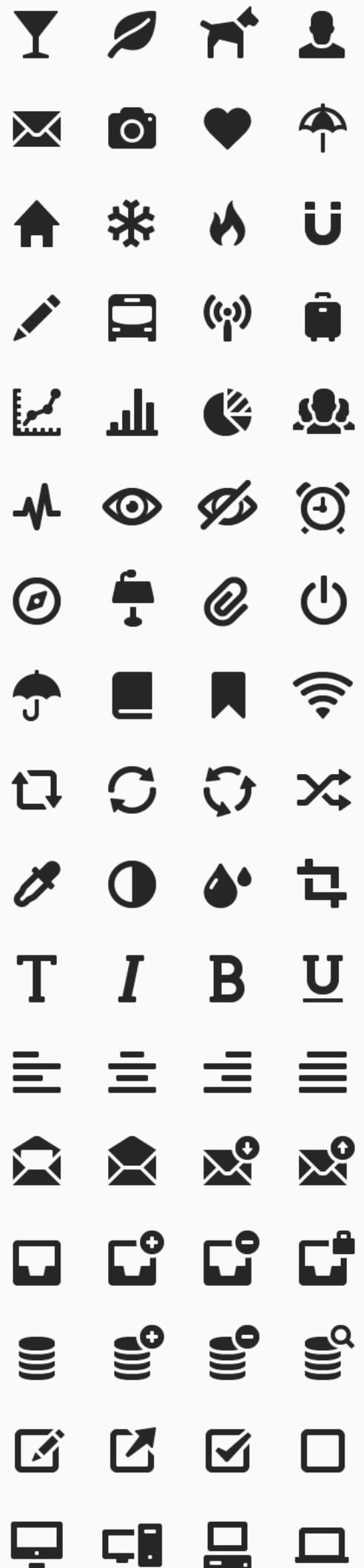


# GLYPHICONS®

## Handbook

v 2.0.1 from April 6, 2019

Get all essential information on working with icons, including practical examples and tips.



Any reproduction of this handbook in part or in whole without written permission is prohibited.

You can always find the current version at:

[glyphicons.com/handbook](https://glyphicons.com/handbook)

Written by Jan Kovařík.

© 2010 - 2019 GLYPHICONS® All rights reserved.

GLYPHICONS® is a registered trademark of GLYPHICONS.com s.r.o.

# Contents

Preface by Jan Kovařík .....	4
GLYPHICONS file structure .....	5
GLYPHICONS anatomy .....	6
How can this work for you .....	7
GLYPHICONS on the desktop .....	8
Desktop icon font and its purpose .....	9
How actually icon font works .....	9
Installing desktop icon font .....	10
How to use font in your desktop app .....	11
Pros and cons of using icon fonts in desktop apps .....	13
GLYPHICONS on the web .....	14
Where are icon fonts for the web .....	14
SVG it is .....	16
SVG embedding methods .....	17
SVGs and colors .....	20
Troubleshooting and tips .....	24
FAQ .....	25

## Preface by Jan Kovařík

GLYPHICONS is a personal project I have been working on for almost a decade and it sent me down on a path of discovery not only of icons and design, but of people as well.

What began as a small set of universal icons and symbols for my daily work is now being used not only by independent designers and developers around the world, but also by many government organizations and the world's largest companies.

If you use them in your designs, you already know that clients usually take them for granted, usually don't even notice them and the same applies for users. While I think that's the best reaction at all, it may be a little unsatisfactory for the designer. But it's even more rewarding in the long term, when it's proven that your design works as intended.

Of course these icons aren't a magic cure-all for every design and if you want to make the most of it, it's also very important to understand how they are made and follow (or occasionally brake) some basic design rules when using them.

I hope you will enjoy this updated version of Handbook, which should help you using them as effectively as possible by better understanding what's behind these icons.

# GLYPHICONS file structure

Each set contains the very same file structure and file naming logic as the Basic set in the example below.

glyphicons_basic
ai
glyphicons-basic.ai
glyphicons-basic@2x.ai
glyphicons-basic@3x.ai
desktop-font
glyphicons-basic-regular.otf
eps
glyphicons-basic.eps
pdf
glyphicons-basic.pdf
individual-pdfs
glyphicons-basic-1-glass.pdf
glyphicons-basic-2-leaf.pdf
glyphicons-basic-3-dog.pdf
...
paged-pdf
glyphicons-basic.pdf
png
glyphicons-basic-1-glass.png
glyphicons-basic-1-glass@2x.png
glyphicons-basic-1-glass@3x.png
...
svg
glyphicons-basic.svg
individual-svg
glyphicons-basic-1-glass.svg
glyphicons-basic-2-leaf.svg
glyphicons-basic-3-dog.svg
...

Files ending with a name of the set contain always all icons in one file.

Files ending with an icon name contain only one individual icon, who would have said? :)

Suffixes @2x and @3x in file names mean twice or three times the size of the original.

## New

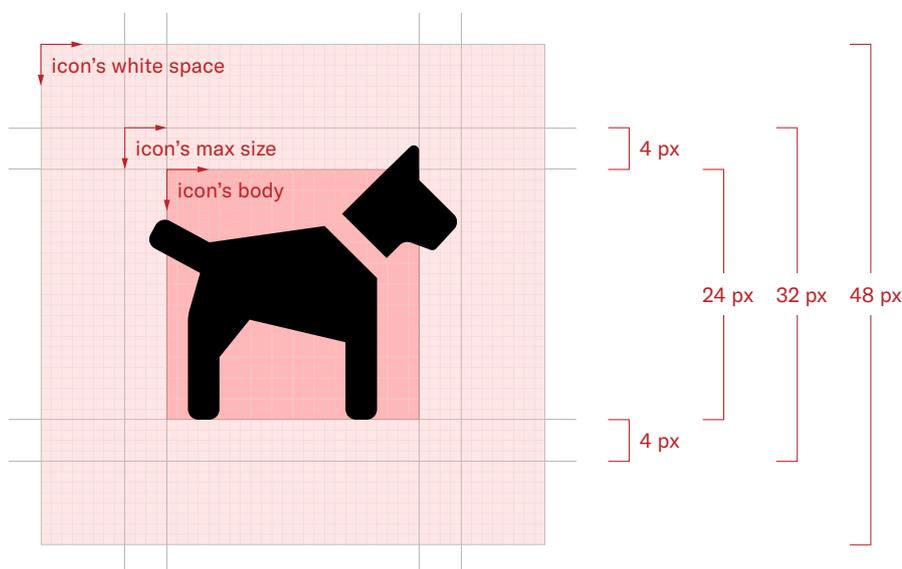
Each icon always has its own unique serial/order number, but this numbers is now displayed also under the name of the each icon on the website. Searching for icons in your files is now really simple, as you may use just a number as a reference.

To stay organized, keep this file structure and also don't change unnecessarily original names of these files in your projects. It will be much easier for you to find what you're looking for when using a search on [www.glyphicons.com](http://www.glyphicons.com) or to update these files, when the new version is available.

# GLYPHICONS anatomy

The construction of icons isn't a rocket science and once you learn how they are made, you'll be able to place icons correctly without even thinking about it. While there is definitely more to it, to use them in your design quickly and correctly, it's enough to know these three basic boundaries:

1. **icon's body** (24x24 px)
2. **icon's max size** (32x32 px)
3. **icon's white space** (48x48 px)



Following example is for 24 px set (such as the Basic set), but it also applies to all other sets, except the Halflings set (which is smaller).

I'm using the same logic for 16 px sets (such as the Halflings set), only with 16 px as a body size, 20 px for max size and 32 px as a white space.

## Icon's body

Each icon is made in 24x24 px (icon's body) and then optically balanced and most of the icons have this approximate size. Why just "approximate"?

Icons are not always strictly squared or centered just for the sake of geometry or some grid, but rather sensitively adjusted to match their real life appearance and compared with other icons to ensure that they remain in the same style and as much seamless and interchangeable with other icons in the set as possible.

### Icon's max size

When it's convenient, icons use up to 4 px on each side, therefore some of them can have a slightly bigger physical size, up to 32x32 px (icon's max size). Thanks to this, various icons look and feel same next to each other, even though they have a slightly different dimensions or aspect ratio.

Also, with this size as a one uniform background, you can quickly drag & drop icons into your design, without having to re-align them individually. And I don't even mention situations where you need to make the same spacing between icons (typically with a menu or a list).

The best thing is that individual files (📁 PNG, 📁 SVG and 📁 PDF) are already exported and positioned on this background!

If you (for any reason) do not want to use this background, you can always re-export from source files, but I do not recommend it, as this space around and their placement in relation one to each other, is as important as the icons themselves and should be always preserved.

### Icon's white space

Last boundary is defined by 48x48 px square, a space around an icon which should always remain empty (icon's white space). This is the recommended minimum; all other visual elements should always remain outside this box.

## How can this work for you

That's it, no complicated or fancy grid system. So how this all is good for you and why is it clever to use these pre-made boundaries?

- ✓ All icons are already vertically and horizontally aligned, exactly where they should be in relation one to each other.
- ✓ It's super easy to replace one for another; because their background have exactly same size, changing an icon will be a breeze.
- ✓ You can easily use icons "folder" and "folder-open" to express various folder states, without having icons re-positioned.
- ✓ You can apply only one global rule for padding or spacing around icons and do not have to worry about their alignment.
- ✓ When you will respect icon's white space and let your design breathe, you can be sure that icons remain always legible.

# GLYPHICONS on the desktop

Each person uses icons differently and I can't go into details for every scenario. Also, which file format to choose on the desktop will depend on the application you use, on the output file format you need and not least on personal preferences.

As I'm getting questions about the meaning of certain file types again and again, I decided to write this short summary to explain, that what might seem useless file type to one, may be in fact crucial and fundamental to the other. Also, you never know when the file you never used before come in handy.

## When to use vector files

- Every time you need as an output file format print-ready files, it's convenient to use vectors (📄 Ai, 📄 PDF, 📄 SVG, 📄 EPS) which will also make your design indefinitely scalable.
- Even when you create something intended only for the desktop, it's better to use vector files in order to have everything sharp on various displays and platforms.
- If you're going to create multi-paged documents (magazines or books), you may use a desktop font (📄 OTF), so you can drag & drop glyphs in to your text, copy & paste them and apply various visual styles on them across the whole document.

You may find more tips on how to use a desktop icon font on the next page.

## When to use raster files

- You may want to use transparent raster images (📄 PNG) or retina-ready images for high-resolution screens with a higher pixel density in a scale factor 2.0 (📄 @2xPNG) or even scale factor 3.0 (📄 @3xPNG) for your iOS (or other) app.
- Even in 2019, you can still get a task to develop something for environment, which supports only classic raster images - it may be an obsolete browser, corporate intranet, older operating system, or simply as a fallback for SVGs.
- When you do not want to expose vector shapes (for various reasons) - typically when your work with a client which you do not know yet or your work is not paid upfront (here's actually smarter to work with vectors on your computer, but sent out just JPGs or PNGs as a preview of a the whole design).

## Desktop icon font and its purpose

I presume that plain vector or raster files will be the best choice for bigger part of my customers. You can quickly drag & drop files directly to your app. To work with individual files is probably much easier than installing and maintaining font files on the desktop, I can recommend to use these whenever possible.

However, there are areas on the desktop, where it is better to have the icons installed as a desktop-font and to have an option to work with them like a text. That's why all GLYPHICONS sets contain a free bonus – a desktop icon font with all icons.

This approach is especially useful when you create multi-paged publications, magazines, manuals or even books. For these, it's convenient to have icons directly in the text and fully control their appearance with text styles, from one place across the whole document (typically in Adobe InDesign).

## How actually icon font works

Regular font files contain letters and these appear as you type on your keyboard. Each letter has it's own pre-defined space inside a font file.

It isn't the same for icon fonts, as by the Unicode Consortium (industry standard for the consistent encoding) all non-standard characters (in our case icons and symbols) have to be placed in one of the three Private Use Areas ("PUA").

If you want to know more about Private Use Area, please visit: [www.en.wikipedia.org/wiki/Private\\_Use\\_Areas](http://www.en.wikipedia.org/wiki/Private_Use_Areas)

It is probably not necessary to dive deeper into more details, now it's important to realize, that we have these icons kind of hidden as characters, glyphs, inside this "PUA" and to display them is not as straightforward as typing a letter on a keyboard.

To use them, you can copy & paste icons directly from [www.glyphicons.com](http://www.glyphicons.com) or display all glyphs inside the font itself directly in your app - I presume that when you're creating multi-paged documents, your app can do that. At any rate, we will try both methods in the examples on the next page.

## Installing desktop icon font

Icon font should be installed exactly the same as a regular font in your OS, so if you know how to do it, you can skip to the next page.

**⚠ Before you start installing the font:** please be sure that you've uninstalled all previous versions of the same font, it's really important. Also you should close any open application.

The following example features the Basic set, but you may apply it for all other GLYPHICONS sets.

### How to install a font on your Mac

1. The font file:  ../desktop-font/glyphicons-basic-regular.otf
2. Double click on the font file and click on "Install Font"
3. Open your favourite desktop app, select the appropriate GLYPHICONS font, start to type somewhere and now you can paste the icon itself (press cmd+v) and voilà :)

The most convenient way to copy & paste icons in to your desktop app is to copy the glyph itself from any table of icons on [glyphicons.com](http://glyphicons.com). More about this method on the next page.

### How to install a font on Windows

1. The font file:  ../desktop-font/glyphicons-basic-regular.otf
2. Double click on the font file and click on "Install"
3. Open your favourite desktop app, select the appropriate GLYPHICONS font, start to type somewhere and now you can paste the icon itself (press ctrl+v) and the icon will appear.

### How to install a font on Linux

1. The font file:  ../desktop-font/glyphicons-basic-regular.otf
2. Double click on the font file and click "Install Font"
3. Open your favourite graphic app, choose the correct GLYPHICONS font, start to type and paste - press ctrl+v and it should works.

If this doesn't work in your Linux distribution, you have to create a new directory in `../usr/share/fonts/opentype`, copy OTF font file in this folder and update the font cache.

Note: if you see in the preview of the font just a series of question marks or empty squares - it's ok, this is caused by the fact that each system uses its own example text and this text is usually made from regular letters (remember "PUA"?). On the next page, we'll try to copy & paste some icons from the website.

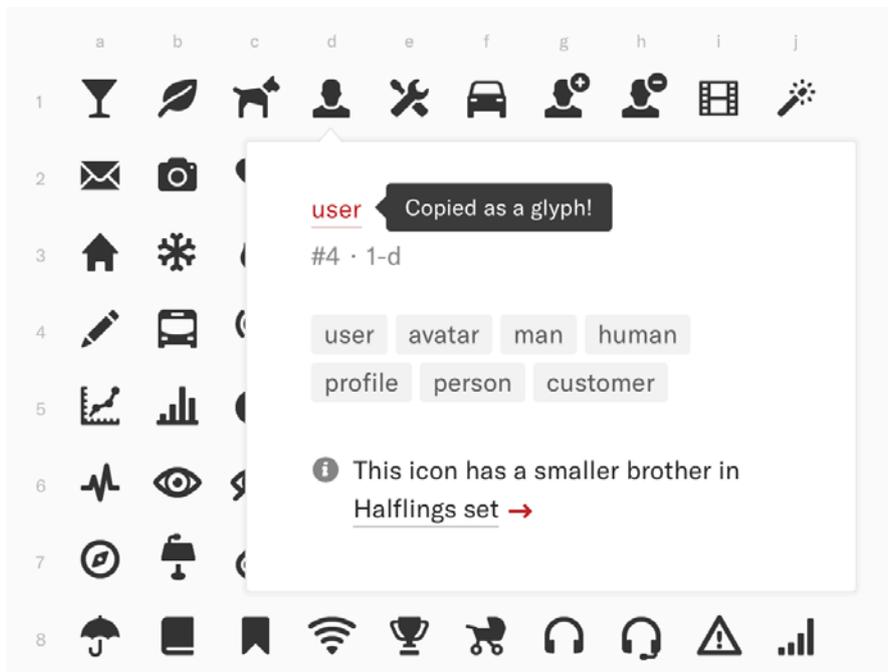
## How to use font in your desktop app

There are more than following two possibilities, but these are the most straightforward methods. If you already have a correct font installed (from previous page), you can continue.

### Copy & paste icons right from the website

This method is suitable for almost anyone and all you need is to display the page with any set on [www.glyphicons.com](http://www.glyphicons.com) and have the corresponding font already installed in your system.

1. Find the icon you want to use and click on its name, that's it. The glyph, font character, is now copied in your clipboard.



2. Now start to type with a text tool in your desktop app (with a correct GLYPHICONS font already selected), put the cursor where you want the icon to appear and paste copied glyph from you clipboard.

That's all, now you can work with this icon as you'd normally do with any other text in your app. If your app allows you to display all characters (glyphs) from your fonts, you may also try the second method of adding icons described on the next page.

In the previous version of [glyphicons.com](http://glyphicons.com) you had to manually highlight and copy icon itself, as icons were there as an icon web font – today, it's just a click on the name of the icon.

#### New

As you may noticed, there are two other values, right under the name of the icon.

The first value indicates the order number of each icon (so it's easy to search for any icon just by its number).

Second, alpha-numeric value, contain coordinates of each icon in table, as it is also saved in source Ai files.

## Enter icons via Glyphs panel (Adobe InDesign)

Following example work similarly in other apps, which are able to display individual Glyphs from the font. I tried this in recent versions of Adobe Illustrator and Adobe Photoshop and it works without problems. It's an alternative way of embedding icons, when you do not have access to the internet or don't want copy icons directly from the [www.glyphicons.com](http://www.glyphicons.com).

1. Open the Glyphs panel by choosing:  
Type > Glyphs or Window > Type & Tables > Glyphs

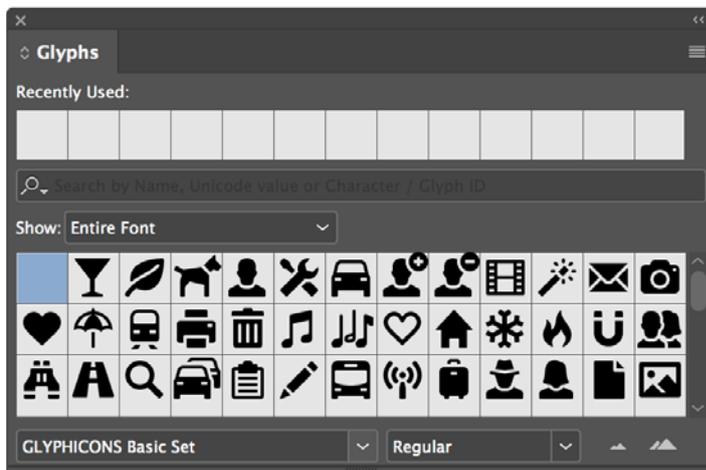


2. Select the GLYPHICONS font (set) you want to use by choosing one from the selection in the lower left corner.



If you now can't see GLYPHICONS font in the list, it isn't probably installed in your system and you should repeat the steps on the previous page.

3. Your Glyph panel should now look like on the screenshot below, (depending on which GLYPHICONS font did you choose).



InDesign will keep last used glyphs you inserted above, so if you work longer with similar icons, it saves you a lot of time with searching.

4. Now use the type tool, click the place where you want to display the icon, get back to Glyphs panel and just double-click the icon you want to insert to the text and the icon should appear.

When working with an icon font in InDesign, it's especially useful to create a custom "Character Styles", so you can later control all icons in your document, without having edited each separately.

I usually change the color of the icon, its size and adjust the baseline shift, because when you want to use icons as a bullet image for a list, it's usually necessary to set custom position to your paragraphs.

## Pros and cons of using icon fonts in desktop apps

Every coin has two sides, but from my perspective the advantages far outweigh possible disadvantages.

- ✓ You can change the appearance of the icon exactly as you normally change the appearance of any other text font, including changing its size, color, baseline-shift, rotating, etc.
- ✓ When the new version of the font is available, you just replace one font for another and all icons are automatically updated :)
- ✓ If you keep your cursor right behind the icon and hit spacebar, you'll get its ideal white space, because the size of the space is in all GLYPHICONS fonts already defined as an exact half of the icons body.
- ✗ You should keep in mind, that it's still a font and therefore it may be necessary not to forget to outline the text when exporting to PDF, because others without having this font installed would see just empty squares.
- ✗ Icons will remain in a fact a text, so naturally it carries everything negative, such as the impossibility of easily relocating outside the text frame, combining them together, etc.

Do you know any other pros or cons? Let me know and I'll add it here for others.

If you know about any other pros and cons, let me know at [glyphicons@gmail.com](mailto:glyphicons@gmail.com) and I'll update this list accordingly.

# GLYPHICONS on the web

There are plenty of ways how to implement icons on websites and which one to choose always depends on personal preferences and also it may vary project from project.

Either way, some methods are objectively better and considered as more modern than others.

## Where are icon fonts for the web

You were used to have all icons as an icon font for the web in the last few versions of GLYPHICONS, but it turned out to be a bit irresponsible and short-sighted, from me as an author. Why?

The first problem appeared as the total number of icons in one font file exceeded a certain amount (especially in Basic set as there were 800 icons and it's 900+ now) and the file size of the web font started to be ridiculously (unnecessarily is perhaps a better word) large. That got worse with each update. Other major problem was web icon font rendering issues, hinting and other things over which you have virtually no control. Last but not least, I have to mention accessibility problems, when fonts didn't load properly or at all, you'd see just empty squares, etc.

Overall, even though it looks like icon fonts still have some advantages, such as fewer HTTP requests and speed when using a large amount of icons on one site, now we have a better way how to include icons on websites, also in vectors, but more semantic, accessible and user-friendly – yes, it's SVG.

As I said, I'm not a programmer or HTML coder, so if you're eager for some arguments, I am going to refer you to the article that made me switch: [Inline SVG vs Icon Fonts](#) by Chris Coyier.

Of course, some people will still want to use icon fonts, and the choice is yours. But if you're going to create and use icon fonts on the web from GLYPHICONS, please read the next page.

More info about version 2.0 and all related changes in detail on the blog: <https://www.glyphicons.com/blog/introducing-version-2-0/>

If I didn't discourage you from using icon fonts on the previous page and you still want (or you are forced by the client) to create an web icon font for your web project, I'm not going to stop you, I just can't no longer recommend it as a default embedding method.

Today, there are web based tools (some of them free) which will help you to generate icon font for you.

As a source, it's the best to use individual SVG files, which you may find in  `../svg/individual-svg/`

But before you start, I'd like to ask you politely to keep an eye on a few things when generating the font:

- ➔ please, don't mix GLYPHICONS with other icons in one file
- ➔ please, keep the word "glyphicons" in your file name, e.g.
  -  `glyphicons-custom.eot`
  -  `glyphicons-custom.svg`
  -  `glyphicons-custom.ttf`
  -  `glyphicons-custom.woff`
  -  `glyphicons-custom.woff2`
- etc.
- ➔ please, do not use more than 150 icons in one file/project

That's all, I did not include any of this in my license text, because firstly, I wanted to keep it as simple as possible and secondly, I presume that my customers are clever and considerate people, who don't need to be strictly limited in this by some license.

For the most people, it's probably more convenient to use some icon font web generator such as IcoMoon App ([www.icomoon.io/app/](http://www.icomoon.io/app/)). It's nice that this app has also an offline version as an extension in the [Chrome Web Store](#).

That's all, if you'd expected some tutorial on how to create icon fonts, I have to disappoint you.

If you still want to (or have to) use an icon font on the web, you can create these files by yourself, the choice is yours - but there are now SVGs instead as a default and recommended embedding method. I'll try to describe all basics on the next few pages.

## SVG it is

SVG stands for Scalable Vector Graphics and SVG graphic is defined in XML, which means you can edit/create them using a graphic software or a text editor, pretty cool, isn't?

Today, SVG format has a wide support across all major browsers and if you don't plan to create a site which have to work on IE8 or very old Android, you're good to go!

If you need support for some older browser or OS, just use PNG files as a fallback, for example with Modernizr.

In theory, SVG can contain not only vector paths, but also raster images and texts, but our SVGs will contain only paths defining the shape of our icons and necessary markup.

This markup remains normally invisible and when we say SVG file, you probably imagine a picture, such as this grinning smiley below (from [GLYPHICONS Smileys set](#)). But inside the file, it looks like on the screenshot on the right.



```
glyphicons-smileys-1-grinning.svg — individua
1 <svg id="glyphicons-smileys" xmlns="http://www.w
  viewBox="0 0 32 32">
2   <path id="grinning"
  d="M18,13.5v-1a1.5,1.5,0,0,1,3,0v1a1.5,1.5,0,0,1
  0008,1.50008,0,0,0,14,13.5v-1a1.5,1.5,0,0,0-3,0v
  ,0,0,0,12.5,15ZM28,16A12,12,0,1,1,16,4,12.01343,
  ,16Zm-3,0a9,9,0,1,0-9,9A9.01016,9.01016,0,0,0,25
  2285A1.00048,1.00048,0,0,0,21.65137,17H10.34863a
  0-.94238,1.3335,6.99172,6.99172,0,0,0,13.1875,0A
  0,22.46777,17.42285Z"/>
3 </svg>
4
```

Line: 2:434 | Plain Text | Tab Size: 4

SVG uses CSS for styling and JavaScript for scripting. You can play with gradients, patterns, clipping, masking, filter effects or even animations while keeping all this in a relatively small size.

On the following page, we'll try to cover some basic methods how to include SVGs into the website .

## SVG embedding methods

Unexpectedly, there are many ways how to integrate SVGs into the website and our goal here is not cover all possibilities, mark one of them as the best or teach you how to write HTML and CSS. Our goal is to make an overview of the most common methods and pick the simplest and easiest to use methods, with some examples.

- ➔ Using an image tag (recommended method)
- ➔ Using CSS as a background image (recommended method)
- ➔ Directly in code as inline SVG (alternative method)
- ➔ Using an embed tag
- ➔ Using an object tag
- ➔ Using an iframe tag

Embedding SVG directly into the code is appropriate to use mostly when you need to quickly load an image, usually as one of the first things on the page (e.g. when page is loading and you want to display some kind of loading animation). It should save some time on HTTP request and reduce the time needed for page to load.

If you're going to use only very few icons on one page and don't mind the bigger HTML code, you can try to paste SVG code directly to your HTML.

`<embed>` and `<object>` tags intended for somehow interactive or external content. It's hard to tell when to use what. `<embed>` is an older HTML tag used in the past mostly for a Flash content. `<object>` tag is more modern and it lets you define a fallback content, so it seems like a better choice. `<iframe>` it's very similar to `<object>` tag and it's usually used when you need to embed a whole HTML page or content.

Possibly, there are other methods, but here we'll use only the first two methods as a basic demonstration, because these methods are the most straightforward and easy to use for most of us.

I presume that you already know the basics of HTML and CSS, if not, it's better to learn at least the minimum first.

### Simple method using an image tag

Using the HTML `<image>` tag, you can embed and handle SVG images the same way you do raster image formats. This is also safer than using other methods, because when we use this tag, some browsers disable scripts.

If you need to run any scripts inside the SVG, it's probably better to use `<object>` tag instead.

Let's try it together, step by step. First, we need to find an icon you want to embed.

You may search directly in files on disk for icons file name or go on [www.glyphicons.com](http://www.glyphicons.com), choose one of the icon sets and use search input in the right top corner. This is a better method if you aren't sure what you're looking for, as most of the icons have search tags (which I'm constantly improving and updating), so you'll more likely find an icon you need. You may also use the selection in top left corner and filter only icons from a certain group.

Anyway, once you find the SVG file which you want to embed (for now, we use again our grinning smiley (from [GLYPHICONS Smileys set](#)), it's time to add it in your code like this:

```
1 
2
3
4
5 
7
8
9
10 
```

File names of icons are a bit longer, but please keep them as they are, if it's possible. It will help you to stay organized, you can easily replace files after updates and you help me to fight with various copycats.

On line #1, it's just a minimum, but to make it valid, you would also have to add at least empty alt attribute. You can leave attribute alt empty, but only when the icon have purely decorative function or the meaning of the icon is unclear or it is not possible to express its meaning with a text.

However, there is usually no reason to leave alt attribute empty, in fact the best practice is to think about the context of the icon - what does this icon represent on your website and what's around it. For example, if you use the icon "wrench" for a link to settings, it's better to use the word "settings" than "wrench", which wouldn't be much helpful for users or screen readers assisting blind and visually impaired users. This important attribute is on line #6.

This is actually quite complex issue and if you're eager for more information, visit: [webaim.org/techniques/alttext/](http://webaim.org/techniques/alttext/)

Now, if you don't specify the width and height attribute (as you don't have to), the browser will use the original size of the SVG file. It's obviously better to define the size of the icon in CSS, so the size definition in HTML on line #11 is a demonstration of that.

### Simple method using CSS background image

Again, you may use SVGs with CSS exactly as you'd do with PNGs or any other raster formats such as JPG or GIF.

```
1  .smiley {
2  background-image: url(glyphicons-smileys-1-grinning.svg);
3      }
4
5
6  .smiley {
7  width: 32px;
8  height: 32px;
9  background-image: url(glyphicons-smileys-1-grinning.svg);
10     }
11
12
13 .smiley {
14 width: 32px;
15 height: 32px;
16 background: url(glyphicons-smileys-1-grinning.svg)
17 center center no-repeat;
18     }
```

Once you define the URL of the icon itself (line #1), it's wise to define also the size of the icon (but this value is mostly already inherited) on lines #7 and #8.

In reality, all properties are usually written in a single declaration as shorthand which starts on line #16 and #17.

It may not be necessary, but in case you have to provide fallback with PNGs, use <https://modernizr.com>

## SVGs and colors

Again, there are plenty of ways how to colorize your SVG icons and there is no "best solution", it all depends on personal preferences and your workflow. Here are a few options:

- ➔ Edit SVGs in a vector editor (⚠ not-a recommended method)
- ➔ Edit the color with CSS (for advanced users)
- ➔ Edit the color SVGs manually (the recommended methods)
  - ➔ With a text editor (keeps SVGs clean and lightweight)
  - ➔ With SVG tool (easy + keeps SVGs clean and lightweight)

### Edit the color of SVGs in a vector editor

This is probably the first thing you can think of: it's an image so I use my Illustrator, Sketch, Inkscape, Corel... change the color of the icon and hit save, right? Though there is a big chance that somewhere during the process of re-saving or re-exporting files, you add a lot of unnecessary markup, make files bigger or even disrupt the shapes themselves. This is objectively the worst method you can use, unless you have a complete control over the output.

### Edit the color of SVGs with CSS

It would be great to control the appearance of SVGs simply by using CSS, but apparently they can be styled with CSS only when the SVG is inlined directly in your HTML, as an object, iframe or embed tag. Then it's possible to use CSS "fill:" or "filter:" and change the color of the paths inside the SVG with various CSS properties.

But there is one specific scenario, where is better to inline SVG above any other methods and that's when you're using the SVG icon as a loading or a placeholder symbol. Why? Mainly because:

- ➔ SVG is a part of the DOM and those loaded very quickly.
- ➔ It saves HTTP request and the whole page will load faster.

But... there are some drawbacks: if you use more than very few icons, your HTML code will become bulky and when you decide to change any icon, you'd have to do it across your whole website.

Here is a nice article about how CSS fill property works: <https://css-tricks.com/almanac/properties/f/fill/>

## Edit the color of SVGs manually

You can edit SVGs with a simple text editor (for me it will be always a great Textmate) or now even more easily, with the new SVG tool.

### → With a text editor

Do you remember the screenshot from page #16? Here it's again. As you can see, by default, there is no color defined.



```
glyphicons-smileys-1-grinning.svg — individua
1 <svg id="glyphicons-smileys" xmlns="http://www.w
  viewBox="0 0 32 32">
2   <path id="grinning"
  d="M18,13.5v-1a1.5,1.5,0,0,1,3,0v1a1.5,1.5,0,0,1
  0008,1.50008,0,0,0,14,13.5v-1a1.5,1.5,0,0,0-3,0v
  ,0,0,0,12.5,15ZM28,16A12,12,0,1,1,16,4,12.01343,
  ,16Zm-3,0a9,9,0,1,0-9,9A9.01016,9.01016,0,0,0,25
  2285A1.00048,1.00048,0,0,0,21.65137,17H10.34863a
  0-.94238,1.3335,6.99172,6.99172,0,0,0,13.1875,0A
  0,22.46777,17.42285Z"/>
3 </svg>
4
```

Click right behind the end of the line #2, in the place where the closing of the path is and add the "fill" property with the color in hex format (fill="#ff0000"). Now just save the file.



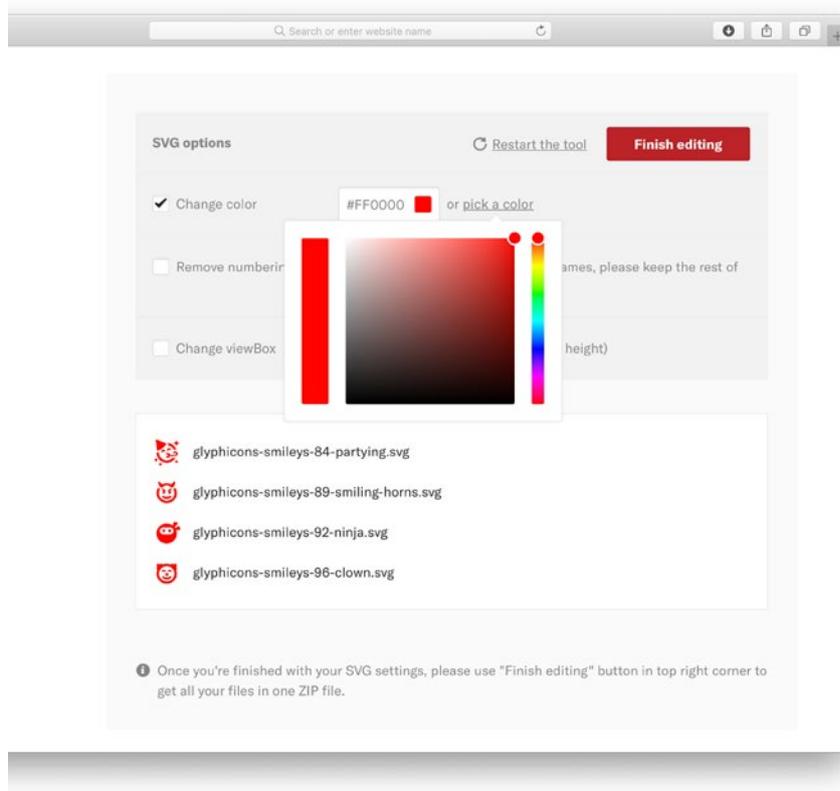
```
glyphicons-smileys-1-grinning.svg — individua
1 <svg id="glyphicons-smileys" xmlns="http://www.w
  viewBox="0 0 32 32">
2   <path id="grinning"
  d="M18,13.5v-1a1.5,1.5,0,0,1,3,0v1a1.5,1.5,0,0,1
  0008,1.50008,0,0,0,14,13.5v-1a1.5,1.5,0,0,0-3,0v
  ,0,0,0,12.5,15ZM28,16A12,12,0,1,1,16,4,12.01343,
  ,16Zm-3,0a9,9,0,1,0-9,9A9.01016,9.01016,0,0,0,25
  2285A1.00048,1.00048,0,0,0,21.65137,17H10.34863a
  0-.94238,1.3335,6.99172,6.99172,0,0,0,13.1875,0A
  0,22.46777,17.42285Z" fill="#ff0000"/>
3 </svg>
4
```

To get the color code in hex format (if you don't use any graphic software), just try any color picker available online, such is this from w3schools: [https://www.w3schools.com/colors/colors\\_picker.asp?color=ff0000](https://www.w3schools.com/colors/colors_picker.asp?color=ff0000)

➔ **Use free SVG tool on glyphicons.com**

The [SVG tool](#) is a simple web tool which runs in almost any common desktop browser and it lets you upload up to 25 individual SVG files, edit some of their basic attributes (without changing anything else in their optimized code) and download all icons back to your computer in one ZIP file.

Simply upload your selection of icons, change their appearance and click on "Finish editing". Now you can download all edited SVG files back in one ZIP file, it will take just a second.



New

SVG tool is available for free:  
<https://www.glyphicons.com/tools/svg/>

The [SVG tool](#) is not a feature-rich app or a heal-all solution for SVGs on the web and also definitely not a replacement of your favorite desktop vector editor, but it can be a great help anytime you need to quickly get ready multiple GLYPHICONS SVGs for your new web project, without having to worry about losing their optimization and small file size.

If you want to know more about how it works, there's [a blog post about SVG tool's first version](#).

### **That's all for basic SVG editing**

These are just very austere examples, but now it should be clearer how to handle SVG files at a basic level, including how to embed them in website and change their color.

You should never copy this procedure of embedding SVGs blindly, as these examples are intended just for demonstration purposes. If you don't have any previous experience with HTML and CSS or if struggle to make it work at all, I suggest you to rather hire somebody more experienced or invest some time and learn how to write a code yourself.

## Troubleshooting and tips

Here are few more tips which you may find useful, links to advanced techniques, common problems and also some alternative ways not described on previous pages. If you happen to be an expert in this area and find something which should be added or improved, please let me know and I'll update this page accordingly!

### **An in-depth SVG tutorial**

<https://flaviocopes.com/svg/>

### **Fallbacks for SVGs**

If you for any reason need to have fallback for user who runs older systems or browsers, I can recommend Modernizr: <https://modernizr.com>, which can help you not only with PNG fallbacks. If you want to learn more about fallbacks in general, here's a great article: <https://css-tricks.com/a-complete-guide-to-svg-fallbacks/>

### **SVGs don't show up on your own server at all**

If SVG icons don't show up, your web server isn't possibly set up correctly, more about it and simple solution with .htaccess file here: [https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Getting\\_Started#A\\_Word\\_on\\_Webservers](https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Getting_Started#A_Word_on_Webservers)

### **Do you want to be sure if it's ok to use something?**

Are you in doubt about any tag, file format, etc.? Is it ok to use this or that for some browsers or operating systems? Just open <https://caniuse.com/#search=svg> and type what's bothering you.

### **Try this for a larger number of the same icons on one site**

<https://developer.mozilla.org/en-US/docs/Web/SVG/Element/use>

### **A great article about everything what's in SVGs header**

<https://medium.com/@pnowelldesign/stuff-at-the-top-of-an-svg-f3ad198eb54e>

# FAQ

The most asked questions about GLYPHICONS. If you have questions about anything else or if you're looking for information about your order, license, etc., you may try to find an answer in [FAQ on the website](#).

## **Where are HTML/CSS files from previous versions?**

I created these files few versions ago (in version 1.7 and they were part of the PRO version till version 1.9.2). I was continuously updating, improving and maintaining them. The truth is, that some people were using them as a framework (not as an example) and in short: everything came to a point where it wasn't possible to manage these files and related support in any reasonable way. I've decided to shift my focus back to icons and invest more time in them, as I did in earlier versions.

## **Are SVGs already optimized?**

Yes, all individual SVG files in GLYPHICONS sets are already optimized in a sense, that their code doesn't contain any futile content and they are not unnecessarily large. They are rather balanced to keep the ability to work with them dynamically if needed and have the paths as unchanged as possible.

If you want to try optimizing them even more and reduce their size (which is really not necessary), you can use free **SVGO optimizer**: <https://github.com/svg/svgo>. Just be cautious, because you may lost their original shape and some fine details.

## **Can you take a look on my code?**

No, I'm really sorry, but I'm too busy and I barely have a time for other things around me. Also, I'm not an expert in HTML/CSS, I'm just a designer selling my drawings and certainly not a somebody who should give advices on programming ;) Just to be clear, I have to always consult everything related to programming in the Handbook and I always hire somebody else to write a code, as most of this is above my head. The intention behind all this is to give you some basic idea how things works.

If you want to stay in touch and get all the news, please  
[follow @glyphicons on Twitter](#)

Do you have any questions or want to say hello?  
[Send me an email at: glyphicons@gmail.com](mailto:glyphicons@gmail.com)

**GLYPHICONS®**

© 2010 - 2019 GLYPHICONS® All rights reserved.  
GLYPHICONS® is a registered trademark of GLYPHICONS.com s.r.o.