برنامهسازى پيشرفته

Hill

نویسنده: محمدمهدی مهدیزاده

مدرسان: رامتین خسروی، محمدامین صادقی

آشنایی با فرمتهای تصویر برداری

این سند بهمنظور آشنایی مقدماتی با فرمتهای ذخیرهی تصاویر برداری SVG و Canvas تهیه شده است. شما می توانید با استفاده از انواع منابع برخط آموختههای خود را از نظر کمی و کیفی بهبود ببخشید.

آشنایی با SVG

svg یا Scalable Vector Graphics یک زبان نشانه گذاری مبتنی بر XML برای توصیف تصاویر دوبعدی است. در سالهای اخیر استفاده از این زبان در صفحات وب برای نمایش فایلهای تصویر بسیار گسترش یافته است. در این قسمت ابتدا کمی با ساختار زبانهای مبتنی بر XML آشنا می شویم، سپس ساختار یک فایل svg را بیان می کنیم و در نهایت قسمت هایی از این زبان را که ممکن است شما در این تمرین نیاز به استفاده از آنها پیدا کنید به صورت اجمالی بررسی می کنیم.

زبانهای XML، برای ذخیره اطلاعات طراحی شدهاند. در عمل آنها کاری را انجام نمی دهند و فقط اطلاعاتی را به صورت ساختار یافته در اختیار نرمافزارهای دیگر قرار می دهند. در زبان XML، هر داده داخل یک تگ با نامی مشخص قرار می گیرد. تگها در این زبان معمولا به صورت جفت هستند و داده در میان آنها قرار دارد. شروع تگهای جفت با <> و پایان آنها با </> مشخص می شود. داده ی داخل تگها خود می تواند شامل چند تگ دیگر نیز باشد. در این زبان می توان از تگهایی به صورت تکی نیز استفاده کرد. این تگها در یک </> قرار می گیرند و فاقد داده ی درونی هستند. همچنین به هر تگ می توان چند خصیصه اختصاص داد. علاوه بر این برای هر خصیصه می توان یک مقدار نیز مشخص کرد. به مثال زیر توجه کنید:

<tagname atrribute-name1 attribute-name2="value">Content goes here ...</tagname>
<singletag attributeA attributeB="valueB"/>

در این مثال singletag و singletag نام تگها است. دادهای نیز در بین تگ tagname قرار گرفته است. در این تگ از دو خصیصهی atrribute-name1 و atrribute-name2 استفاده شده است و به atrribute-name2 مقدار value اختصاص داده شده است.

همانطور که گفته شد از زبان XML صرفا برای ذخیره دادهها به صورت ساختار یافته استفاده می شود و این زبان کاری انجام نمی دهد. گونه هایی از این زبان برای ذخیره دادههای خاص طراحی شده اند. به عنوان مثال همانطور که احتمالا قبلا شنیده اید، زبان HTML، برای توصیف محتوای یک صفحه ی وب استفاده می شود. در این زبان تعدادی تگ به صورت قراردادی تعریف شده اند که استفاده از آنها باعث ایجاد خروجی یکسانی در مرورگرهای وب می شود. به عنوان مثال استفاده از تگ </mm در HTML باعث می شود. که یک عکس در مرورگر نمایش داده شود. همچنین داده های تگ </m> به عنوان یک پاراگراف نمایش داده می شود.

SVG نیز یک زبان مبتنی بر XML است که برای توصیف تصاویر برداری استفاده می شود. در تصاویر برداری، بر خلاف تصاویر شطرنجی، تصویر به جای اینکه به صورت مجموعه ای از پیسکلها ذخیره شود، به صورت مجموعه ای از مشخصات هندسی نقاط، خطها، منحنی ها و چند ضلعی ها ذخیره می شوند. با آشنایی بیشتر با svg، نحوه توصیف تصاویر برداری در کامپیوتر را بهتر درک خواهید کرد.

برای نمایش فایلهای svg میتوانید این فایلها را در مرورگرهای وب مانند Chrome و Firefox بازکنید. توصیه میشود قبل از خواندن ادامه متن، برای آشنایی بیشتر با ساختار و فایلهای زبانهای مبتنی بر XML، کمی راجع به HTML گوگل کنید.

ساختار کلی یک فایل SVG

ساختار یک فایل svg به صورت زیر است:

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"

"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg version="1.1"

xmlns="http://www.w3.org/2000/svg">

Content Here ...
```

شما نیازی به فهمیدن و دانستن معنای تگهای ابتدایی این فایل نیستید و تنها لازم است که محتوایی که تولید میکنید را به جای ... Content Here قرار دهید.

در فایل های SVG بالا گوشه سمت چپ به صورت پیش فرض مبدا مختصات در نظر گرفته می شود. در یک فایل svg شما می توانید از واحدهای گوناگونی همانند سانتی متر، میلی متر، پیکسل و ... استفاده کنید؛ اما در این تمرین نیازی به استفاده از واحدها ندارید. بنابراین در مشخص کردن یک شکل کافیست اعداد مربوط به آنرا بدون واحد بنویسید. به مثال زیر توجه کنید:

<rect x="2" y="2" width="5" height="3"/>

</svg>

تگهایی که شما برای تولید محتوای svg نیز دارید احتمالا شامل SVG نیز نیاز است. در خواهد بود. همچنین برای اعمال مشخصههایی مانند رنگ، دورخط و غیره، به تعدادی از attributeهای SVG نیز نیاز است. در جدول ۱ هر کدام از این تگها و attributeهای مربوط به آنها به صورت اجمالی توضیح داده شدهاند. در این جدول به تکی یا جفتی بودن تگها دقت داشته باشید. همچنین دقت داشته باشید که نیازی به استفاده از تمام attributeهای معرفی شده برای یک تگ نیست. در صورتی که از یک attribute در یک تگ استفاده نشود، مقداری پیش فرض به آن attribute نسبت داده می شود. به علاوه، در ادامه جدولی نیز برای توضیح attributeهایی که تقریبا روی تمام تگهای معرفی شده قابل استفاده هستند، آمده است. دقت داشته باشید که برای انجام این تمرین شما از آزادی خوبی در استفاده از امکانات SVG برخوردارید، پس بهتر است برای حل مشکلات خود و همچنین آشنایی دقیق تر با تگهای معرفی شده و معرفی نشده در syg از گوگل استفاده کنید. می تواند از سایتهای زیر نیز به عنوان منبع جستجوی خود استفاده کنید. همچنین برای کشف نحوه عملکرد امکانات مختلف در شرایط مختلف می توانید خودتان با ساخت یک فایل syg، به صورت دستی آزمایش کنید.

- https://www.w3schools.com/graphics/svg_intro.asp
- https://www.w3schools.com/graphics/svg_reference.asp
- https://developer.mozilla.org/en-US/docs/Web/SVG

جدول ۱ _ برخی از تگهای SVG و مشخصات آنها

ر تحقای ۲۷ و مسحصات انها	· · · · · · · · · · · · · · · · · · ·	
مشخصهها	عملكرد	نگ
x -> مختصات افقی د - x مختصات عمودی - y اندازه افقی - width - اندازه افقی اندازه عمودی - height اندازه افقی - width گرد کردن افقی گوشهها - rx گرد کردن عمودی گوشهها - ry	یک مستطیل را مشخص میکند	<rect></rect>
cx -> مختصات افقی مرکز دایره -> مختصات عمودی مرکز دایره -> شعاع دایره	یک دایره را مشخص میکند	<circle></circle>
CX -> مختصات افقی مرکز بیضی Cy -> مختصات عمودی مرکز بیضی TX -> شعاع افقی بیضی ry -> شعاع عمودی بیضی	یک بیضی را مشخص میکند	<ellipse></ellipse>
x1 -> مختصات افقی نقطه شروع خط y1 -> مختصات عمودی نقطه شروع خط x2 -> مختصات افقی نقطه پایان خط y2 -> مختصات عمودی نقطه پایان خط	یک خط را مشخص میکند	<line></line>
points -> مختصات نقاط چندضلعی به صورت زوج مرتب به عنوان مثال برای مشخص کردن نقاط یک مثلث قائم الزاویه و متساوی الساقین با اضلاعی به طول ۱۰ مشخصه points به صورت زیر خواهد بود: points="0,0 10,0 0,10"	یک چند ضلعی را مشخص میکند	<polygon></polygon>
x، y -> مختصات متن همچنین متن، به عنوان دادهی این تگ وارد میشود. به عنوان مثال: <text>Hello World!</text>	یک متن را مشخص میکند	<text></text>
xlink:href -> آدرس نسبی فایل عکس را مشخص میکند - x, y, width, height -> محل عکس و طول و عرض آن	یک عکس را مشخص میکند	<image/>
این تگ مجموعهای از attributeها مانند x، y، stroke، fill و غیره را میپذیرد. برای کشف نحوهی عملکرد این attributeها روی این تگ، آنها را در یک فایل ساده svg امتحان کنید.	برای گروه بندی استفاده میشود	<g></g>

جدول ۲ _ برخی از ATTRIBUTEهای عمومی

توضيحات	نحوه استفاده	عملكرد	مشخصه	
در صورتی که مقدار این مشخصه none باشد، دورخط رسم نمیشود	stroke="black"	رنگ دورخط را مشخص میکند	stroke	
	stroke-width="3"	سایز دورخط را مشخص میکند	stroke-width	
	fill="red"	رنگ شکل را مشخص میکند	fill	
نیازی به استفاده از فونتهایی غیرپیش فرض نیست. به عنوان مثال ،Courier، Arial است Impact، Times بعضی از فونتهای پیشفرض (که در اکثر کامپیوترها شناخته میشوند) هستند.	font-family="Verdana"	نوع فونت متن را مشخص میکند	font-family	
	font-size="35"	سایز فونت متن را مشخص میکند	font-size	
مشخصه transform کاربردهای متفاوتی در تغییر اشکال دارد. در این تمرین شما تنها مجاز به استفاده از امکان rotate این مشخصه هستند و استفاده از سایر امکانات این مشخصه همانند scale و translate و ممنوع است.	transform="rotate(45)"	المان را به اندازه مشخص شده حول مرکز میچرخاند	rotate transform	
این مشخصه مقداری بین ۰ تا ۱ را می پذیرد. مقدار یک به معنی شفافیت صفر و مقدار صفر به معنی شفافیت حداکثری (ناپدید شدن) است.	opacity="0.5"	مقدار شفافیت جسم را مشخص میکند.	opacity	

آشنایی با Canvas

Canvas یا بوم نقاشی یک تگ مخصوص در زبان HTML است که در آن به صورت آزادانه میتوان اشکال را رسم نمود. بر خلاف SVG، در بوم تصاویر به صورت شطرنجی نمایش داده می شود. علاوه بر این، برخلاف SVG که در آن تصاویر توصیف می شدند، در بوم شما با اجرای ترتیبی فرامین تصویر را رسم می کنید.

موارد استفاده از بوم به مراتب کمتر از موارد استفاده از SVG است، اما پیش بینی می شود که با گسترش بیشتر WebGL، محبوبیت استفاده از بوم به طرز قابل توجهی افزایش یابد. برای استفاده از تمام قابلیتهای بوم باید با زبان جاوا اسکریپت آشنا بود، اما شما در این تمرین تنها دستورهایی که بیان شدهاند را تولید نموده و نیازی نیست که به جاوا اسکریپت آشنا باشید. همچنین خروجیای که برنامهی شما در قالب بوم ارائه می دهد، از جزئیات کمتری نسبت به خروجی SVG شما برخوردار است.

ساختار کلی

همانطور که گفته شد، بوم یک تگ از زبان HTML است. بنابراین خروجی شما باید یک فایل HTML باشد. برای ساخت این فایل از قالب زیر استفاده کنید:

```
<!DOCTYPE html>
<html>
<html>
<body>

<canvas id="myCanvas" width="XX" height="XX" style="border:1px solid #d3d3d3;">
Your browser does not support the canvas element.
</canvas>

<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");

Content Here ...

</script>

</body>
</html>
```

در این قسمت نیز شما نیازی به فهمیدن و دانستن معنی تگهای ابتدایی ندارید و تنها لازم است که محتوایی که تولید میکنید را به جای Content Here قرار دهید. توجه داشته باشید که هنگام تولید این فایل، باید اندازه بوم را تعیین کنید. برای انجام اینکار ابتدا باید اندازه تصویر رسم شده توسط کاربر را بدانید، سپس به جای height و width (که در قالب با رنگ آبی مشخص شدهاند) قرار دهید. همچنین در بوم نیز همانند SVG، مبدا مختصات بالا سمت چپ صفحه است.

دقت داشته باشید که در بوم، شما به وسیلهی اجرای فرامین تصویر خود را کامل میکنید. پس به عنوان مثال برای اینکه یک شکل را روی یک شکل دیگر رسم کنید، باید دستور رسم آنرا پس از دستور رسم شکل اول بیاورید. به مثال زیر توجه کنید. در این مثال مستطیل آبی روی مستطیل قرمز قرار خواهد گرفت، زیرا بعد از آن رسم شده است.

```
ctx.fillStyle = "red";
ctx.fillRect(0,0,150,75);
```

```
ctx.fillStyle = "blue";
ctx.fillRect(20,20,150,75);
```

در ادامه روشهایی برای رسم اشکال به وسیله بوم را ارائه می دهیم. همانطور که قبلا نیز گفته شد، به وسیلهی این روشها، از شما انتظار نمی رود دقیقا همان خروجی SVG را ایجاد کنید. به عنوان مثال رسم عکس، رسم بیضی، چرخاندن اشکال و تغییر شفافیت اشکال از امکاناتی است که نیازی به پیاده سازی و رعایت آنها در خروجی بوم نیستید. بنابراین، خروجی بوم شما، خروجی ناکاملی خواهد بود. اما همچنان شما را به استفاده از گوگل به هنگام مشکلات خود و یا زمانی که خروجی مورد نظر خود را بدست نمی آورید، تشویق می کنیم. همچنین لینک های زیر نیز می توانند مفید واقع شوند.

- https://www.w3schools.com/graphics/canvas intro.asp
- https://www.w3schools.com/graphics/canvas reference.asp

رسم خط

```
ctx.beginPath(); // شروع رسم یک شکل جدید
مختصات نقطه شروع خط
ctx.moveTo(0,0); // مختصات نقطه پایان خط
مختصات نقطه پایان خط
ctx.stroke(); // نهایی کردن رسم خط
```

برای تغییر ضخامت و رنگ خط رسم شده دو دستور زیر را قبل از ()ctx.stroke قرار دهید. دقت کنید که این دستوراها مقدار ضخامت و رنگ تمامی خطوطی که بعد از آنها رسم می شود را تغییر می دهند. بنابراین بهتر است که مقدار آنها را بعد از اجرا دستور ()ctx.stroke دوباره به مقدار پیش فرض برگردانیم.

```
ctx.lineWidth=5;
ctx.strokeStyle="red";
```

رسم مستطيل

برای رسم مستطیل توپر به صورت زیر عمل کنید:

```
ctx.fillStyle = "yellow";
ctx.fillRect(20,20,150,75);
```

دقت کنید که دستور fillStyle نیز همانند دو دستور قسمت قبلی، رنگ تمامی اشکالی که بعد از اجرای آنها رسم می شود را تغییر می دهد. بنابراین مقدار آن را پس از اجرای دستور fillRect به مقدار پیش فرض برگردانید. در صورتی که مستطیل شما دارای stroke است از دستور زیر استفاده کنید.

```
ctx.lineWidth=3;
ctx.strokeStyle="brown";
ctx.strokeRect(20,20,150,100);
```

رسم چندضلعی

```
ctx.beginPath();
ctx.lineWidth=2;
```

```
ctx.strokeStyle="black";
ctx.moveTo(10,10);
ctx.lineTo(10,30);
ctx.lineTo(30,30);
ctx.lineTo(10,10);
ctx.fillStyle = "yellow";
ctx.fill();
ctx.stroke();
```

بدیهی است که در صورتی که چند ضلعی فاقد دورخط باشد، رنگ دورخط آن را همان رنگ چندضلعی در نظر باید گرفت.

رسم دایره

```
ctx.beginPath();
ctx.lineWidth=1;
ctx.strokeStyle="blue";
ctx.arc(95,50,40,0,2*Math.PI);
//.مان آخر دستور قبل ثابت هستند و رسم یک دایره ی کامل را بیان می دهند.
ctx.fillStyle = "white";
ctx.fill();
ctx.stroke();
```

رسم نوشتار

```
ctx.font = "30px Arial";
ctx.fillText("Hello World",10,50);
```

اگر نوشته شما شامل دورخط باشد، می توانید از دستور زیر استفاده کنید:

```
ctx.strokeText("Hello World10,50,");
```