Archivability

Stanford University Press shares with you the belief that digital academic arguments deserve treatment in the scholarly record equal to that of traditional print media, and we have a serious commitment to ensuring the longevity of your work through responsible approaches to sustainability and archiving. Your conscientious authoring and design choices are the first step toward sustaining the long-term fidelity of your work. But because we acknowledge that even the most sustainably designed web interfaces are susceptible to normal decay, we must anticipate the necessity of preserving your project’s content so that readers and researchers can access it even after changing web environments begin to disrupt the project’s dynamic and interactive features. To that end, we intend to take multiple approaches to archiving your content and preserving documentation of its intended presentation.

Unlike a typical monograph publication, for which the publisher implements the material architecture that contains the content, you, the author, hold the creative responsibility for your project’s format, design, and execution. Nevertheless, as your publisher, we share the responsibility of the sustainability of your work and therefore recommend, encourage, and require certain technology and design standards. We have compiled these archivability guidelines, along with the enclosed style sheets to help you make informed decisions on the types of technological considerations that will impact your project’s longevity, and its amenability to current web archiving standards and practices.

In addition to your use of these recommendations, it is important you complete the technical questionnaire so we may begin developing the best publishing plan for your project, including its most suitable archiving environment.

Archivability Spectrum

A digital project falls into a spectrum whose endpoints are usefully analogized as object and performance. Whereas a book can be considered an object, with its performative qualities limited to the audience’s experience while reading the book, an interactive digital project shares certain qualities of ephemerality that characterize a performance. It only functions as an object insofar as the device displaying the project or facilitating the reader’s experience with the material remains intact. Because technology changes so rapidly, these devices and standards are always in flux. Thus, depending on the complexity of the tools and applications needed to deliver the project in a reader’s web browser, the project may be more or less susceptible to decay as technology evolves and renders certain tools and applications obsolete or as external data or media change locations. To help you determine where on the spectrum your project falls, and what further safeguards will most likely be required because of its location on the spectrum, we have divided the possible archivability states into three broad and non-exclusive categories: 1) archive ready, 2) archive amenable, and 3) archive resistant.
Type 1: Archive Ready
This type of project is extremely rare. Even though it functions as a performance, it is more analogous to an object than most digital projects. The look and feel, structure, and functionality can be preserved and should not be susceptible to decay as technologies evolve, or they can be reproduced in future environments with minimal resources and no loss of fidelity. Such a project uses standard file-and-folder systems and includes a sitemap, xml or otherwise, that provides fixed urls for each page and state. Each state is a static HTML page, and all data and media presented on those pages are hosted with the project and not queried from third-party sources. The project does not link to external websites or content other than that stored along with the project itself. This kind of project does not use proprietary software or applications, which would require updating and/or patching. There are few or no dynamic scripts that will glitch or fail when the coding standards or libraries change. In addition to satisfying all the style recommendations in the accompanying documents, it adheres to web archivability and accessible standards as published by the World Wide Web Consortium, Library of Congress, Stanford University Libraries, and the Web Accessibility Initiative:

- World Wide Web Consortium List of Standards: https://www.w3.org/TR/
- Stanford’s Archivability Standards: http://library.stanford.edu/projects/web-archiving/archivability
- Web Content Accessibility Guidelines: https://www.w3.org/standards/webdesign/accessibility

Additional Requirements:
Because this kind of project is made up of pages that can be automatically crawled and recorded as well as a file system containing all HTML pages and media objects, with no reliance on database protocols or third-party queries, few, if any, additional materials are necessary to aid in the archiving process. Nevertheless, as with all projects Stanford University Press publishes, documentation of the composition process as well as description (written as well as video-recorded and/or screen-captured) of its functionality and featured contents is encouraged. (See enclosed “Documentation” guide.)

Type 2: Archive Amenable
While the original functionality of this type of project will last longer than one that is archive resistant, certain features can be expected to break sooner than those of an archive-ready project. An archive-amenable project can still be archived, but the preparation stages will take more time and could require the author to provide alternative formats or paratext. The maintenance and preservation of the project’s archive also requires more work and frequent attention. Additionally, the archived version will likely act and/or look differently than the initially released publication. This type of project might unnecessarily employ a database structure (Wordpress, Drupal, Scalar, etc.) that could impede the efficiency of navigating the site as well as put undue pressure on the server, slowing down the project’s interactive elements. Consideration should be given to whether the user experience requires the framework offered by the database, or if that database functioned primarily for the authoring process and is not needed to deliver the published content. Oftentimes, for example, a site built in Wordpress does not actually need the scaffolding of that platform on the reader’s end. In such cases it might be advantageous to scrape the content and restructure it using HTML(5) and file-and-folder systems, an architecture which puts far less strain on a server and requires fewer updates to security protocols. Other factors that characterize an archive-amenable, rather than archive-ready, project are javascript
that will require updates, web fonts that need to be downloaded each time a page loads, external links to websites that might disappear or change location, etc.

Additional Requirements:
When we perform the technical review of your project, we will compile a formal list of further requirements to aid in the pre-publication archiving process. Although it’s difficult to anticipate the form certain archive-amenable projects will take, we can at the very least suggest that you be prepared to either edit platform and code base or to build in the time it could take for SUP to perform such actions in the production phase. You should also strive to collect cached datasets from any third-party sources your project queries and gather and save all media elements that are currently hosted separately from your project (e.g. Vimeo, Soundcloud, Imgur objects, etc.)

Type 3: Archive Resistant
Because most authors utilize dynamic authoring tools and cutting-edge technologies, a majority of projects fall under this category. Like a performance, it is possible to record an instance of this kind of project, but preserving the original experience beyond five years is a significant challenge with current archiving technologies. This type of project is likely designed using custom code that might not be accessible to someone who may be required to update it in the future for it to work properly in evolving web environments. While the technologies needed to render the project are current now, they will certainly change in the near future. The best approach to preserving such projects is emulation, a still inconsistent and imperfect system that requires stacks of multiple technologies such as specific browser versions, operating systems, and code libraries, all of which must work in conjunction with any third-party data that is likely to become unavailable or incompatible with the old environments being emulated. Until a stable, reliable solution is implemented that can recall today’s technologies in the future, we must at the very least safeguard against the loss of content by creating an archived version of the project that necessarily omits the more experimental and less stable features. Such features include script-based animations and visualizations, network and database visualizations, and embedded APIs that query third-party sources. Also problematic are links to external websites that will very likely become broken as content moves or is renamed and as security protocols update the urls used to locate that web content. While we don’t restrict your use of these tools and formats, we cannot recommend them, especially when they are the only method of delivering content, and we cannot guarantee their long-term (or even short-term, in some cases) fidelity beyond the initial release of your work. Likewise, if you choose to employ such technologies, you acknowledge their ephemerality and understand that though your content will be archived, the fidelity of the initial user experience will likely be suspended until more efficient and sustainable emulation technology is developed. Although this type of project is dynamic and employs design features that may be creative, impressive, and engaging at the time of publication, these same features will most likely also render the site shorter-lived in its intended form, and even upon its initial release slow or inoperable on older machines.

Additional Requirements:
In order to convey to future audiences the project’s original functionality, it will need to be documented and supplemented with paratext, such as video presentation or screencast by the author or SUP that thoroughly describes the project’s purpose, scope, functionality, and key features. Such a project will also likely require any range of the following: a thorough written description of the project’s development and the principles that guided the technology choices, a visual blueprint of the project’s organization and structure, if feasible a complete static HTML version of the site that contains placeholders (e.g. jpg versions of interactive visualizations) for any moving parts that are expected to
degrade over time, and access information for a public or shared repository of any custom-built programs or applications.

Resources

Design:
For more information about sustainable web design, we recommend you read these more detailed resources:
- An article on the advantages of static websites over database-driven sites: https://www.smashingmagazine.com/2015/11/modern-static-website-generators-next-big-thing/

Delivery and Access:
To test your project’s code and accessibility compliance, use the following tools:
- Web Accessibility Evaluation Tool (WAVE): http://wave.webaim.org/
- W3C’s Markup Validation Service: http://validator.w3.org/
- W3C’s CSS Validation Service: http://validator.w3.org/

Web-Archiving:
For more information on web-archive format and emulation solutions, read the following:
- Bibliothèque nationale de France’s Description of WARC format: http://bibnum.bnf.fr/WARC/
- BWFLA’s Description of Emulation: http://bw-fla.uni-freiburg.de/

WARC in Action:
Use these tools to glean the functionality of your project as it would be rendered in WARC format. Note that a web-archived version of the project is not the same as its initial fully-operative published version as the crawlers that capture the pages do not have access to the scripts behind those pages. This is why conscientious design, and design that avoids third-party querying, is so important.
- Web Recorder: http://webrecorder.io
  (First, record the url and browse the site for a while. When finished, play it back and note what works and what doesn’t.)
- Wayback Machine: https://archive.org/web/
  (If your project currently has a public url and has been crawled by Internet Archive, see what it looks like as a crawled website.)
Documentation

Current standards in the digital humanities and social sciences, as well as recent reviews of our own projects, place strong emphasis on the importance of documentation in digital work. Recording the processes of development, the technologies employed in that process, the technical and system requirements on the user’s end, and the reader’s experience of interacting with the project ensures that the type of information needed for proper use and archiving remains available for all stages of the project’s development, publication, and preservation. We ask that you maintain detailed notes on all aspects of your project’s technical formats, dependencies, and updates and that you develop this information for inclusion in the project or its archive.

What to Document

Development

The processes and choices undertaken during the development of your project represent important information for digital humanities scholars and computational social scientists. By elucidating your development processes, you provide aspiring authors a valuable resource and contribute to the growth of digital scholarship. To that end, you may wish to include the following information in an “About” page or sub-page of the project:

- Native development environment, including operating system, interfaces, programs, code libraries, and languages used;
- Browsers and browser versions in which you tested or built the project;
- Software and software versions used;
- Datasets queried, consulted, or employed;
- Server specifications and licenses if hosted elsewhere before production or publication;
- Reasoning for technological choices;
- Reasoning for design decisions;
- Reasoning for file naming structures;
- Collaboration and consultation;
- Results of tests run

User Experience

Assuming all web-based work is to some extent ephemeral (see “Archivability” guide), it is important to mitigate any project’s decay with a variety of resources that convey and describe the intended functionality of the work. In addition to a list of technical and system requirements, this kind of documentation may include video of a user interacting with the project; narrated screencasts showing and describing the project’s content, features, and functionality; visual blueprints of the content structure or project architecture; spreadsheets of the content files and their relationships; and narrative description of the user experience. A reader’s map or instructions on suggested project navigation can also be very useful. You may choose to an extent whether to include any combination of this material within the project itself or as supplementary material to be stored in the repository with the project’s archive.
Where to Document

All projects should include an “About” section that is prominently linked from the landing or index page that lists, at the very least, the technical specifications for optimal performance of the project and an explanation of the rationale behind the project’s design and delivery in a digital, as opposed to print, medium. It might be helpful to consider this section part of the meta- or para-text of the project. This area might additionally include, if desired or appropriate, any of the assets mentioned above.

Any assets listed in this document that you choose not to include in the “About” section should be submitted to SUP as supplementary archival material so we may organize it as part of the permanent archive in Stanford Digital Repository.
Digital Labor and Data Sources

Labor
The highly collaborative nature of digital projects means that authors often employ the talents of a diverse and revolving team throughout the development of their work. Just as authors of books acknowledge the people whose input and assistance facilitated the research and writing process, authors of digital projects must credit those whose time and talents went into the development of their projects. The idea of digital labor in humanities and computational social sciences is a prominent topic of discussion and scholarship, and as Humanist scholars and publishers, we are especially aware of the value of the labor that drives new forms of scholarship.

To advocate for the collaborative spirit of digital work, we require that each project contain a section crediting the people who worked on the project, whether authors, programmers, student assistants, or otherwise. While credits may take unique forms, we ask that you include somewhere in your project a list of each contributor’s name and role. Ideally such a list will be clearly accessible and linked from the Home or About page. In addition to building the credit into your project, we ask that you submit a text-file version of this list so we may use it in our metadata for the project’s archive.

Data
Projects that use or display data should clearly indicate the source of that data. Any data visualizations should include or link to a citation of the source of data and the visualization method or API that actions the models. Just as you would cite a source of information, you should also cite the tools used to render that information. For example, if your project employs an interactive map, that map includes geodata, like latitude and longitude, from one or more services and sources. Likewise, if specific APIs or tools are embedded in the project to query and deliver data from third-party sources, those interfaces should be identified. As with the contributing parties in the creation of the project, the sources of the information behind the arguments are essential to a digital project’s rich ecology, and as authors and publishers of this work, we value all the contributions that drive it.
Interoperability

Readers will want to employ a variety of devices and browsers to view your project, so it is important to consider the operability of your project in multiple contexts. If you want your project to function on mobile devices, you’ll need to include meta tags for responsive display. If, on the other hand, you prefer your project to be viewed on a computer screen, you should enable a message on the landing or introduction page that the project is best viewed in the context you stipulate. Likewise, if you know your project works best in a particular browser or browser version, note those parameters up front as well.

BROWSERS

Whether you realize it or not, you’ve likely already identified the optimal environment for readers to experience your project. What operating system are you using to build your project? What browser do you use when you test how your project functions? Which version of that browser are you currently running? All these specs should be noted in your documentation of the project, but they should also appear as recommendations to your readers in an easily-identifiable location such as the landing page, a script that triggers a warning if the reader is using a different browser/version, or in an instructional section.

OPERATING SYSTEMS

Because your project is web-based, it is unlikely that a reader’s operating system will directly affect the look, feel, or functionality of the content. However, it is a good idea to include documentation of the operating system you used to design your project and to consider any style elements that might render differently on different machines. Fonts, for example, are not the same by default on Mac and Windows machines. (See the “Fonts” guide for more on this topic.) Likewise, Mac users will likely be using Safari rather than Internet Explorer while PC users are more likely to use IE than Safari. As of 2017, however, Google Chrome was the most used web browser around the globe. Be sure to test your project in multiple browsers and on multiple operating systems to ensure a consistent experience for all your readers.

MOBILE DEVICES

For a variety of reasons, you might prefer your readers to only access your project on a computer. If this is the case, include a message to this effect on the project’s landing or instruction page. However, it is worth considering that many readers, especially students, do much of their work on tablets and smartphones, which are easier to port around campus and to access on the fly. If you want your project responsive to multiple viewing contexts, and you are using HTML5 and/or stylesheets, make sure to include responsive meta tags.

Example

<meta name="viewport" content="width=device-width, initial-scale=1.0"
File Names and Formats

File Names
A file’s name has the potential to convey important information about its contents. Establishing a clear naming system for project files should be a priority in early development stages, but may also need to be implemented later in the process in order to make production, publication, and archiving proceed smoothly. Ideally, file names should logically reflect the subject matter contained in the file. This makes archiving easier, and it can also aid in managing the files during review and editing stages. The key in any file naming system is consistency. While many techniques are acceptable, it is crucial to maintain a consistent method. The following are additional common and basic recommendations for file naming:

- Do not use spaces. If the file name consists of multiple words, use hyphens or camelCase.
- Do not use arbitrary numbers as filenames. Unless there is an obvious reason to use numbers instead of words for filenames, only use numbers when appending to words in order to distinguish states or versions.
- Do not use punctuation other than hyphens. Many special characters and punctuation marks trigger actions in the back end of a site that could cause errors.

File Formats
As in the case of file naming, it is important to use consistent file formats throughout your project. For images you might need to employ a variety of formats for quality and functionality (see “Images” guide). However, for other media, including video and audio, it is preferred that you use the following formats whenever possible. Depending on how your project operates, you might need to use other file formats defined by an outside API or platform, and these recommendations might need to be reconsidered in those cases. If your preferred format is not listed below, please contact us so we can assess the needs and requirements of the project’s media storage and delivery.

Images
- *jpg/jpeg*: This format works for most images.
- *png*: Use this format if the image contains transparent layers
- *tiff*: Only use this format for images that need extremely high quality for zooming purposes. In a digital environment, the need for this will be rare.

Video
- *mp4*: Use this format if you will be employing an HTML5 player to deliver video locally. This method is preferred for projects that do not include a lot of video.
• **mov**: This format is preferred for delivery through our digital repository. If your project contains a lot of video, we may determine that this is the best delivery method, in which case you will need to provide all video for the project as soon as possible so we may deposit the files in the repository and provide you with their persistent urls.

Audio

• **mp3**: Use this format if you employ the HTML5 audio player in your project.
• **wav**: This is the preferred format of our repository’s player. If you have a lot of sound files, we may determine that storing them in and delivering them through our digital repository is the best method. In this case, you will need to provide all audio for the project as soon as possible so we may deposit the files in the repository and provide you with their persistent urls.

Text

If you are using a platform such as Scalar, your text files will be automatically rendered within the system. For all custom built sites, text-based files should be html(5) or txt. Do not use PDFs.

Resources

• W3C’s overview of video formats for HTML5 players: [https://www.w3schools.com/html/html5_video.asp](https://www.w3schools.com/html/html5_video.asp)
• W3C’s overview of audio formats for HTML5 players: [https://www.w3schools.com/html/html5_audio.asp](https://www.w3schools.com/html/html5_audio.asp)
Links

Links form the basis of navigation in digital projects, so it is crucial to properly deploy them. To ensure a smooth transfer between development and publication server environments and to avoid link breakage, we recommend the following protocols.

Internal Links

Use relative links rather than absolute urls when linking to media or content within the project. This practice will ensure that when the project moves from its development environment to our publication server, the links will still point to the correct files and won’t break as they try to find content at an old location.

For example, use `<a href="file_name.ext">` rather than `<a href="http://domain.org/file-name.ext">`. The same concept applies to style sheets and media src tags, including images and videos.

External Links

Ideally, a project will not link unnecessarily or arbitrarily to outside content or websites. Because we cannot control the behavior of outside material, we cannot ensure the links will continue to point to active content or the content that was originally stored at the linked url at the time of development or initial publication. When links to outside content are necessary, it is best to limit the number of links to that content. For instance, links may be included in a bibliography or reference page but not within the content pages that reference those sources. Instead, references within content pages may link to the bibliography, where a reader can then access the external link embedded in a full citation to the source. It is important that a bibliography include full citations along with links so once a link does inevitably fail, a reader can still discern the reference source and potentially find the linked content at its new location.

Link Targets

While it is up to each author to determine the most appropriate target function for links, we do recommend consistency and deliberate structure regarding how links open. Whether a link opens in a new tab/window or replaces the current content of a tab/window carries certain rhetorical implications. It is important to consider whether you want links to generate new and multiple reading portals or to take the reader along a single path that they determine through their interaction with the content.

Resources

World Wide Web Consortium recommendations on links:
https://www.w3schools.com/html/html_links.asp
Fonts

To ensure readers are able to experience your intended design features, it is important to choose fonts that are readable by a variety of machines, operating systems, and web browsers. Web safe fonts are those that, because they are already installed on a computer or device’s operating system, are rendered on screen by the device without any necessary file download. Web fonts are those that are contained within the site files or linked to in the CSS. These fonts must be downloaded by the browser before they can be displayed. Although this download occurs automatically, it can slow down the site’s performance as it must query the server or an outside source for the font files before the text can be displayed properly.

WEB SAFE FONTS

The following web safe fonts and font families are common on multiple operating systems and thus the most likely to be read and rendered consistently across multiple environments:

Serif: Times, Georgia, Palatino
Sans-serif: Arial, Arial Black, Verdana, Trebuchet MS, Tahoma
Monospaced: Courier New

WEB FONTS

When using web fonts, it is preferable to use open access fonts, such as those available from Google Fonts. Google Fonts provide templates for embedding fonts into a page’s or site’s CSS and host the font files on its own servers. As in all design choices, you should consider the sustainability of a font. Although Google Fonts are widely supported and accessible now, you should consider whether or not they will still exist on Google or third-party servers in the distant future. Even if they do, the more information a project must pull from external sources and third-party servers, the more likely the project will not look or function in the future like it did in its original state.

RECOMMENDATIONS

In all cases, but especially when the preferred font is a web font, it is recommended you establish fallbacks so if the first choice is not available on the server or the operating system, the font will default to a second or third choice that is available. Any CSS should include at the very least the desired font (e.g. “Times New Roman,” “Arial Narrow,” etc.) or font family (e.g. Times, Arial, etc.) as well as generic family (e.g. serif, sans-serif, etc.) values. When rendering a font on screen, the browser will display the first choice as long as it is accessible on the reader’s device (if a web safe font) or the server (if a web font). If the first choice is unavailable, the browser will instead render the second choice. If both fail, the
third choice, ideally a general family like serif or sans-serif, will be rendered using the browser’s default font selections.

EXAMPLES
CSS for paragraphs using “Times New Roman” web safe font as first choice, Times family as second choice, and serif as failsafe third choice:

(in <style>)

p {
    font-family: “Times New Roman”, Times, serif;
    font-size: 1em;
}

CSS for paragraphs using “Arsenal” web font from Google:

(in <head>)

<link href=“https://fonts.googleapis.com/css?family=Arsenal” rel=“stylesheet” type=“text/css”>

(in <style>)

p {
    font-family: “Arsenal”, sans-serif;
    font-size: 1em;
}

ANTI-ALIAS
While anti-aliasing or font-smoothing scripts can appear to make fonts more legible, on some displays including the ever more common high resolution monitors, these scripts have a negative effect on the readability of a font. The best approach to determining if a font is legible at different resolutions and specs is to view the project on multiple displays. Usually, you can avoid issues with fuzziness or pixilation in a font by employing a standard font like those listed above and ensuring the size is 10 pixels or greater. If a reader wishes to see more text at once in a display, they may choose to zoom out in their browsers.

RESOURCES
For more information on fonts and lists of common fonts and font families, see the following resources:

- Dan’s Tools CSS Fontstacks: http://cssfontstack.com (list of web safe fonts; CSS generators for font stacks)
• W3Schools’ CSS Web Safe Font Combinations: 
  http://www.w3schools.com/cssref/css_websafe_fonts.asp (lists and examples of commonly used web safe font combinations)
• Google Fonts: https://fonts.google.com/ (list and examples of web fonts; CSS generators for all available fonts)
Images

In addition to meeting permissions policies, all images should adhere to certain format quality, accessibility, and archivability requirements to safeguard against breakage, unreadability, or deterioration over time.

FORMAT QUALITY

Recommendations on image size, file size, and resolution differ depending on the purpose of the images you use. In cases where an argument is dependent on an image or when a project’s purpose is to provide access to digitized visual resources, higher resolutions and file sizes are necessary. In these cases, it is assumed you and your team know how best to handle the format and quality of images. However, when an image is an accessory or supplement to an argument, we encourage the following considerations.

Images on the web will display according to the resolution specs of a reader’s screen. In most cases, since web projects are not print objects, when creating your own images the resolution is not as important as the image size and the file size. Larger file sizes take longer to load and, when resized to fit a browser window, often do not display any differently from a file that takes up less space on the server. Unless an image is meant to be zoomed to display minute detail, keep resolution to a reasonable rate and pixel dimensions to actual screen proportion and capability. 72 ppi is usually fine for average monitors and keeps the file size small, while 130 ppi will display better on large screens but will take up more server space and slow the load time of the page.

If you are using images from other sources or photos you’ve captured with a camera, do not scale them up from their original dimensions. Doing so usually causes pixilation. If you need an image to display large on screen, make sure its quality is not compromised by scaling.

Images should be formatted as JPEG files. If background transparency is needed, use PNG. If close zoom is a mandatory feature of your project, you might need to use TIFF files. (Also see “File Names and Formats” guide.)

ACCESSIBILITY

All images should include alt tags so readers using screenreading technology will be able to ascertain the contents and purpose of each image. Insert appropriate alt text in each image via HTML or your platform’s media editor.

If design-appropriate, images should contain descriptive captions in a format consistent to that of your chosen citation style. The caption should include a citation of its source or point to the related listing if the source appears in the bibliography page or list of references.
ARCHIVABILITY

Like all media files, images should be hosted on the same server as the project, not linked to outside sources like Imgur or Flickr. Pages take longer to load when the browser must retrieve files from multiple sources. Having all the files in one place that are necessary to display a page ensures not only that the page loads more quickly and smoothly but also that the links to those files remain intact. In the event that a media object must be linked to an outside source rather than contained in the site directory, the files should be provided to SUP so we may deposit them in our repository and/or load them onto our own media servers.
Color

When choosing colors for custom images, icons, text, links, backgrounds, and other design elements, keep in mind the readability of your palette. Although certain design choices will necessarily reflect and embody your argument, you should consider your readers’ possible physical or computational limitations. The following recommendations will help you determine if the colors you are using may be inaccessible to readers or their displays.

COLOR SPACE

To optimize color representation for screen displays, use RGB or greyscale for color profiles when creating original images. In CSS, you may use color names, (R,G,B), or hex values to assign font, border, and background colors. Avoid CMYK profiles as those are intended for print rather than screen display.

ACCESSIBILITY

Readers with color vision deficiencies will not be able to perceive certain colors as contrasts to each other, so avoid relying on color in images and graphics to indicate contrast. An alternative when graphical contrast is necessary is to use greyscale or scaling grades of a single color that are discernable when rendered in greyscale by a browser or operating system’s display setting. If color contrast is necessary, provide readable textual or otherwise discernable graphic indication (e.g. shape, size, etc.) of the values represented in addition to the color-based representation.

RESOURCES

Code Beautify’s Color Values: http://codebeautify.org/rgb-to-hex-converter (lists accepted HTML color names; shows RGB and hex values for any selected color in a palette)