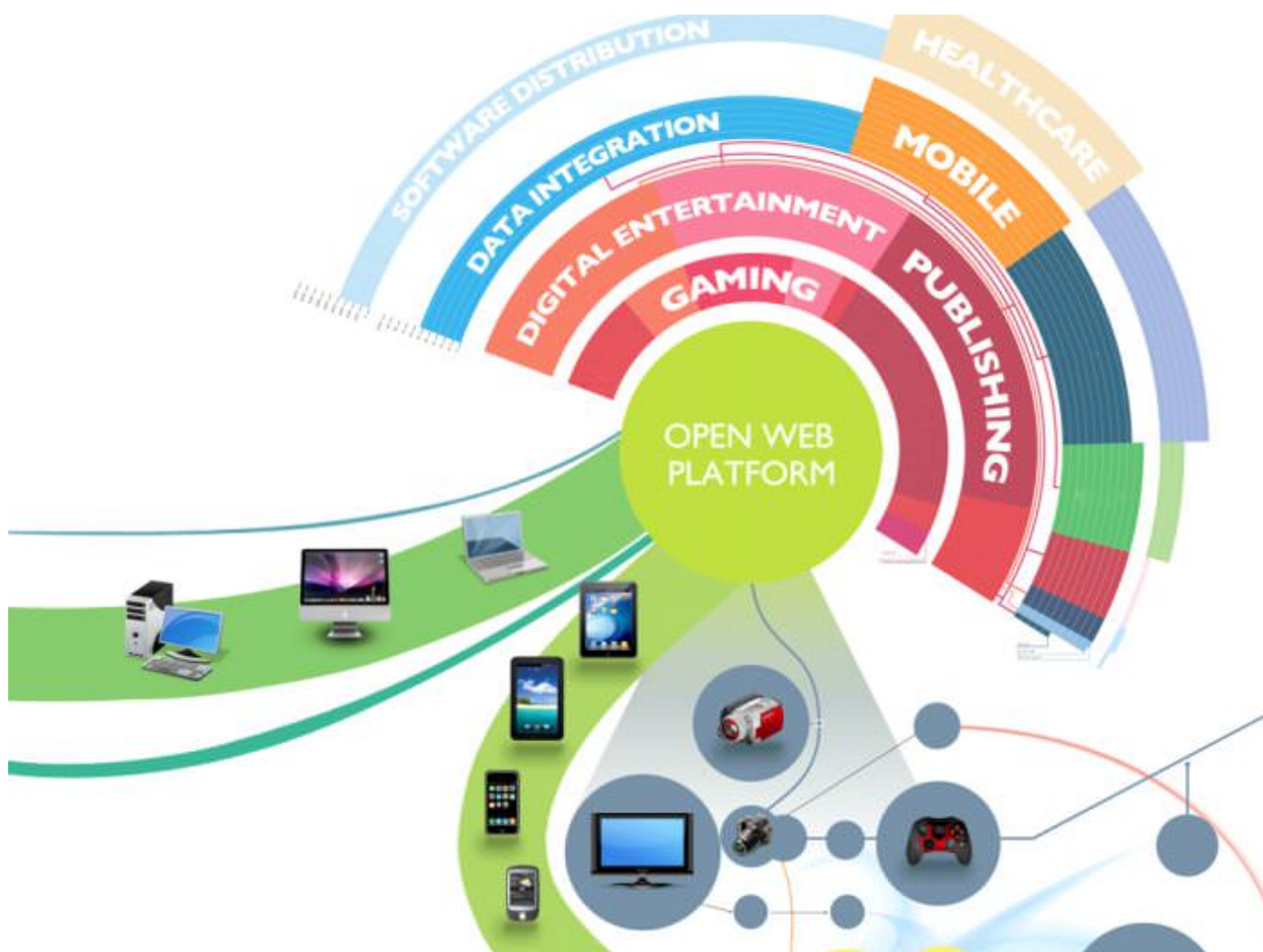


چند روز پیش [مطلبی](#) به عنوان اشتراک در سایت جاری معرفی شده که به ما یادآوری می‌کرد، ما تنها استفاده کنندگان سیستم‌های کامپیوتری، به خصوص اینترنت نیستیم و معلولین هم نیازمند استفاده از این فناوری‌ها هستند.

[WAI-ARIA](#) که برگرفته از *Web Accessibility Initiative - Accessible Rich internet Application* است به معنی برنامه‌ی اینترنتی تعامل گرا با خاصیت دسترسی پذیری بالا می‌باشد و یک راهنماست که توسط [کنسرسیوم وب \(+\)](#) معرفی گشته است تا وب سایت‌ها با رعایت این قوانین، دسترسی سایت خود را بالاتر ببرند. این قوانین به خصوص برای سایت‌هایی با محتوای پویا هستند که از فناوری‌هایی چون Ajax, Javascript, HTML و دیگر فناوری‌های مرتبط استفاده می‌کنند.

امروزه طراحان وب بیش از هر وقتی از فناوری‌های سمت کلاینت چون جاوااسکریپت برای ساخت رابط‌های کاربری استفاده می‌کنند که html به تنها قادر به ایجاد آن‌ها نیست. یکی از تکنیک‌های جاوااسکریپت، دریافت محتوای جدید و به روزآوری قسمتی از صفحات وب است بدون اینکه مجدداً کل صفحه از وب سرور درخواست گردد که به این تکنیک **Rich Internet Application** هم می‌گویند. تا به اینجای کار هیچ مشکلی نیست و خوب هم هست؛ ولی مشکلی که در این بین وجود دارد این است که این نوع تکنیک‌ها باعث از بین رفتن خاصیت دسترسی پذیری معلولین می‌گردند. معلولینی که از [صفحه خوان‌ها](#) استفاده می‌کنند یا به دلیل معلولیت‌های خود قادر به حرکت دادن ماوس نیستند.

ARIA با استفاده از خصوصیت‌ها Properties، نقش‌ها Roles و وضعیت‌ها States به طراحان برنامه‌های وب و سازندگان فناوری‌های یاری رسان، اجازه می‌دهد که با ابزارهای کمکی معلولان ارتباط برقرار کنیم و یک صفحه‌ی وب ساده را به یک صفحه‌ی پویا تبدیل کنیم. ARIA تنها یک استاندارد برای وب نیست، بلکه یک فناوری چند پلتفرم است که برای بازی‌های رایانه‌ای، موبایل‌ها، دستگاه‌های سرگرمی و سلامتی و دیگر انواع برنامه‌ها نیز تعریف شده است.



### فریمورک ARIA

خود HTML به تنهایی نمی‌تواند نقش‌های هر المان و ارتباط بین آن‌ها را به درستی بیان کند و به این منظور ARIA به کمک می‌آید. با استفاده از نقش‌ها میتوان هدف هر المان را مشخص کرد و با استفاده از خصوصیات ARIA نحوه‌ی عملکرد آن‌ها را تعریف کرد.

### نقش‌ها

نقش‌ها طبق مستندات کنسرسیوم وب بر 4 نوع هستند:

*Abstract Roles*

*Widget Roles*

*Document Roles*

*Landmark Roles*

فریمورک ARIA با استفاده از Landmark Roles یا نقش‌های راهنما، به معرفی بخش‌های ویژوالی می‌پردازد تا فناوری‌های کمکی به آن دسترسی سریعی داشته باشند. هشت نقش راهنما وجود دارد که در زیر آن‌ها را بررسی می‌کنیم.

این قسمت که عموماً برای اجزای مهمی مثل هدر سایت قرار می‌گیرد و شامل معرفی وب سایت هست و در تمامی صفحات وجود دارد که شامل لوگو، اطلاعات عمومی سایت و اسپانسرها و ... می‌گردد و بسیار مهم است که تنها یکبار در صفحه‌ی وب به کار برود و تکرار آن پرهیز شود.	<b>Banner</b>
این نقش به محتوای اصلی وب سایت اشاره می‌کند و نباید بیشتر از یکبار در هر صفحه‌ی وب به کار برود و عموماً بهتر است این خصوصیت در تگ div قرار گیرد:  <div Role="main"></div>  یا در HTML5 به طور مفهومی ساخته می‌شود:  <main role="main">.....	<b>Main</b>
اشاره به یک ناحیه پر از المان‌های لینک برای ارتباط با صفحات دیگر	<b>Navigation</b>
مشخص سازی ناحیه‌ای که اطلاعات اضافی درباره‌ی محتوای اصلی سایت دارد؛ مانند بخش مقالات مرتبط، آخرین کامنت‌ها و ...	<b>Complementary</b>
این نقش که بیشتر برای فوتر مناسب است برای محتوایی به کار می‌رود که در آن به قوانین کپی رایت و ... اشاره می‌شود.	<b>ContentInfo</b>
برای اشاره به فرم‌ها که دارای قسمت‌های ورودی کاربر هستند.	<b>form</b>
در صورتیکه فرمی دارید و از آن برای گزینه‌ی جست و جو استفاده می‌کنید، از این نقش استفاده کنید.	<b>search</b>
برای اینکه وب سایت خود را به صورت یک وب اپلیکیشن معرفی کنید؛ تا یک صفحه وب معمولی، استفاده می‌شود و برای وب سایت‌های قدیمی یا با حالت سنتی توصیه نمی‌شود و به برنامه‌های کمکیار معلولین می‌گویند که از حالت عادی به حالت application سوئیچ کنند؛ پس با <a href="#">دقت بیشتری</a> باید از این گزینه استفاده کرد.	<b>application</b>



### خصوصیت‌ها و وضعیت‌ها

در حالیکه از نقش‌ها برای معرفی هر المان یا تگ استفاده می‌کنید؛ خصوصیت‌ها و وضعیت‌ها به کاربر اطلاعات اضافی می‌دهند که چگونه از آن استفاده کنند. برای معرفی خصوصیت‌ها و وضعیت‌ها از یک خصوصیت که با `aria-` شروع می‌شود استفاده کنید. از معروفترین آن‌ها خصوصیت `aria-required` و وضعیت `aria-checked` می‌باشند، تا به ترتیب به کاربر اعلام کنید این تگ نیاز به پر شدن دارد، یا المانی نیاز به تغییر وضعیت انتخابی دارد. نحوه‌ی استفاده از آن‌ها به شکل زیر است:

```
<div id="some-id" class="some-class" aria-live="assertive"><div>
```

`aria-live` سه مقدار می‌پذیرد که عبارتند از `Off`, `Polite`, `Assertive` و مشخص می‌کنند که المان مورد نظر آپدیت پذیر هست یا خیر و اگر آری، نحوه‌ی به روز شوندگی آن به چه نحوی است.

### ساخت ارتباطات میان المان‌ها با خصوصیت‌های ارتباطی

مثال شماره یک

```
<div role="main" aria-labelledby="some-id">
  <h1 id="some-id">This Is A Heading</h1>
  Main content...
</div>
```

خصوصیت `aria-labelledby` به تعریف المان‌هایی با نام جاری می‌پردازد. این خصوصیت معرفی کننده‌ی برچسب هاست. به

عنوان مثال بین المان‌ها ورودی و برچسب آنان ارتباط ایجاد می‌کند تا ابزارهای معلولین، مانند صفحه خوان‌ها، در خواندن به مشکل برنخورند.

مثال شماره دو

در این مثال هر گروهی از المان‌ها یک برچسب و بعضی المان‌ها یک برچسب اختصاصی دارند که توسط خصوصیت معرفی شده aria-labelledby کامل شده‌اند:

```
<div id="billing">Billing Address</div>

<div>
  <div id="name">Name</div>
  <input type="text" aria-labelledby="name billing"/>
</div>
<div>
  <div id="address">Address</div>
  <input type="text" aria-labelledby="address billing"/>
</div>
```

مثال شماره سه

در این مثال گروهی از radio button با برچسبشان ارتباط برقرار می‌کنند.

```
<div id="radio_label">My radio label</div>
<ul role="radiogroup" aria-labelledby="radio_label">
  <li role="radio">Item #1</li>
  <li role="radio">Item #2</li>
  <li role="radio">Item #3</li>
</ul>
```

### خصوصیت‌ها و وضعیت‌های aria را با چرخه‌ی فعالیت‌های صفحه به روز کنید

در صورتیکه چرخه‌ی فعالیت صفحه‌ی شما تغییر می‌کند و تگ‌ها نیاز به مقادیر جدیدی از aria دارند، حتما این مقادیر را هم به نسبت تغییراتی که در صفحه رخ می‌دهد، تغییر دهید تا وضعیت بحرانی برای کاربر به خصوص در حین کار با فرم‌ها و ... پیش نیاید.

### هر aria را دوباره استفاده نکنید

امروزه به خصوص با آمدن html5 و ویژگی‌هایی چون تگ‌های مفهومی، کار بسیار راحت‌تر شده‌است و مرورگر به طور خودکار می‌تواند aria را بر روی بعضی از المان‌ها پیاده کند. به عنوان نمونه:

```
<form></form>
<form role="form"></form>
```

همچنین به عنوان مثال با استفاده از خصوصیات HTML مثل hidden کردن یک شیء نیازی به استفاده از وضعیت aria-hidden نمی‌باشد. مرورگر به طور پیش فرض آن را لحاظ می‌کند.

امروزه شاهد پیشرفت فناوری در همه‌ی عرصه‌ها هستیم و همیشه این پیشرفت‌ها ما را ذوق زده کرده‌اند، ولی یکی از بی نظیرترین استفاده‌های فناوری روز، استفاده در صنایع سلامتی است که نه تنها ما را ذوق زده می‌کند، بلکه از لحاظ احساسی هم ما را به وجد می‌آورد و جزء زیباترین نتایج فناوری می‌باشند. بسیاری از شرکت‌ها چون گوگل در این راستا فعالیت‌های زیادی کرده‌اند تا بتوانند سلامت جامعه را کنترل کنند، از ساخت [لنز چشمی برای کنترل دیابت](#) گرفته تا [ساخت قاشق خوری برای بیماران پارکینسون](#)، ولی استفاده‌های ساده از مسائلی مانند بالا به افراد معلول این مژده را می‌دهد که ما آن‌ها را فراموش نکرده‌ایم.