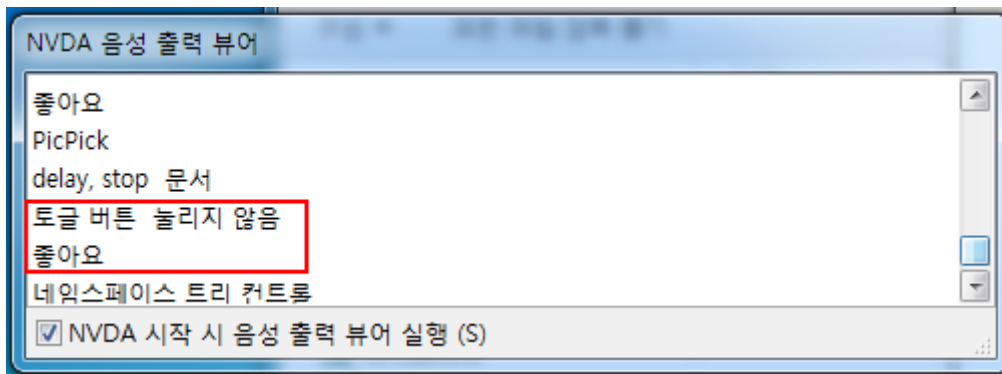
```
<button class="btnLike" aria-pressed="false">좋아요</button>
```

또한 클래스제어 없이도 속성선택자를 이용해 스타일을 제어할 수 있습니다.

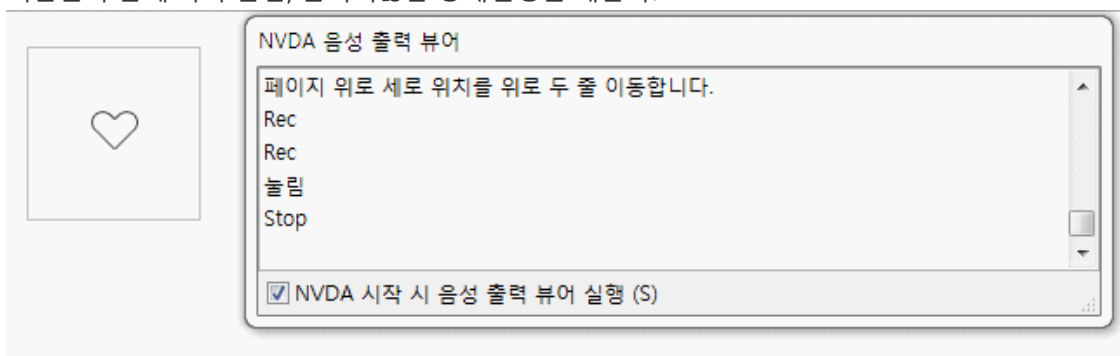
```
.btnLike[aria-pressed="true"]:after{background-position:100% 0}
```

테스트결과

1. 버튼을 토글버튼으로 읽어준다.



2. 버튼을 누를때 마다 눌림, 눌리지않음 상태설명을 해준다.



아코디언 UI

FAQ, QNA 등등 운영페이지, 이벤트 내 유의사항 영역에서 많이 사용하는 아코디언 UI
아코디언메뉴가 펼쳐지거나 닫혀있거나 상태를 설명해줘야 정확히 내용을 확인할 수 있습니다.
또한 스크린리더 사용자들은 해당 객체가 레이어를 열거나 하위 목록을 여는 것인지 알기 어렵다는 점
이 있습니다.

위에서 설명드렸듯 상태가 변할때 마다 버튼텍스트를 변경하는 DOM조작은 성능이슈, 상태변경시 설명 해주지 않는 단점이 있습니다.

꼭확인하세요

aria-expanded

aria-expanded 속성을 사용하면 단점을 보완하고 간편하게 사용할 수 있습니다.

aria-expanded = **"true"** //아코디언 메뉴, 팝업 활성화시

aria-expanded = **"false"** //아코디언 메뉴, 팝업 비활성화시

aria-expanded은 다른 UI를 제어하는 속성이기 때문에 **aria-controls** 속성과 함께 쓰입니다.
(사실 aria-controls속성이 없어도 스크린리더에서는 동일하게 읽어주지만,
aria-controls속성을 사용하면 직접적으로 접근할 수있는 기능이 있다고 합니다.)

aria-controls = **"제어하는 UI의 ID값"**

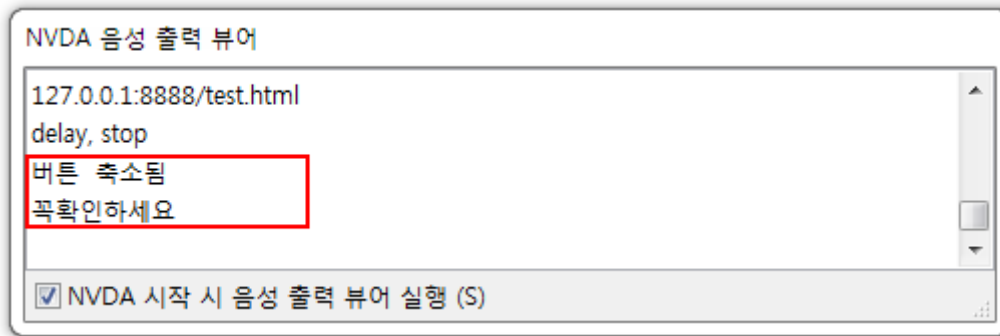
```
<div class="noticeArea">
  <button class="btnNotice" aria-expanded="false" aria-
controls="noticeContent">
    꼭확인하세요
  </button>
  <div id="noticeContent" class="listNoticeArea">
    <ul class="listNotice">
      <li>1. 유의사항</li>
      <li>2. 유의사항</li>
      <li>3. 유의사항</li>
      <li>4. 유의사항</li>
      <li>5. 유의사항</li>
    </ul>
  </div>
</div>
```

또한 클래스제어 없이도 형제선택자를 이용해 유의사항내용을 펼치고, 닫을 수 있습니다.

```
.btnNotice[aria-expanded="true"] + .listNoticeArea{display:block}
```

테스트결과

1. "축소됨" 상태를 읽어줌에 따라 해당버튼이 아코디언 버튼이라는 것을 알 수 있다.



2. 유의사항 버튼을 누를때마다 **확장됨**, **축소됨** 상태를 알려준다.



커스텀 팝업

커스텀 팝업은 접근성확보가 안되는 대표적인 UI 입니다.

팝업경우 활성화가 되면 아래컨텐츠에 포커스를 막고, 팝업안에서만 포커스가 이동 할 수있도록 제어해야 합니다.

팝업에 관련된 접근성을 지키려면 스크립트가 많이 적용됩니다.

(커스텀 팝업 관련 스크립트는 따로 정리하겠습니다~)

스크린리더 사용자들은 해당 객체가 레이어를 열거나 하위 목록을 여는 것인지 알기 어렵다는 점이 있습니다.
이번에 소개하는 속성은 클릭하는 버튼이 팝업을 여는 버튼이라는 것을 알려주는 역할을 합니다.

aria-haspopup

aria-haspopup 속성은 WAI-ARIA 1 버전에서는 true, false 값만 지원이 되었지만 1.1 버전에서는 값이 세부화 되었습니다.

value	
true	<code>menu</code> 와 동일한 의미.
menu	<code>menu(role)</code> 팝업이 연결됨. <code>menu(role)</code> 는 링크 목록.
dialog	<code>dialog(role)</code> 팝업이 연결됨. <code>dialog(role)</code> 는 상호작용 요소(버튼 또는 컨트롤)가 포함된 현재 문서의 하위창.
listbox	<code>listbox(role)</code> 팝업이 연결됨. <code>listbox(role)</code> 는 선택 가능한 <code>option(role)</code> 을 포함한 콤보박스.
tree	<code>tree(role)</code> 팝업이 연결됨. <code>tree(role)</code> 는 하위 <code>list(role)</code> 을 포함하며 접고 펼칠 수 있음.
grid	<code>grid(role)</code> 팝업이 연결됨. <code>grid(role)</code> 는 행과 열로 구성된 선택 가능한 위젯. 상호작용 가능한 셀이기 때문에 <code>table(role)</code> 과는 역할이 다름에 유의.
false(default)	연결된 팝업이 없음을 의미.

value값은 활성화하는 팝업의 role에 맞춰서 사용하면 됩니다.

아래 예제에서는 dialog팝업을 사용하였기 때문에 버튼태그에 **aria-haspopup="dialog"**를 사용하였습니다.

```
<button class="btnPopup" aria-haspopup="dialog" aria-controls="login-dialog">
  이벤트참여하기
</button>
<div class="popupwrap" role="dialog" id="login-dialog" aria-modal="true">
  <div class="popupContent">
    <h3 class="tit">이벤트 참여하시겠습니까?</h3>
    <div class="btnArea">
      <button id="btnOk">예</button>
      <button id="btnNo">아니요</button>
    </div>
  </div>
</div>
```

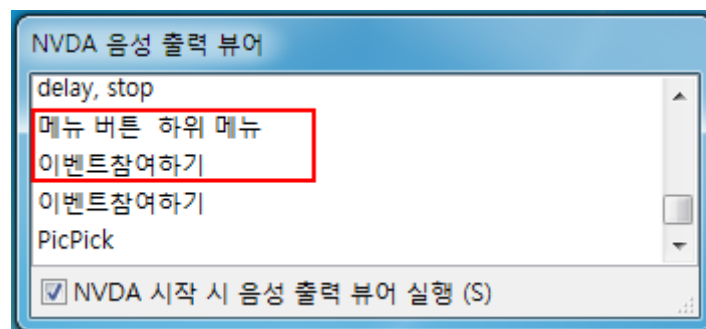
aria-haspopup은 aria-expanded 속성과 같이 다른 UI를 제어하는 속성이기 때문에 **aria-controls** 속성과 함께 쓰입니다.

aria-controls = "제어하는 UI의 ID값"

테스트결과

버튼을 누르면 팝업이 뜨는 것을 알 수 있다.

(value값으로 itemlist, tree, dialog 다양한 속성을 사용해도 동일하게 출력됩니다.)



현재상태(현재페이지, 현재단계, 현재시간, 현재날짜) 표시

GNB, 페이지네이션에서 **현재위치**표시나 주문서, 회원가입에서 **현재단계**표시를 해야합니다.

현재페이지 텍스트를 계속 메뉴에 넣어주는 방법도 있지만 DOM 조작은 성능저하로 추천하는 방법이 아닙니다.

두번째 방법으로 현재페이지는 강조태그를 써서 넣어주는 방법도 있지만 직관적인 방법도 아니지만, 스크린리더에서 포커스가 가지 않으면 읽어주지 않는 이슈도 생길 수 있습니다.

aria-current

aria-current속성으로 간단하게 해결할 수 있습니다.

aria-current 속성은 value값이 다양합니다.

사용하는 상황에 맞게 사용하면 됩니다.

value	
page	현재 '페이지'와 일치하는 시각적으로 강조한 링크.
step	현재 '단계'와 일치하는 시각적으로 강조한 링크.
location	플로우 차트에서 현재 '위치'와 일치하는 시각적으로 강조한 이미지.
date	달력에서 현재 '날짜'와 일치하는 날짜.
time	시간표에서 현재 '시간'과 일치하는 시간.

아래예제에서는 aria-current="page" 를 사용했습니다.

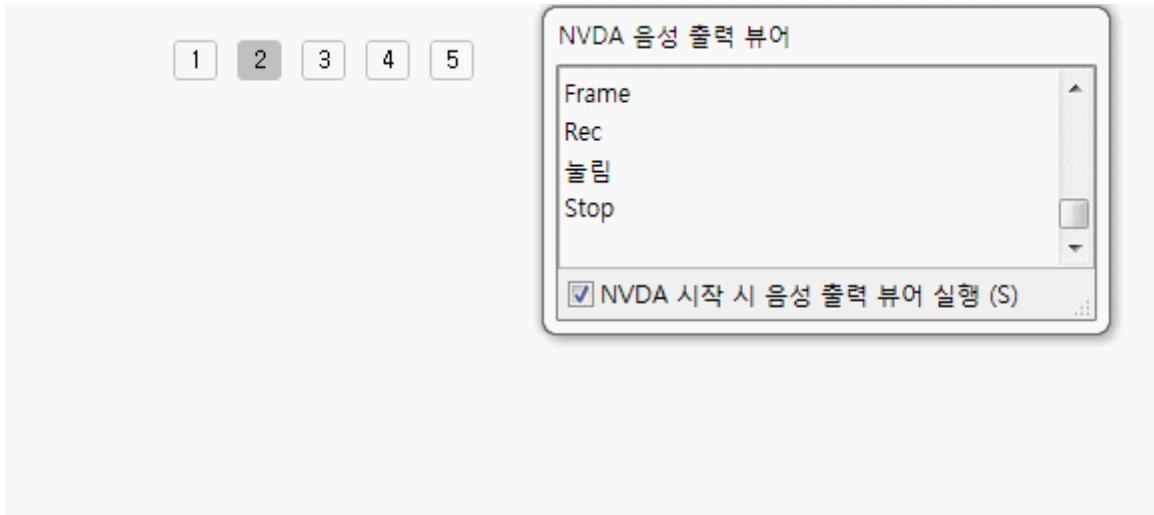
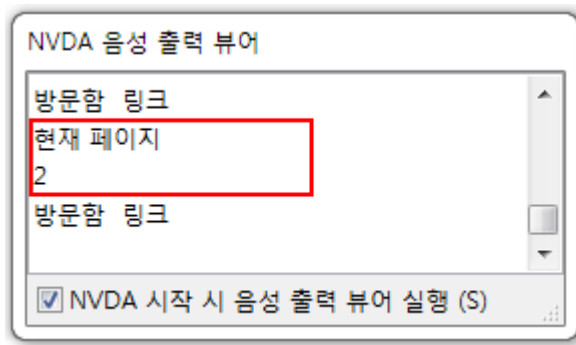
```
<div class="pagination">
  <h3>페이지네이션</h3>
  <ul class="pages">
    <li><a href="#">1</a></li>
    <li aria-current="page"><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
  </ul>
</div>
```

또한 속성선택자로 스타일을 적용할 수 있습니다.

```
.pages li[aria-current="page"]{background:silver;}
```

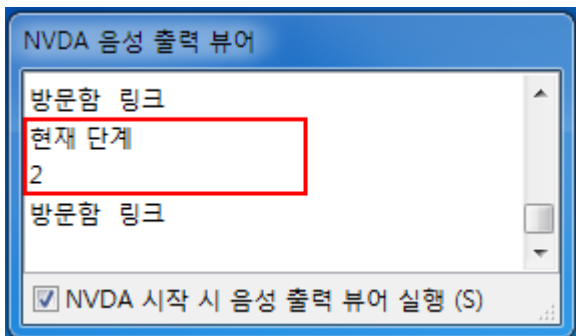
테스트결과

속성이 적용된 페이지네이션에 현재페이지라고 읽어줍니다.

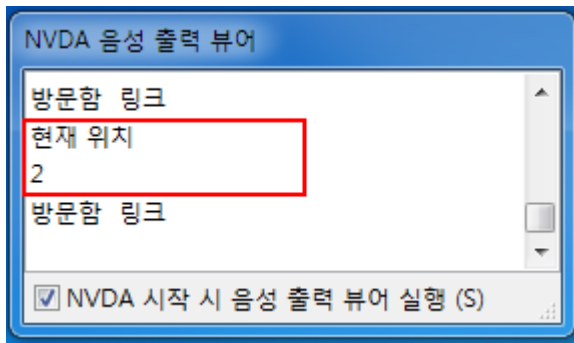


다른속성들 적용결과

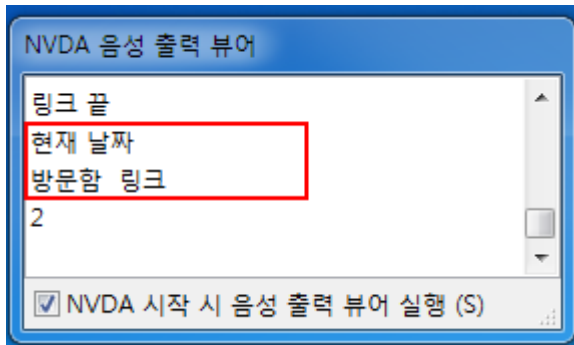
```
<div aria-current="step">2</div>
```



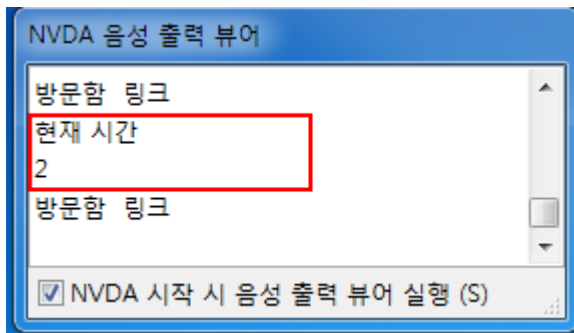
```
<div aria-current="location">2</div>
```

```
<div aria-current="date">2</div>
```



```
<div aria-current="time">2</div>
```



UI에 설명추가하기

UI에 설명을 추가해줘야하는 경우 마크업으로 추가할 수 있지만
aria 속성을 사용하면 UI에 포커스가 가면 설명을 같이 읽어주는 장점이 있습니다.

게시물 삭제

게시물 영구삭제

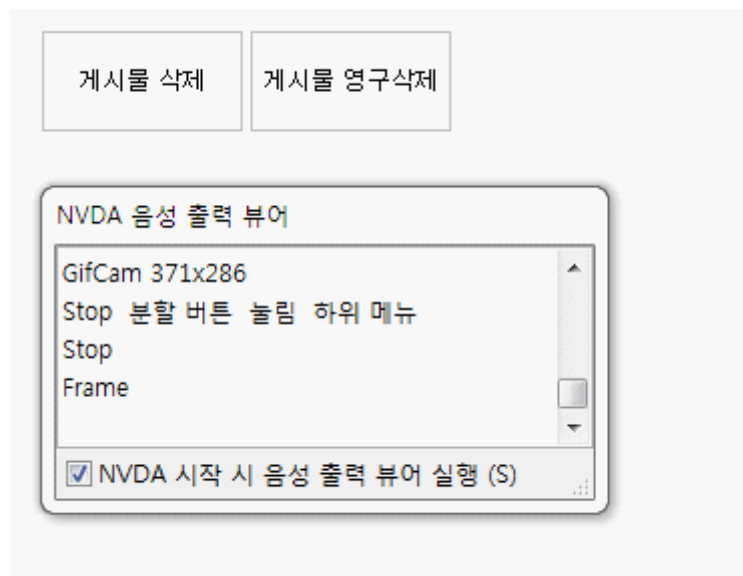
aria-labelledby, aria-label, aria-describedby

aria-labelledby="간결한 설명 참조하는대상 ID" ,
aria-label="간결한설명" ,
aria-describedby="자세한설명 참조하는대상 ID"

```
<button class="btnDelete" aria-describedby="TIPDEL">게시물 삭제</button>  
<button class="btnDelete" aria-describedby="TIPDEL">게시물 영구삭제</button>  
<p id="TIPDEL" role="tooltip" hidden>게시물 삭제 후 복원할 수 없음.</p>
```

테스트결과

버튼 누를때 마다 설명이 출력됩니다.



WAI-ARIA TIP!

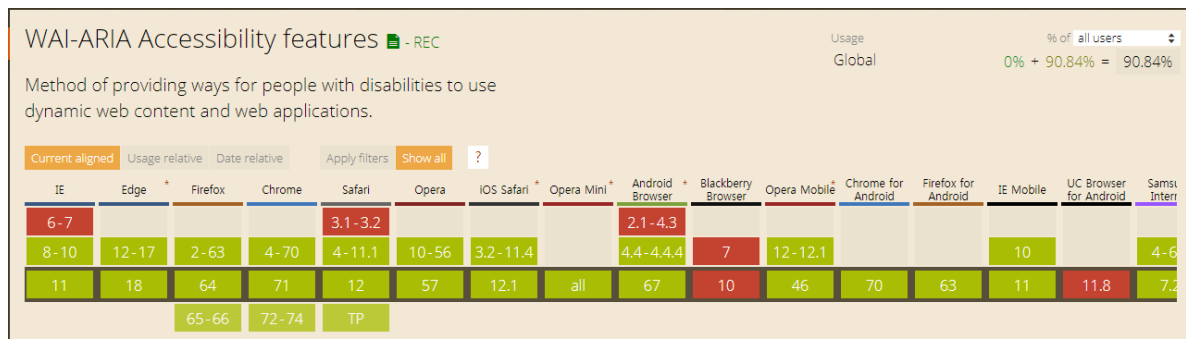
aria-pressed, aria-checked, aria-selected 속성이 헷갈리는 경우

aria-pressed : 토글버튼에 사용합니다. (음소거버튼, 앱알림버튼, 좋아요버튼)

aria-checked : 네이티브checkbox, radio태그가 아닌 커스텀 한 체크박스과 라디오 버튼에 사용합니다.

aria-selected : 탭UI와 같이 선택할 수 있는 요소에 사용됩니다.

이제는 최신브라우저에서 많이 지원을 하고 있는 WAI-ARIA 속성
요즘은 많이들 사용하려고 하는 추세인 것 같습니다.
간단한 속성부터 적용해 보면 좋을 것 같습니다.



참고한사이트 목록

- [널리](#)
- [레진개발가이드](#)
- [WAI-ARIA 1.1](#)
- [HTML5](#)