



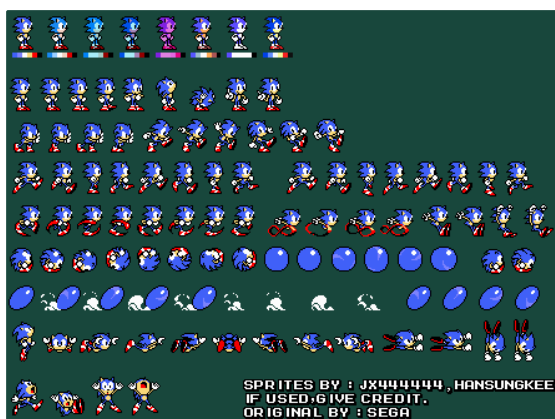
Sprite Image Technique

SSAFY 특강 : UI / UX / Sprite Image Technique

▶ 목차

Sprite Image

스프라이트 이미지는 여러 이미지를 한 장의 이미지에 배치하는 것을 말합니다. 게임 그래픽 디자인에서 자주 사용되던 방법입니다. Sonic 스프라이트 이미지(좌), 스프라이트 애니메이션 제작 도구(우)

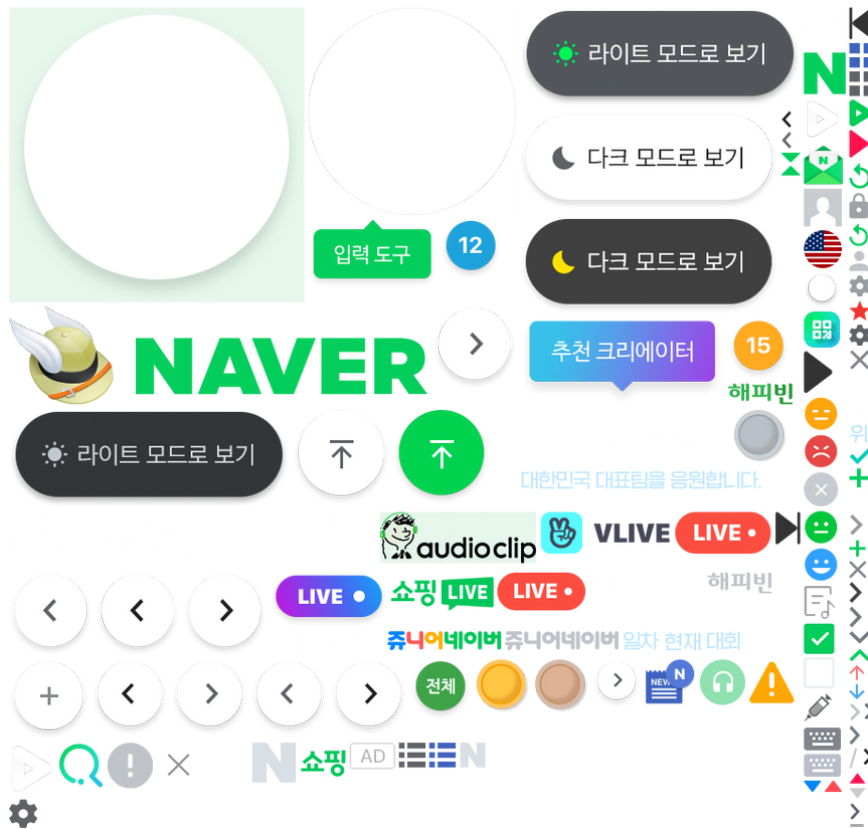


<https://bit.ly/3B0iJoo>



<https://www.aseprite.org>

웹 디자인에서 또한 서버 요청 횟수를 효과적으로 줄일 수 있어 UI를 구성하는 그래픽을 한데 묶어 사용합니다.



NAVER 포털 사이트에서 사용 중인 스프라이트 이미지 (비트맵)

이와 같이 수많은 이미지를 한 장에 포함 배치시키면 서버에 수십번 요청하는 대신, 단 한 번의 요청으로 서비스에 모든 이미지를 사용할 수 있어 매우 효과적입니다. 다만, 이 방법의 단점은 이미지의 픽셀 x, y 좌표 위치 값을 알아야 하고 표시할 이미지 크기에 맞춰 너비와 높이 값을 변경해야 합니다.

```
#gnb .list_nav .ico_mail { display: inline-block; width: 21px; height: 21px; background-position: -420px -73px; background-repeat: no-repeat; vertical-align: top; margin: 3px 9px 0 0; }
```

참고

Aseprite

Animated sprite editor & pixel art tool



<https://www.aseprite.org/>



Piskel - Free online sprite editor

Piskel, free online sprite editor. A simple web-based tool for Spriting and Pixel art. Create pixel art, game sprites and



<https://www.piskelapp.com/>



Sprite Cow

Sprite Cow helps you get the background-position, width and height of sprites within a spritesheet as a nice bit of copyable css. Load the example image, and have a click around. It becomes pretty obvious.



<http://www.spritecow.com/>

Retina CSS Sprite Generator

Sprites reduce bandwidth used to request images from a server and help speed up your page load time by combining icons, logos, and other static images into a single file. Retina and other double density pixel



<https://jcthepants.github.io/Retina-CSS-Sprite-Generator/>

SVG Sprites

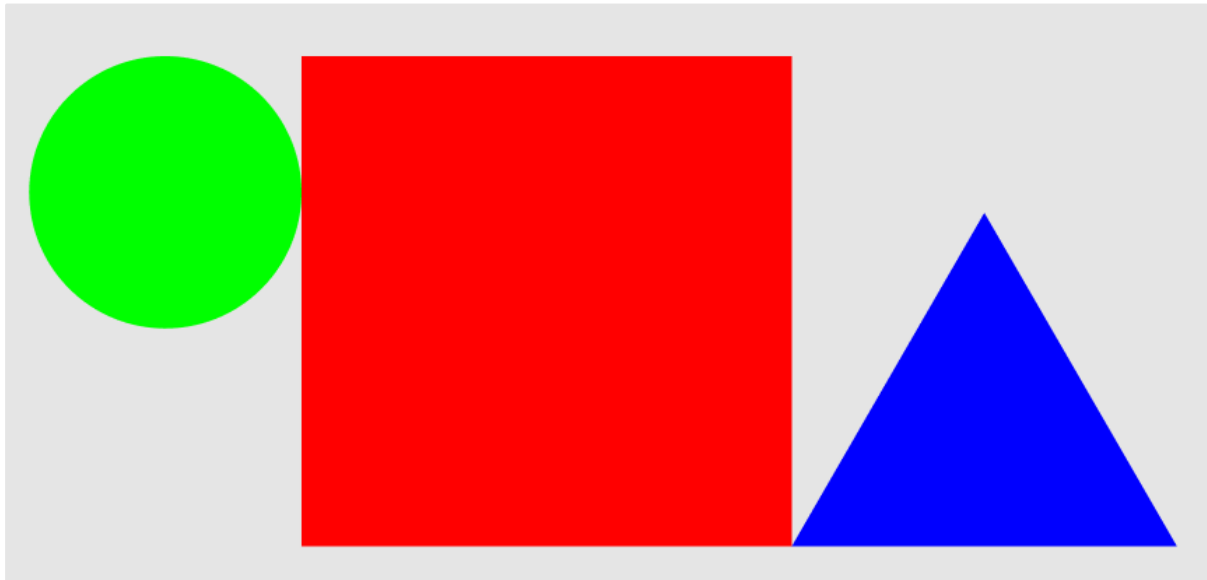
SVG는 다음의 이유로 로고 및 아이콘, 다이어그램에 이상적입니다.

- 클라이언트, 서버 환경 어디서나 손쉽게 SVG를 만들 수 있습니다.
- 비트맵 포맷과 달리, SVG는 품질 저하 없이 모든 크기로 확장 가능합니다.
- 웹 폰트 아이콘과 달리, SVG는 선명하고 여러 색상 및 그레디언트 또는 필터를 활용할 수 있습니다.

SVG 스프라이트 이미지

SVG 또한 비트맵 스프라이트 이미지처럼 여러 SVG 포맷의 이미지를 단 하나의 SVG 파일에 배치하여 사용할 수 있습니다. 차이점은 각 SVG 이미지의 픽셀 위치가 아닌, ID 참조로 표시합니다.

가장 일반적으로 사용되는 방법은 SVG `<symbol>` 요소 안에 개별 이미지를 정의하는 것입니다. 예를 들어, 다음의 SVG 스프라이트 이미지는 녹색 원(circle), 빨간색 사각형(square), 파란색 삼각형(triangle)이 포함되어 있습니다.



```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg"> <symbol
id="circle"> <circle cx="50" cy="50" r="45" stroke-width="5"
stroke="#0f0" fill="#0f0" fill-opacity="0.5" /> </symbol> <symbol
id="square"> <rect y="5" x="5" width="90" height="90" stroke-width="5"
stroke="#f00" fill="#f00" fill-opacity="0.5" /> </symbol> <symbol
id="triangle"> <path d="M20,7 L92,50 L6,93 z" stroke-width="5"
stroke="#00f" fill="#00f" fill-opacity="0.5" /> </symbol> </svg>
```

SVG `<use>` 요소를 사용해 SVG 스프라이트 이미지의 ID를 참조해 이미지를 화면에 표시할 수 있습니다.

```
<svg width="100" height="100"> <use href="./spriteuse.svg#circle" />
</svg> <svg width="100" height="100"> <use href="./spriteuse.svg#square"
/> </svg> <svg width="100" height="100"> <use
href="./spriteuse.svg#triangle" /> </svg>
```

다만 이 방법은 IE와 같은 구형 브라우저에서 정상적으로 작동하지 않습니다. 가장 좋은 해결 방법은 SVG 스프라이트 이미지 코드를 HTML 문서에 포함하고 각 이미지의 ID를 참조해 사용하는 것입니다.

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg"> <symbol
id="circle"> <circle cx="50" cy="50" r="45" stroke-width="5"
stroke="#0f0" fill="#0f0" fill-opacity="0.5" /> </symbol> </svg>
```

```
<svg width="100" height="100"> <use href="#circle" /> </svg>
```

SVG 스프라이트 스택(Stack)

앞서 살펴본 `<use>` 요소는 `<svg>` 안에서만 사용 가능합니다. 그러므로 위 방법은 ``, `<iframe>`, `<object>` 요소로 사용할 수 없고 CSS 배경 이미지로도 사용할 수 없습니다.

2012년 [@simurai](#), [@erikdahlstrom](#)은 위에서 거론한 제약 사항을 우회하는 SVG 스택 기술을 소개합니다.

이 방법은 개별 이미지에 `"sprite"` 클래스 이름을 설정하고 화면에 표시되지 않도록 `display: none` 설정합니다. 하지만 ID가 참조될 경우 화면에 인라인 요소로 표시되도록 `display: inline` 설정하는 것입니다.

```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg"> <defs>
<style><![CDATA[ .sprite { display: none; } .sprite:target { display:
inline; } ]]></style> </defs> <g class="sprite" id="circle"> <circle
cx="50" cy="50" r="45" stroke-width="5" stroke="#0f0" fill="#0f0" fill-
opacity="0.5" /> </g> <g class="sprite" id="square"> <rect y="5" x="5"
width="90" height="90" stroke-width="5" stroke="#f00" fill="#f00" fill-
opacity="0.5" /> </g> <g class="sprite" id="triangle"> <path d="M20,7
L92,50 L6,93 z" stroke-width="5" stroke="#00f" fill="#00f" fill-
opacity="0.5" /> </g> </svg>
```

ID 참조는 아래 코드에서처럼 `` 요소에 SVG 스프라이트 이미지 경로를 설정하고 해시(`#`) 뒤에 ID 이름을 설정합니다. (`#target-name`)

```

```

이 방법은 CSS 배경 이미지로도 활용할 수 있습니다.

```
.sprite-image { background: url('./sprite.svg#circle'); }
```

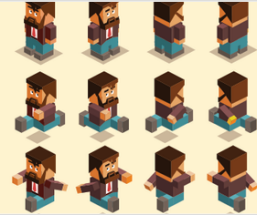
IE 9+ 에서도 이 방법은 잘 작동합니다.

참고

How to Use SVG Image Sprites - SitePoint

This tutorial describes two ways to create sprites containing many images. However, it uses SVG rather than the more well-

➔ <https://www.sitepoint.com/use-svg-image-sprites/>



SVG Sprites 자동 생성

명령어를 사용해 프로젝트 디렉토리를 생성하고, 생성된 디렉토리로 이동합니다.

```
mkdir auto-generate-svg-sprites && cd $_
```

mkdir(1) - Linux man page

mkdir - make directories Create the DIRECTORY(ies), if they do not already exist.

📄 <https://linux.die.net/man/1/mkdir>

기본 제공 함수 정보 - PowerShell

PowerShell의 기본 제공 함수에 대해 설명합니다. PowerShell에는 모든 PowerShell 세션에 로

📄 <https://docs.microsoft.com/ko-kr/power...>

📢 [oh-my-zsh](#) 사용자는 `take` 명령을 사용해 디렉토리 생성, 이동이 보다 손쉽습니다.

```
take auto-generate-svg-sprites
```

Git 초기화

main 브랜치 이름으로 Git 초기화 명령을 실행합니다.

```
git init -b main
```

Git - git-init Documentation

q --quiet Only print error and warning messages; all other output will be suppressed. --bare Create a bare repository. If GIT_DIR environment is not set, it is set to the current working directory. --object-format=

 <https://git-scm.com/docs/git-init>

NPM 초기화

초기화 NPM 명령을 실행해 프로젝트 루트 위치에 **package.json** 파일을 생성합니다.

```
npm init -y
```

npm-init | npm Docs

npm init can be used to set up a new or existing npm package. initializer in this case is an npm package named create- , which

 <https://docs.npmjs.com/cli/v8/commands/npm-init>



종속성 모듈 설치

“SVG 스프라이트 이미지 자동 생성”에 필요한 패키지를 설치합니다.

```
npm i -D svg-sprite minimist chokidar live-server rimraf npm-run-all
```

svg-sprite

SVG sprites & stacks galore - A low-level Node.js module that takes a bunch of SVG

 <https://www.npmjs.com/package/svg-sp...>

minimist

parse argument options This module is the guts of optimist's argument parser without all

 <https://www.npmjs.com/package/minimist>

chokidar

Minimal and efficient cross-platform file watching library Node.js fs.watch: Doesn't

 <https://www.npmjs.com/package/chokidar>

rimraf

The UNIX command `rm -rf` for node. Install with `npm install rimraf`, or just drop `rimraf.js`

 <https://www.npmjs.com/package/rimraf>

live-server

simple development http server with live reload capability. Latest version: 1.2.2, last

 <https://www.npmjs.com/package/live-se...>

npm-run-all

A CLI tool to run multiple npm-scripts in parallel or sequential.. Latest version: 4.1.5, last

 <https://www.npmjs.com/package/npm-r...>

디렉토리 구성

디렉토리를 다음과 같이 구성합니다.

```
. |─ public/ | |─ assets/ | |─ index.html |─ src/ | |─ svgs/ |─
package-lock.json |─ package.json
```

자동 생성 스크립트 작성

SVG 스프라이트 자동 생성 코드를 작성합니다.

`index.js`


```
const chokidar = require('chokidar'); const { exec } =
require('child_process'); const log = console.log.bind(console); const
argv = require('minimist')(process.argv.slice(2)); const INPUT_DIR =
argv.i || argv.input || 'src/input'; const OUT_DIR = argv.o ||
argv.outdir || 'src/output'; const FILENAME = argv.f || argv.filename ||
'sprites'; async function generateSvgSprite() { await exec(`rimraf -rf
${OUT_DIR}`); await exec(`npx svg-sprite --svg-xmldecl=false --svg-
doctype=false -s --symbol-prefix='' --symbol-sprite='${FILENAME}' --
symbol-bust=false --symbol-example=false --symbol-dest=${OUT_DIR}
`${INPUT_DIR}/**/*.svg` ); } generateSvgSprite(); const watcher =
chokidar.watch(INPUT_DIR, { persistent: true }); watcher .on('add',
(path) => { log(`${path} 파일 추가`); generateSvgSprite(); })
.on('change', (path) => { log(`${path} 파일 수정`); generateSvgSprite(); })
.on('unlink', (path) => { log(`${path} 파일 제거`);
generateSvgSprite(); });
```

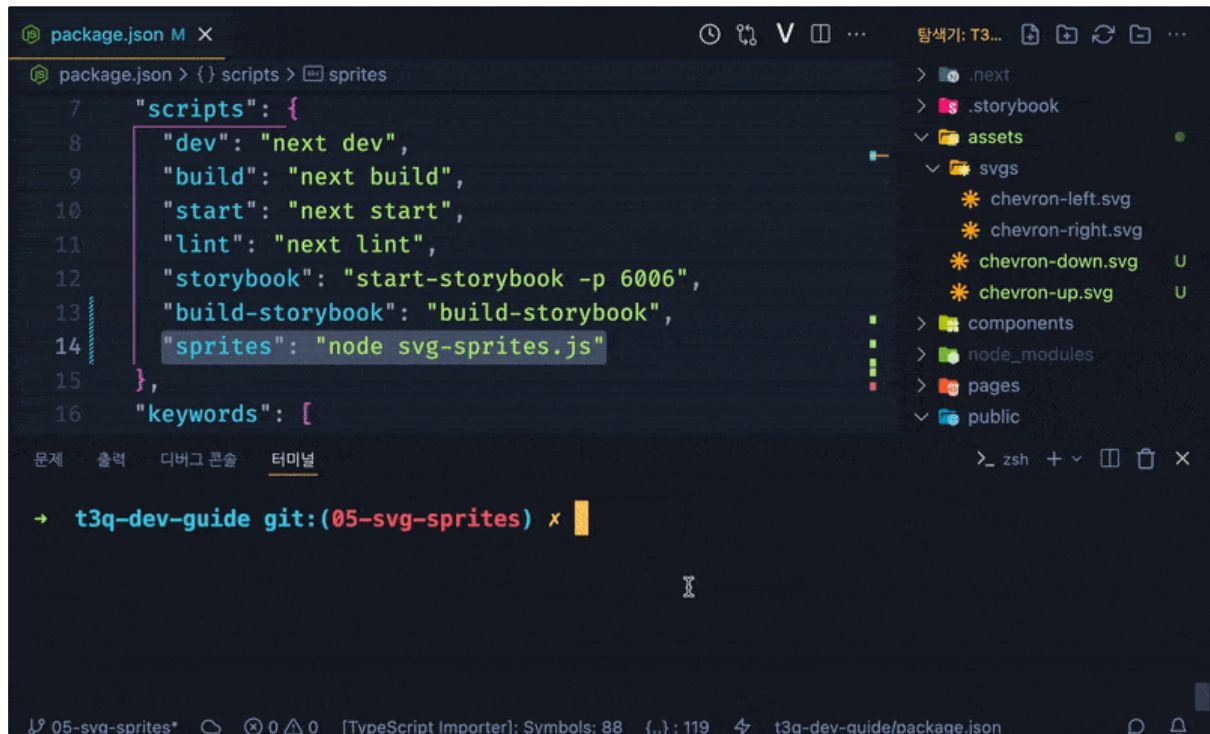
NPM 스크립트 명령 작성

`package.json` 파일에 NPM 스크립트 `sprites` 명령을 추가합니다.

`package.json`

```
{ "scripts": { "dev": "run-p sprites serve", "serve": "live-server public
--host=localhost --port=3000 --no-browser --cors", "sprites": "node
index.js -i src/svg -o public/assets/sprites -f svg-sprites" }, }
```

등록된 명령을 실행하면 자동으로 SVG 스프라이트 이미지를 생성하고, 인풋 디렉토리가 변경(추가, 수정, 제거)되면 이를 감지해 자동으로 다시 스프라이트 이미지를 생성합니다.



참고

실습 완성 파일을 다운로드 받아 확인해보세요.

 auto-generate-svg-sprites.zip 123.4KB

COPYRIGHT RESERVED. 2022 @ EUID

