

2017-11-21

zer da SVG bat?

Irudi bektorial bat da, hau da, kalkulu matematikoez **nabigatzaileak interpretatu, eskalatu eta aldatu** dezakeen irudi edo irudi sorta formatua. **Illustrator** edo **InkScape** bezalako programekin sor daitezke SVGak. Pixel formatuko irudiak ez bezala (JPEG, PNG...), hauek nahi adina eskala daitezke kalitaterik galdu gabe.

Webgunez ari garenean, aipatu beharreko **gai nagusietako bat irudien trataera da**. Webguneen moteltasuna irudien **optimizazio txarraren** ondorio nagusia izaten da. Baina zer egingo genuke irudirik gabe webgune batean? Hain zuzen ere, webguneen atal grafikoa askotan errekurtsio bakarra izaten da. Zer esanik ez GIF-etaz, blog batean nagusi ez izan arren, Twitter edo Telegram bezalakoek ez lukete zentzurik izango horiek gabe.

Eta esaten badizut **SVG animazio batek** irudi normal baten **%10a bakarrik pisatzen duela?** Normalean irudi bakoitzak gehienez 100kb inguruko pisua izan beharko luke "optimizazio egokia" duela esateko. SVG animazio batek 10-20Kb inguru izango lituzke, nahiz eta ez egon egoki optimizatua.

Artikulu honetan erakutsiko dizuet nola sartu eskua **SVG** batean **XML eta CSS ezagutza pixka batekin**. Horretarako tanit.eus webgunerako sortu ditudan animazioak erabiliko ditugu.

PNG formatuan



SVG formatuan

01 SVG Prestaketa (XML)

Hasteko ilustrazio bat egin edo aukeratuko dugu, eta aukeratu ondoren kode editore batekin irekiko dugu. Nire kasuan, nik sortu dudana goiko antena erabiliko dut.

Ilustrazio horren kodea hurrengoa da:

```
<svg xmlns="https://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" id="Capa_1" data-name="Capa 1"
viewBox="0 0 300 300">
  <defs>
    <style>
      .cls-1,.cls-4{ fill:none;}
      .cls-1,.cls-4,.cls-5{ stroke:#2f3a39;stroke-miterlimit:10;}
      .cls-2{ clip-path:url(#clip-path);}
      .cls-3{ fill:#2f3a39;}
      .cls-4,.cls-5{ stroke-width:10px;}
      .cls-5{ fill:#fff;}
    </style>
```

2017-11-21

```
<clipPath id="clip-path">
  <circle class="cls-1" cx="150" cy="150" r="150"/>
</clipPath>
</defs>
<title>4.0</title>
<g class="cls-2">
  <path class="cls-3"
d="M150,10a140,140,0,0,1,99,239A140,140,0,1,1,51,51a139.08,139.08,0,0,1,99-41
m0-10A150,150,0,1,0,300,150,150,0,0,0,150,0h0Z"/>
  <g class="base">
    <line class="cls-4" x1="203.66" y1="191.56" x2="202.98"
y2="286.25"/>
    <line class="cls-4" x1="220.8" y1="172.42" x2="267.08" y2="238.91"/>
    <line class="cls-4" x1="203.02" y1="280.6" x2="244.48" y2="205.66"/>
    <line class="cls-4" x1="203.66" y1="191.56" x2="244.48"
y2="205.66"/>
    <line class="cls-4" x1="267.08" y1="238.91" x2="203.36"
y2="233.34"/>
    <line class="cls-4" x1="220.8" y1="172.42" x2="203.36" y2="233.34"/>
  </g>
  <g class="antena">
    <path class="cls-5"
d="M232.09,97.76c29.92,31.42,13.7,65.38-17.72,95.3s-66.14,44.45-96.05,13c-9-9
.44,9.19-41.35,40.6-71.27S223.1,88.32,232.09,97.76Z"/>
    <ellipse class="cls-5" cx="175.21" cy="151.93" rx="78.55" ry="23.61"
transform="translate(-56.45 162.73) rotate(-43.6)"/>
    <circle class="cls-5" cx="174.41" cy="151.09" r="10.34"
transform="translate(-56.09 161.95) rotate(-43.6)"/>
    <line class="cls-5" x1="181.55" y1="158.59" x2="188.99"
y2="171.41"/>
    <line class="cls-5" x1="181.55" y1="158.59" x2="193.99"
y2="166.65"/>
  </g>
  <g class="lora-1">
    <ellipse class="cls-4" cx="135.72" cy="113.57" rx="41.02" ry="10.02"
transform="translate(-40.55 129.23) rotate(-45)"/>
  </g>
  <g class="lora-2">
    <ellipse class="cls-4" cx="104.44" cy="84.06" rx="79.47" ry="11.21"
transform="translate(-28.85 98.47) rotate(-45)"/>
  </g>
</g>
<circle class="cls-1" cx="150" cy="150" r="150"/>
</svg>
```

Nahasgarria dirudi, baina lasai, joango gara **atalak identifikatzen**. XML lengoia erabiltzen du marrazkia egituratzeko. Ingeles pixka bat jakinda antzeman ditzakegu **<path>**, **<ellipse>**, **<line>**... bezalako etiketak. Horietan batzen dira marrazkiaren koordenadak. Lerro, elipse edo forma horiek **<g> eta </g> artean sartuta daudela** ikusten duzue. Horiek dira lerro edo **forma taldeak (groups)**. Gure marrazkia sortzean formak taldekatzen baditugu, lan hau aurreztuko dugu gerora. Bestela, ireki eta irakurriz identifikatu behar ditugu taldekatu nahi ditugun formak eta eskuz

 etiketen artean sartu.

Zertarako taldekatzen ditugu? Animazio aginduak irudi multzoei batera emateko. Adibidez, nik antenaren forma zirkularra eta barnean duen elipsea bakarrik mugitu nahi ditut. **Illustrator programan horiek batu ditut talde batean eta aparteko kapa batean jarri ditut "antena" izenarekin, kodearen 26. lerroan ikusi daitekeen bezala.** Honekin **<g> etiketak "antena" klasea hartu du.** Berdina gertatu da **33 eta 36 lerroetan ikusi daitezkeen "lora-1" eta "lora-2" klaseekin.** Hauek izango dira animatuko ditugun atalak.

2 Animazioa (CSS)

Hasteko, prest dugu SVGa. Antena, eta bi uhinak animatzeko **CSS klaseak zehaztuak ditugu.**

Animazioa egiteko **@keyframes** deklarazioa erabiliko dugu, eta **animazioaren atal bakoitzean gertatuko dena zehaztuko dugu ehunekoetan.** @keyframes ondoren nahi dugun **izena jarriko diogu animazioari, nire kasuan "rotate".**

<g class="antena"></g> barruan dagoena animatzen hasiko gara. CSS-an klaseari aginduak emateko aurretik puntu bat idatziko dugu (**.antena**) eta agindu nahi dioguna kortxeteen (**{ }**) artean idatziko dugu **“;”** batez banatua. Kasu honetan **transform:rotate(Xdeg);** erabiliko dugu **errotazio bat zehazteko animazioan.** Baina transformazio hori **antenaren zentrotik gertatzeko,** antenari **transform-origin:50%;** esango diogu. **Kodean bertan idatziko dizkizuet aginduen azalpenak. bestela [hemen duzue CSS "transform" etiketaren dokumentazioa.](#)**

```
@keyframes rotate{
  0%{ /** Animazioaren hasiera **/
    transform:rotate(0deg); /** Rotazioa 0n kokatzen dugu **/
  }
  50%{ /** Animazioaren erdia **/
    transform:rotate(10deg); /** 10 gradu mugituko da **/
  }
  100%{ /** Animazioaren bukaera **/
    transform:rotate(0deg); /** Berriro 0ra itzuliko da **/
  }
}

.antena{
  transform-origin:50%; /** Bere zentrotik egingo du rotazioa**/
  animation: rotate 2s infinite ease-in-out; /** Animazioan segidan jartzen
dira: animazioaren izena, iraupena,                               errepikapena, efektua (sarreran
eta irteeran moteltzea). **/
}
```

Emaitza hau izango litzateke:

Goazen, uhinak animatzera.

2017-11-21

```
@keyframes lora{
  0%{
    opacity:0; /** Transparentzia balorea - 0 = 0% / .5 = 50% / 1 = 100%
  **/
    transform:translateY(0) translateX(0); /** X eta Y koordenadetan zenbat
  mugituko diren **/
  }
  50%{
    opacity:1;
    transform:translateY(-80px) translateX(-80px);
  }
  100%{
    opacity:1;
    transform:translateY(-190px) translateX(-190px);
  }
}

.lora-1{
  animation: lora 2s infinite ease-in;
  animation-delay: .2s; /** Animazioa egiten hasteko itxarongo duen denbora
**/
}
.lora-2{
  animation: lora 2s infinite ease-in;
}
```

Eta horrekin kito! Ondorengo emaitza edukiko genuke:

Hau oso sinplea izan da!

Zer gertatuko litzateke zerbait konplexuagoa egin nahiko bagenu?

Zabaldu:

- [Click to share on Twitter \(Opens in new window\)](#)
- [Click to share on Facebook \(Opens in new window\)](#)
- [Click to share on Telegram \(Opens in new window\)](#)
- [Click to share on WhatsApp \(Opens in new window\)](#)
- [Click to share on Pocket \(Opens in new window\)](#)