IATI Sphinx Theme Documentation

IATI Secretariat

Oct 02, 2026

Contents

1 Videos 1

2 Image examples 3

3 PDF styling 4

4 Kitchen Sink 12

5 Installation 46

6 Configuration 47

7 Page ordering 49

8 Custom roles 51

9 Translation 52

Python Module Index 54

Chapter 1

Videos

1.1 Youtube

Videos on YouTube or Vimeo can be embedded using the sphinxcontrib-youtube plugin:

```
.. youtube:: mL72ArkyAJs
```

https://youtu.be/mL72ArkyAJs

1.2 Other Services

Videos on other services can be embedded as long as there is an iframe option:

```
.. raw:: html

<iframe src="https://commons.wikimedia.org/wiki/File:Animation_Of_
↪Webb%27s_Orbit.webm?embedplayer=yes"
    width="512"
    height="288"
    frameborder="0"
    loading="lazy"
    allow="autoplay; picture-in-picture"
    allowfullscreen>
</iframe>
```

1.3 Videos not already online

If the video is not already on the web somewhere or there isn't a good way to embed it (e.g. if the embedding is serving adverts), then video can be added to the Sphinx site. To do this, place the video file in the docs/_static directory and add it to git.

The video can then be added to a page using the sphinxcontrib-video plugin . See the plugin's website for all available options and usage instructions.

Adding large (> 10MB) or multiple videos may make the repo quite large, and therefore slow to work with. If this becomes an issue, speak with a friendly dev about solutions.

```
.. video:: ../_static/Animation_of_Rotating_Earth_at_Night.webm
:align: center
```

(This is in the public domain because it was solely created by NASA. NASA copyright policy states that its material is not subject to copyright.)

Chapter 2

Image examples

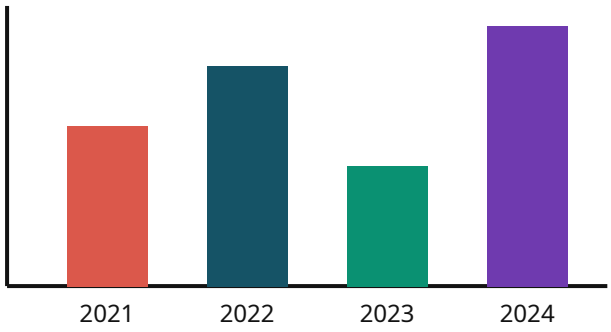
Raster images (PNG, JPEG) and vector images (SVG) both embed with the standard reStructuredText `image` and `figure` directives, and both render in the HTML site and the downloadable PDF.

2.1 SVG

SVG is a good choice for diagrams and charts, because it stays sharp at any size. The theme converts SVGs to PDF automatically during the PDF build (see *PDF styling*), so the same SVG works in both HTML and PDF - there's no need to keep a separate PNG copy.

```
.. image:: ../_static/svg-example.svg
   :alt: Bar chart drawn as a vector SVG in the IATI brand colours
   :width: 360
```

Example vector SVG



Chapter 3

PDF styling

PDF builds and HTML are styled completely separately: the website is styled with CSS, and the PDF is styled with LaTeX, a typesetting system with its own syntax. This page is a primer on how that works for anyone working on this theme who hasn't used LaTeX before.

The PDF-specific configuration lives in `iati_sphinx_theme/latex.py`.

3.1 How PDF output works

Building the website and building a PDF share the first step, then diverge completely:

1. Sphinx reads your `.rst` files.
2. Sphinx then writes out a `.tex` file: a plain-text document full of `commands`.
3. A LaTeX *engine* - a separate program - reads that `.tex` file and lays out the actual PDF pages.

`latex.py` only touches step 2. It doesn't write any of your content - it injects a block of setup commands, called a **preamble**, before Sphinx's generated content.

A handful of LaTeX terms cover almost everything in that preamble:

Term	What it means
preamble	The setup block before real content starts. Anything defined here - a colour, a font, a page margin - applies for the rest of the document.
`usepackage{x}`	An import. LaTeX ships with a small core; almost everything useful (colour, custom headers, hyperlinks, tidy tables) comes from a package you opt into by name.
`command{}`	A function call. Curly braces `{}` are required arguments; square brackets `[]` are optional ones. `definecolor{iatiorange}{HTML}{DB584B}` reads as "define a colour named `iatiorange`, from an HTML hex code, equal to `DB584B`."
`renewcommand`	Overrides a command LaTeX (or a package) already defines.
engine	The program that actually compiles the `.tex` file into a PDF. Sphinx defaults to `pdflatex`; this theme changes that default (see *Building PDFs locally*).

3.2 What `latex.py` configures

`latex.py` incorporates styling from the IATI design system. Most of it - the brand colours and the logo - is derived from the design system automatically at build time (see *Keeping in sync with the design system* below). The fonts are the one deliberate exception.

3.2.1 Brand colours

```
\usepackage{xcolor}
\definecolor{iatiorange}{HTML}{DB584B}
\definecolor{iatiteal}{HTML}{155366}
\definecolor{iatigrey}{HTML}{121212}
\definecolor{iatigreen}{HTML}{0A9172}
\definecolor{iatipurple}{HTML}{6F3AAF}
```

Same idea as CSS custom properties: each brand colour is defined once, then used by name in every block below instead of repeating hex codes.

These `definecolor` lines are generated from the design system's colour tokens at build time (see *Keeping in sync with the design system*).

3.2.2 Page geometry

```
\usepackage{geometry}
\geometry{
    a4paper,
    margin=2.5cm,
    top=3cm,
```

(continues on next page)

(continued from previous page)

```
    bottom=3cm
}
```

Sets the paper size and the white space around content.

3.2.3 Document layout

```
"extraclassoptions": "oneside,openany"
```

LaTeX's manual/report document classes default to book-style layout, but these PDFs are downloaded and read on a screen, so this line sets that up: `oneside` keeps margins identical on every page, and `openany` lets chapters start on the very next page instead of skipping one to land on an odd page number.

This is set via Sphinx's `extraclassoptions` `latex_elements` key (passed as document class options) rather than the raw preamble, since by the time the preamble runs, the document class has already been loaded with its default options.

3.2.4 Brand fonts

```
\setmainfont{NunitoSans}[...]
\setsansfont{HankenGrotesk}[...]
\setmonofont{RobotoMono}[...]
```

These are the brand fonts: Nunito Sans for body text; Hanken Grotesk for headings; Roboto Mono for code; provided by the `.ttf` files shipped in `iati_sphinx_theme/fonts/`.

Unlike the colours and logo, these font files are committed to the repository rather than pulled from Google Fonts at build time to ensure that there's always a local font file on disk for LaTeX to use. See *Keeping in sync with the design system*.

It's set via Sphinx's `fontpkg` `latex_elements` key rather than the raw preamble, since Sphinx already loads the `fontspec` package itself (as part of its own `fontenc` handling) when the engine is XeLaTeX or LuaLaTeX. `fontpkg` just needs to say *which* fonts to use.

Headings use `sffamily` to switch to Hanken Grotesk, since `setsansfont` binds a font to LaTeX's existing "sans family" concept rather than introducing a new command.

> **Note**
>
> `fontspec` (and therefore these brand fonts) only works under the XeLaTeX or LuaLaTeX engines, not the default `pdflatex` - see *Building PDFs locally*.

3.2.5 Header and footer

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
```

(continues on next page)

(continued from previous page)

```
\fancyhead[L]{\sffamily\textcolor{iatiorange}{\textbf{IATI}}}
\fancyhead[R]{\sffamily\textcolor{iatigrey-light}{\leftmark}}
\fancyfoot[C]{\thepage}
```

`[L]`, `[R]` and `[C]` mean left, right and centre. `leftmark` is a LaTeX built-in that always resolves to the current chapter's title, so chapter names in the header stay in sync automatically.

3.2.6 Chapter and section headings

```
\usepackage{titlesec}

\titleformat{\chapter}[display]
    {\normalfont\sffamily\huge\bfseries\color{iatiorange}}
    {\chaptertitlename\ \thechapter}
    {20pt}
    {\Huge}

\titleformat{\section}
    {\normalfont\sffamily\Large\bfseries\color{iatiteal}}
    {\thesection}
    {1em}
    {}
```

Chapters are large and orange; sections and subsections step down in size and switch to teal - the same hierarchy as the HTML theme, expressed in LaTeX's more verbose syntax.

3.2.7 Hyperlinks

```
\usepackage{hyperref}
\hypersetup{
    colorlinks=true,
    linkcolor=iatiteal,
    urlcolor=iatiorange,
    citecolor=iatigreen
}
```

Internal cross-references are teal, external URLs are orange, citations are green.

3.2.8 Custom title page

```
\renewcommand{\sphinxmaketitle}{
    \begin{titlepage}
        \centering
        \includegraphics[width=0.5\textwidth]{logo-colour.png}
        {\sffamily\Huge\bfseries\color{iatiorange}\@title\par}
        {\Large\color{iatigrey}\@author\par}
        {\large\color{iatigrey-light}\@date\par}
```

(continues on next page)

(continued from previous page)

```
    \end{titlepage}
}
```

This configures the title page to have a logo, a coloured title, and the author/date.

`@title` and `@author` aren't hardcoded - they come from Sphinx's `latex_documents` setting. In the standardised IATI setup the `conf.py` holding `latex_documents` is synced identically across repos, so it derives those values from the project's `project_info.py` (the `project` name and author) rather than having anyone hand-edit `conf.py`. (`@date` isn't part of `latex_documents` - it defaults to the build date.) So every project gets its own cover with the shared IATI look.

Important

This has to override `sphinxmaketitle`, **not** the standard `maketitle` command. Sphinx's document classes call `sphinxmaketitle` directly and never call plain `maketitle`, so a `renewcommand{maketitle}{...}` here would silently have no effect at all - which is exactly what happened until this was caught.

3.2.9 Tables, code blocks and admonitions

```
\usepackage{booktabs}
\renewcommand{\arraystretch}{1.3}
```

The rest is set via Sphinx's `sphinxsetup` element rather than raw LaTeX: nicer table rules, and colour-coded borders/backgrounds for `.. note::`, `.. warning::`, `.. tip::` and `.. important::` blocks, matching the same four brand colours used everywhere else in this document.

3.3 How it's wired into Sphinx

`latex.py` doesn't run on its own - `iati_sphinx_theme/__init__.py` connects its `configure_latex_defaults` function to Sphinx's `config-inited` event, a point early in the build where every project's `conf.py` has just finished executing, but nothing has been generated yet.

That function merges the theme's defaults into whatever `latex_elements` a project already set, without clobbering it:

- If a project's `conf.py` already sets a given `latex_elements` key, the project's value wins outright.
- The `preamble` key is the exception: the theme's preamble is *prepended* to the project's, so a project can add its own LaTeX underneath the branding without losing it.
- Likewise, a project's `sphinxsetup` string is *appended* to the theme's.

The same function also switches the default `latex_engine` from Sphinx's own default (`pdflatex`) to `lualatex`, but only if a project hasn't already chosen a different engine - see

below.

3.4 SVG images

Sphinx's LaTeX builder only accepts PDF, PNG and JPEG images, so an SVG referenced in your docs is silently dropped from the PDF - even though it renders fine on the website. To avoid that, the theme registers the `sphinxcontrib.rsvgconverter` extension automatically (in `iati_sphinx_theme/__init__.py`), which converts SVGs to PDF - as crisp vectors - during the PDF build using the `rsvg-convert` tool from librsvg.

The upshot: you can use SVG images in your docs and they render in both HTML and PDF, with no need to keep a separate PNG copy.

System requirement. `rsvg-convert` must be present in the build environment:

- Read the Docs - add it via `build.apt_packages` in `.readthedocs.yaml`:

```yaml
build:
  apt_packages:
    - librsvg2-bin
```

- Local, Debian/Ubuntu (including the devcontainer) - `apt-get install librsvg2-bin`.
- Local, macOS - `brew install librsvg`.

If `rsvg-convert` is missing, SVGs are simply skipped in the PDF (with a warning), exactly as before - HTML builds are never affected.

> **Note**
>
> Consumer sites get the extension automatically from the theme (it's a theme dependency, and the theme registers it - no `conf.py` change needed). Each site only needs to install `librsvg2-bin` in its own build environments: its `.readthedocs.yaml` and its devcontainer.

3.5 Keeping in sync with the design system

The theme consumes the IATI design system as an npm dependency, pinned in `package.json`. Its contract is the compiled stylesheet, `dist/css/iati.css`: `npm run build` compiles it into the theme's CSS via Sass (so the theme can `@extend` its classes), and the theme's own styles use its CSS custom properties with `var()`, e.g. `var(--color-grey-60)`.

A few things can't be taken straight from CSS, so `scripts/generate-brand-assets.mjs`, which runs as part of `npm run build`, derives them from the custom properties on `:root` in `iati.css` and from the package's assets:

- `iati_sphinx_theme/_generated/brand_colors.json` - the brand colours as hex codes. `latex.py` reads this to emit the `definecolor` lines and the admonition border colours.

- `styles/_generated/_breakpoints.scss` - screen breakpoints as Sass variables, because CSS custom properties can't be used in `@media` conditions.
- `iati_sphinx_theme/static/logo-colour.svg` and `logo-colour.png` - copied and rasterised from the design system's logo SVG. The HTML header uses the SVG; the PDF title page uses the PNG. (SVG images in docs *content* are converted to PDF automatically - see *SVG images* - but the title-page logo deliberately uses the pre-rasterised PNG.)
- `iati_sphinx_theme/static/iati.js` - the design system's JavaScript, served from `_static` rather than a CDN so it always matches the pin.

If a token the theme needs is missing from `iati.css`, the script fails and names it.

These are gitignored build artifacts, exactly like the compiled CSS - they aren't committed to this repository. They're regenerated wherever the CSS already is: local development, Read the Docs (in its `pre_install` step), and the PyPI publish workflow (before `python -m build`, so they're baked into the released package).

The pinned version is the unit of sync. Builds are reproducible against whatever `iati-design-system` version is pinned in `package.json`; brand values only change when someone bumps that pin deliberately - the same moment the CSS would change.

Dependabot (`.github/dependabot.yml`) opens a pull request bumping the pin whenever a new `iati-design-system` is published, including major versions. CI on that pull request is the compatibility check: it builds against the new version, and the job summary shows a diff of the generated assets against `main`, so a reviewer can see any brand-value changes before merging.

The fonts are the exception to all of this - see *Brand fonts* for why they're vendored rather than generated.

3.6 Building PDFs locally

```
make -C docs latexpdf
```

This requires a LaTeX distribution (for example TeX Live) that includes the `lualatex` engine, since the brand fonts depend on the `fontspec` package, which only works under XeLaTeX or LuaLaTeX - not the `pdflatex` engine Sphinx uses by default. This theme sets the default engine to `lualatex` automatically, so no extra configuration is needed in a project's `conf.py`.

Note

Of the two `fontspec`-compatible engines, this theme picks `lualatex` over `xelatex`: Sphinx's mechanism for force-wrapping long unbroken tokens (URLs, in particular) inside code blocks silently fails under XeLaTeX, letting them overflow the page instead of wrapping. The same content wraps correctly under LuaLaTeX.

3.7 Known limitations

Warning

Sphinx marks wrapped long lines in code blocks with a special character (U+2423, "open box"), which Roboto Mono doesn't include. LaTeX skips the missing glyph with a warning rather than failing the build, so wrapped code lines just lose that wrap indicator.

Chapter 4

Kitchen Sink

This section exists as a dump of all the things that Sphinx has.

It has exactly one goal: to be a good checklist of things to stylise within Sphinx. This also means that it is a complete showcase of *everything* that vanilla Sphinx provides and includes all sorts of edge cases.

(from <https://github.com/sphinx-themes/sphinx-themes.org>)

4.1 Admonitions

Sphinx provides several different types of admonitions.

4.1.1 `topic`

This is a topic.

This is what admonitions are a special case of, according to the docutils documentation.

4.1.2 `admonition`

The one with the custom titles

It's got a certain charm to it.

4.1.3 `attention`

Attention

Climate change is real.

4.1.4 `caution`

Caution

Cliff ahead: Don't drive off it.

4.1.5 danger

Danger

Mad scientist at work!

4.1.6 error

Error

Does not compute.

4.1.7 hint

Hint

Insulators insulate, until they are subject to _____ voltage.

4.1.8 important

Important

Tech is not neutral, nor is it apolitical.

4.1.9 note

Note

This is a note.

4.1.10 seealso

See also

Other relevant information.

4.1.11 tip

Tip

25% if the service is good.

4.1.12 todo

Todo

This needs the `sphinx.ext.todo` extension.

4.1.13 warning

Warning

Reader discretion is strongly advised.

4.1.14 versionadded

Added in version v0.1.1: Here's a version added message.

4.1.15 versionchanged

Changed in version v0.1.1: Here's a version changed message.

4.1.16 deprecated

Deprecated since version v0.1.1: Here's a deprecation message.

4.2 API documentation

Using Sphinx's `sphinx.ext.autodoc` plugin, it is possible to auto-generate documentation of a Python module.

Tip

Avoid having in-function-signature type annotations with autodoc, by setting the following options:

```
# -- Options for autodoc ------------------------------------------
 ↪---------
# https://www.sphinx-doc.org/en/master/usage/extensions/autodoc.html
 ↪#configuration

# Automatically extract typehints when specified and place them in
# descriptions of the relevant function/method.
autodoc_typehints = "description"
```

```
# Don't show class signature with the class' name.
autodoc_class_signature = "separated"
```

An extensible library for opening URLs using a variety of protocols

The simplest way to use this module is to call the urlopen function, which accepts a string containing a URL or a Request object (described below). It opens the URL and returns the results as file-like object; the returned object has some extra methods described below.

The OpenerDirector manages a collection of Handler objects that do all the actual work. Each Handler implements a particular protocol or option. The OpenerDirector is a composite object that invokes the Handlers needed to open the requested URL. For example, the HTTPHandler performs HTTP GET and POST requests and deals with non-error returns. The HTTPRedirectHandler automatically deals with HTTP 301, 302, 303, 307, and 308 redirect errors, and the HTTPDigestAuthHandler deals with digest authentication.

urlopen(url, data=None) – Basic usage is the same as original urllib. pass the url and optionally data to post to an HTTP URL, and get a file-like object back. One difference is that you can also pass a Request instance instead of URL. Raises a URLError (subclass of OSError); for HTTP errors, raises an HTTPError, which can also be treated as a valid response.

build_opener – Function that creates a new OpenerDirector instance. Will install the default handlers. Accepts one or more Handlers as arguments, either instances or Handler classes that it will instantiate. If one of the argument is a subclass of the default handler, the argument will be installed instead of the default.

install_opener – Installs a new opener as the default opener.

objects of interest:

OpenerDirector – Sets up the User Agent as the Python-urllib client and manages the Handler classes, while dealing with requests and responses.

Request – An object that encapsulates the state of a request. The state can be as simple as the URL. It can also include extra HTTP headers, e.g. a User-Agent.

BaseHandler –

internals: BaseHandler and parent _call_chain conventions

Example usage:

import urllib.request

set up authentication info authinfo = urllib.request.HTTPBasicAuthHandler() authinfo.add_password(realm='PDQ Application',

uri='https://mahler:8092/site-updates.py', user='klem', passwd='geheim$parole')

proxy_support = urllib.request.ProxyHandler({"http" : "http://ahad-haam:3128"})

build a new opener that adds authentication and caching FTP handlers opener = urllib.request.build_opener(proxy_support, authinfo,

urllib.request.CacheFTPHandler)

install it urllib.request.install_opener(opener)

f = urllib.request.urlopen('https://www.python.org/')

class urllib.request.**FancyURLopener**(**args*, ***kwargs*)

Derived class with handlers for errors we can handle (perhaps).

http_error_301(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*)

Error 301 – also relocated (permanently).

http_error_302(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*)

Error 302 – relocated (temporarily).

http_error_303(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*)

Error 303 – also relocated (essentially identical to 302).

http_error_307(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*)

Error 307 – relocated, but turn POST into error.

http_error_308(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*)

Error 308 – relocated, but turn POST into error.

http_error_401(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*, *retry=False*)

Error 401 – authentication required. This function supports Basic authentication only.

http_error_407(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*, *retry=False*)

Error 407 – proxy authentication required. This function supports Basic authentication only.

http_error_default(*url*, *fp*, *errcode*, *errmsg*, *headers*)

Default error handling – don't raise an exception.

prompt_user_passwd(*host*, *realm*)

Override this in a GUI environment!

class urllib.request.**HTTPDigestAuthHandler**(*passwd=None*)

An authentication protocol defined by RFC 2069

Digest authentication improves on basic authentication because it does not transmit passwords in the clear.

class urllib.request.**HTTPErrorProcessor**

Process HTTP error responses.

class urllib.request.**URLopener**(*proxies=None*, ***x509*)

Class to open URLs. This is a class rather than just a subroutine because we may need more than one set of global protocol-specific options. Note – this is a base class for those who don't want the automatic handling of errors type 302 (relocated) and 401 (authorization needed).

addheader(**args*)

Add a header to be used by the HTTP interface only e.g. u.addheader('Accept', 'sound/basic')

http_error(*url*, *fp*, *errcode*, *errmsg*, *headers*, *data=None*)

Handle http errors.

Derived class can override this, or provide specific handlers named http_error_DDD where DDD is the 3-digit error code.

http_error_default(*url*, *fp*, *errcode*, *errmsg*, *headers*)

Default error handler: close the connection and raise OSError.

open(*fullurl*, *data=None*)

Use URLopener().open(file) instead of open(file, 'r').

open_data(*url*, *data=None*)

Use "data" URL.

open_file(*url*)

Use local file or FTP depending on form of URL.

open_ftp(*url*)

Use FTP protocol.

open_http(*url*, *data=None*)

Use HTTP protocol.

open_https(*url*, *data=None*)

Use HTTPS protocol.

open_local_file(*url*)

Use local file.

open_unknown(*fullurl*, *data=None*)

Overridable interface to open unknown URL type.

open_unknown_proxy(*proxy*, *fullurl*, *data=None*)

Overridable interface to open unknown URL type.

retrieve(*url*, *filename=None*, *reporthook=None*, *data=None*)

retrieve(url) returns (filename, headers) for a local object or (tempfilename, headers) for a remote object.

urllib.request.**build_opener**(**handlers*)

Create an opener object from a list of handlers.

The opener will use several default handlers, including support for HTTP, FTP and when applicable HTTPS.

If any of the handlers passed as arguments are subclasses of the default handlers, the default handlers will not be used.

urllib.request.**getproxies**()

Return a dictionary of scheme -> proxy server URL mappings.

Scan the environment for variables named <scheme>_proxy; this seems to be the standard convention. If you need a different way, you can pass a proxies dictionary to the [Fancy]URLopener constructor.

`urllib.request.`**`pathname2url`**(*pathname*)

OS-specific conversion from a file system path to a relative URL of the 'file' scheme; not recommended for general use.

`urllib.request.`**`url2pathname`**(*pathname*)

OS-specific conversion from a relative URL of the 'file' scheme to a file system path; not recommended for general use.

`urllib.request.`**`urlcleanup`**()

Clean up temporary files from urlretrieve calls.

`urllib.request.`**`urlopen`**(*url*, *data=None*, *timeout=<object object>*, *, *context=None*)

Open the URL url, which can be either a string or a Request object.

data must be an object specifying additional data to be sent to the server, or None if no such data is needed. See Request for details.

urllib.request module uses HTTP/1.1 and includes a "Connection:close" header in its HTTP requests.

The optional *timeout* parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default timeout setting will be used). This only works for HTTP, HTTPS and FTP connections.

If *context* is specified, it must be a ssl.SSLContext instance describing the various SSL options. See HTTPSConnection for more details.

This function always returns an object which can work as a context manager and has the properties url, headers, and status. See urllib.response.addinfourl for more detail on these properties.

For HTTP and HTTPS URLs, this function returns a http.client.HTTPResponse object slightly modified. In addition to the three new methods above, the msg attribute contains the same information as the reason attribute — the reason phrase returned by the server — instead of the response headers as it is specified in the documentation for HTTPResponse.

For FTP, file, and data URLs and requests explicitly handled by legacy URLopener and FancyURLopener classes, this function returns a urllib.response.addinfourl object.

Note that None may be returned if no handler handles the request (though the default installed global OpenerDirector uses UnknownHandler to ensure this never happens).

In addition, if proxy settings are detected (for example, when a *_proxy environment variable like http_proxy is set), ProxyHandler is default installed and makes sure the requests are handled through the proxy.

`urllib.request.`**`urlretrieve`**(*url*, *filename=None*, *reporthook=None*, *data=None*)

Retrieve a URL into a temporary location on disk.

Requires a URL argument. If a filename is passed, it is used as the temporary file location. The reporthook argument should be a callable that accepts a block number, a read size, and the total file size of the URL target. The data argument should be valid URL encoded data.

If a filename is passed and the URL points to a local resource, the result is a copy from local file to new file.

Returns a tuple containing the path to the newly created data file as well as the resulting HTTPMessage object.

4.3 Blocks

4.3.1 Block Quotes

Block quotes consist of indented body elements:

> My theory by A. Elk. Brackets Miss, brackets. This theory goes as follows and begins now. All brontosauruses are thin at one end, much much thicker in the middle and then thin again at the far end. That is my theory, it is mine, and belongs to me and I own it, and what it is too.
>
> ---Anne Elk (Miss)

Epigraph

https://docutils.sourceforge.io/docs/ref/rst/directives.html#epigraph

> My theory by A. Elk. Brackets Miss, brackets. This theory goes as follows and begins now. All brontosauruses are thin at one end, much much thicker in the middle and then thin again at the far end. That is my theory, it is mine, and belongs to me and I own it, and what it is too.
>
> ---Anne Elk (Miss)

Pull quotes

https://docutils.sourceforge.io/docs/ref/rst/directives.html#pull-quote

> My theory by A. Elk. Brackets Miss, brackets. This theory goes as follows and begins now. All brontosauruses are thin at one end, much much thicker in the middle and then thin again at the far end. That is my theory, it is mine, and belongs to me and I own it, and what it is too.
>
> ---Anne Elk (Miss)

Highlights

https://docutils.sourceforge.io/docs/ref/rst/directives.html#highlights

> My theory by A. Elk. Brackets Miss, brackets. This theory goes as follows and begins now. All brontosauruses are thin at one end, much much thicker in the middle and then thin again at the far end. That is my theory, it is mine, and belongs to me and I own it, and what it is too.
>
> ---Anne Elk (Miss)

4.3.2 Line Blocks

https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#line-blocks

This is a normal text paragraph.

This is a line block. It ends with a blank line.

> Each new line begins with a vertical bar ("|").
> Line breaks and initial indents are preserved.

Continuation lines are wrapped portions of long lines; they begin with a space in place of the vertical bar.

> The left edge of a continuation line need not be aligned with the left edge of the text above it.

This is a second line block.

Blank lines are permitted internally, but they must begin with a "|".

This is a normal text paragraph again.

4.3.3 Monospace Blocks

Sphinx supports many kinds of monospace blocks. This section is meant to showcase *all* of them that as known to the author of this page, at the time of writing.

Production List

https://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html#directive-productionlist

> This directive is used to enclose a group of productions. Each production is given on a single line and consists of a name, separated by a colon from the following definition.

This just shows up as a vanilla `<pre>`, which is... both nice and a bit annoying.

```
try_stmt  ::= try1_stmt | try2_stmt
try1_stmt ::= "try" ":" suite
              ("except" [expression ["," target]] ":" suite)+
              ["else" ":" suite]
              ["finally" ":" suite]
try2_stmt ::= "try" ":" suite
              "finally" ":" suite
              "this-is-intentionally-very-stupidly-long-to-make-sure-that-this-
```

Literal Blocks

https://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html#literal-blocks

> contains a block of text where line breaks and whitespace are significant and must be preserved

This is a normal text paragraph. The next paragraph is a code sample:

```
It is not processed in any way, except
that the indentation is removed.

It can span multiple lines.
```

This is a normal text paragraph again.

They can be quoted without indentation:

```
>> Great idea!
>
> Why didn't I think of that?
```

Listing 1: Literal includes can also have captions.

```
import sys

import sphinx.application

# Make project_info importable regardless of how Sphinx is invoked.
 ↪`python -m
# sphinx`/sphinx-autobuild add the current working directory to sys.
 ↪path
# automatically when run from this directory, but installed console
 ↪scripts
# like `sphinx-build` (e.g. via `make latexpdf`) don't - their sys.
 ↪path[0] is
# their own bin directory instead.
sys.path.insert(0, os.path.dirname(os.path.abspath(__file__)))

```

Doctest Blocks

https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#doctest-blocks

> Doctest blocks are interactive Python sessions cut-and-pasted into docstrings. They are meant to illustrate usage by example, and provide an elegant and powerful testing environment via the doctest module in the Python standard library.

Note

This is fine.

```
>>> print('Python-specific usage examples; begun with ">>>"')
Python-specific usage examples; begun with ">>>"
>>> print("(cut and pasted from interactive Python sessions)")
(cut and pasted from interactive Python sessions)
>>> print("This is an intentionally very long line because I want to
 ↪make sure that we are handling scrollable code blocks correctly.")
This is an intentionally very long line because I want to make sure
 ↪that we are handling scrollable code blocks correctly.
```

Parsed Literals

https://docutils.sourceforge.io/docs/ref/rst/directives.html#parsed-literal-block

> It is equivalent to a line block with different rendering: typically in a typewriter/monospaced typeface, like an ordinary literal block. Parsed literal blocks are useful for adding hyperlinks to code examples.

```
# parsed-literal test
curl -O http://someurl/release-0.1.0.tar-gz
echo "This is an intentionally very long line because I want to make
 ↪sure that we are handling scrollable code blocks correctly."
```

Code Block

https://docutils.sourceforge.io/docs/ref/rst/directives.html#code

> The “code” directive constructs a literal block [containing code].

This has an alias of `code-block`.

```
1  from typing import Iterator
2
3  # This is an example
4  class Math:
5      @staticmethod
6      def fib(n: int) -> Iterator[int]:
7          """Fibonacci series up to n"""
8          a, b = 0, 1
9          while a < n:
10             yield a
11             a, b = b, a + b
12
13
14 result = sum(Math.fib(42))
15 print("The answer is {}".format(result))
```

With caption

Listing 2: Code Blocks can have captions, which also adds a link to it.

```
{
  "session_name": "shorthands",
  "windows": [
    {
      "panes": [
        {
          "shell_command": "echo 'This is an intentionally very long
↪line because I want to make sure that we are handling scrollable
↪code blocks correctly.'"
        }
      ],
      "window_name": "long form"
    }
  ]
}
```

With line numbers

```
1  def some_function():
2      interesting = False
3      print("This line is highlighted.")
4      print("This one is not...")
5      print("...but this one is.")
6      print(
7          "This is an intentionally very long line because I want to
↪make sure that we are handling scrollable code blocks correctly."
8      )
```

With none highlighting

```
# Taken from https://en.wikipedia.org/wiki/Pseudocode#Example
algorithm ford-fulkerson is
    input: Graph G with flow capacity c,
        source node s,
        sink node t
    output: Flow f such that f is maximal from s to t

    (Note that f(u,v) is the flow from node u to node v, and c(u,v)
↪is the flow capacity from node u to node v)

    for each edge (u, v) in GE do
        f(u, v)   0
```

(continues on next page)

(continued from previous page)

```
        f(v, u)   0

    while there exists a path p from s to t in the residual network
↪Gf do
        let cf be the flow capacity of the residual network Gf
        cf(p)   min{cf(u, v) | (u, v) in p}
        for each edge (u, v) in p do
            f(u, v)    f(u, v) + cf(p)
            f(v, u)   -f(u, v)

    return f
```

4.4 Generic items

These are the things that don't quite fit into other groupings.

4.4.1 Inline Markup

One of the nice things about markup languages is the ability to have inline markup. This makes the presentation *much* nicer. The **bold** text shouldn't be too overbearing. It is very common to have `inline code` as well. It is important to ensure that the inline code doesn't have a line height that is greater than the regular prose; otherwise the spacing looks weird.

It is also possible to use explicit roles to do things like a subscript, a superscript, *emphasis*, **strong**, and `literal`.

Hyperlinks

It is a website, so it'll have hyperlinks like http://www.python.org (inline), or Python[2] (external reference), *example* (internal reference), Python web site (external hyperlinks with embedded URI), footnote references (manually numbered **[1]_**, anonymous auto-numbered **[#]_**, labeled auto-numbered **[#label]_**, or symbolic **[*]_**), citation references ([12]), substitution references (), and inline hyperlink targets (see *Targets* below for a reference back to here).

reStructuredText has character-level inline markup too. It's ugly to write, but someone might be using it, so here's an example: **re`Structured`***Text*.

It is also possible to link to documented items, such as `api_sample.RandomNumberGenerator`.

[2] http://www.python.org/

[12] This citation has some `code blocks` in it, maybe some **bold** and *italics* too. Heck, lets put a link to a meta citation[13] too.

[13] This citation will have one backlink.

Interpreted text

The default role for "interpreted text" (AKA single backticks) is *Title Reference*. There are other reference syntaxes as well: PEP 287 and RFC 2822.

If the `--pep-references` option was supplied, there should be a live link to PEP 258 here.

GUI labels

According to the RST demo, GUI labels (like *this label*) are a way to indicate that some action is to be taken by the user. Like inline code literals, GUI labels should not run over line height.

Keys / Menu labels

Key-bindings indicate that the read is to press a button on the keyboard or mouse, for example `MMB`, `C-x C-f` and `Shift-MMB`. Another useful way is `menuselection` to show menus: *My* ▯ *Software* ▯ *Some menu* ▯ *Some sub menu 1* ▯ *Some sub menu 2* ▯ *Some sub menu 3*

For example, `menuselection` should break when it is too long to fit on a single line.

Long inline code wrapping

Let's test wrapping and whitespace significance in inline literals: `This is an example of --inline-literal --text, --including some-- strangely--hyphen-ated-words. Adjust-the-width-of-your-browser-window to see how the text is wrapped. -- ---- -------- Now note the spacing between the words of this sentence (words should be grouped in pairs).`

4.4.2 Math

This is a test. Here is an equation: $X_{0:5} = (X_0, X_1, X_2, X_3, X_4)$. Here is another:

$$\nabla^2 f = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial f}{\partial \theta}\left(\sin\theta\,\frac{\partial f}{\partial \theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 f}{\partial \phi^2} \tag{4.1}$$

You can add a link to equations like the one above (4.1) by using `:eq:`.

4.4.3 Sidebar

Ch'ien / The Creative

Lorem ipsum dolor sit amet consectetur adipisicing elit.

Lorem ipsum dolor sit amet consectetur adipisicing elit.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Code with Sidebar

A code example

With a sidebar on the right.

Listing 3: Code blocks can also have captions.

```
print("one")
print("two")
print("three")
print("four")
print("five")
print("six")
print("seven")
print("eight")
print("nine")
print("ten")
print("eleven")
print("twelve")
print("thirteen")
print("fourteen")
```

4.4.4 References

Footnotes

Here we have a footnote.[1]

Automatically numbered footnotes have a marking in place in the source paragraph text like

```
[#doubleColon]_
```

and at the end of the section set off footnote text like

```
.. [#doubleColon]
   If you look at the source, you see that we are using a
   double colon :: before the set off text to make it literal.
```

The original footnote reference inside the paragraph is bracketed by `[#` and `]_`. This follow the basic pattern in restructured text to use concise symbol combinations inside a paragraph that are unlikely to appear in normal text.

There is also simple markup to do hyperlinks, both to web URL's and as cross references within a Sphinx document, like to *Footnotes*.

More elaborate documents by some of the authors are http://introcs.cs.luc.edu and Dr. Harrington's Python Tutorial.

[1] If you look at the source, you see that we are using a double colon :: before the set off text to make it literal.

Citations

Here's a reference to the above,[Page 24, 12] citation.

Here is another type of citation: *citation*

Targets

This paragraph is pointed to by the explicit "example" target. A reference can be found under *Inline Markup*, above. *Inline hyperlink targets* are also possible.

Section headers are implicit targets, referred to by name. See *Targets*.

Explicit external targets are interpolated into references such as "Python[Page 24, 2]".

Targets may be indirect and anonymous. Thus *this phrase* may also refer to the *Targets* section.

Target Footnotes

4.4.5 Centered text

You can create a statement with centered text with `.. centered::`

This is centered text!

4.4.6 Rubric

> A rubric is like an informal heading that doesn't correspond to the document's structure.
>
> ---https://docutils.sourceforge.io/docs/ref/rst/directives.html#rubric

Wikipedia says it is something different:

> A rubric is a word or section of text that is traditionally written or printed in red ink for emphasis.
>
> ---https://en.wikipedia.org/wiki/Rubric

This is stylized as docutils tells us to stylize it, since it is used for footnote headers (see end of https://docs.python.org/3/reference/lexical_analysis.html)

This is a rubric

Sidebars and Rubrics

Sidebar Title

Optional Subtitle

This is a sidebar. It is for text outside the flow of the main text.

This is a rubric inside a sidebar

Sidebars often appears beside the main text with a border and background color.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

4.4.7 Download Links

`This long long long long long long long long long long long long long long long download link should wrap white-spaces`

4.5 Images & Figures

4.5.1 Images

An image:

A clickable image:

This is a lot of text to go along with a right-aligned image, that is helping make this content feel less linear. It is important to have such a body of text, since the image is meant to be “floated” to the right, which would interfere with the rest of the document otherwise.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis sapiente veritatis doloribus accusantium molestiae modi recusandae excepturi facere, corrupti expedita sit nihil temporibus eius sequi animi, illo libero labore fuga.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis sapiente veritatis doloribus accusantium molestiae modi recusandae excepturi facere, corrupti expedita sit nihil temporibus eius sequi animi, illo libero labore fuga.

This is a lot of text to go along with a left-aligned image, that is helping make this content feel less linear. It is important to have such a body of text, since the image is meant to be “floated” to the right, which would interfere with the rest of the document otherwise.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis sapiente veritatis doloribus accusantium molestiae modi recusandae excepturi facere, corrupti expedita sit nihil temporibus eius sequi animi, illo libero labore fuga.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis sapiente veritatis doloribus accusantium molestiae modi recusandae excepturi facere, corrupti expedita sit nihil temporibus eius sequi animi, illo libero labore fuga.

4.5.2 Figures

Fig. 1: A figure is an image with a caption and/or a legend:

re	Revised, revisited, based on 're' module.
Structured	Structure-enhanced text, structuredtext.
Text	Well it is, isn't it?

This paragraph is also part of the legend.

A figure directive with center alignment

Fig. 2: This caption should be centered.

4.6 Lists

4.6.1 Enumerated Lists

1. Start each line with a number and period
2. Can begin on any number
3. Must have a blank line before and after
4. Can have nested sub-lists
    a. nested lists are numbered separately
    b. nested lists need a blank line before and after
        A. Upper Alpha
5. Roman numerals can also be used
    i. nested lists are numbered separately
    ii. nested lists need a blank line before and after
6. Upper Roman
    I. first item
    II. second item
7. Can have automatic number with the # character.

4.6.2 Definition Lists

Term
: Definition

Term
: [classifier] Definition paragraph 1.

  Definition paragraph 2.

Term
Definition

I have no clue why the definition list below is classified as a different style of definition list than the one above.

Is it the spaces in the term?
Maybe it was the multiple line paragraph in the line below that caused this?

Is it the paragraph above the list maybe?
I guess a lot of these lists don't have leading paragraphs?

Is it everything all at once?
Who knows?!

4.6.3 Option Lists

For listing command-line options:

-a	command-line option "a"
-b file	options can have arguments and long descriptions
--long	options can be long also
--input=file	long options can also have arguments
--very-long-option	The description can also start on the next line.
	The description may contain multiple body elements, regardless of where it starts.
-x, -y, -z	Multiple options are an "option group".
-v, --verbose	Commonly-seen: short & long options.
-1 file, --one=file, --two file	Multiple options with arguments.
/V	DOS/VMS-style options too

There must be at least two spaces between the option and the description.

4.6.4 Field list

Address
123 Example Street Example, EX Canada A1B 2C3

Organization
humankind

Status
This is a "work in progress"

Version
1

Field name
This is a generic bibliographic field.

Field name 2
: Generic bibliographic fields may contain multiple body elements.

  Like this.

4.6.5 Glossary

This is a glossary with definition terms for thing like *Writing*:

Documentation
: Provides users with the knowledge they need to use something.

Reading
: The process of taking information into ones mind through the use of eyes.

Writing
: The process of putting thoughts into a medium for other people to *read*.

Here's another glossary, with more detail. The important bit here is that the contents of the definition are multi-paragraph.

Import Package
: A Python module which can contain other modules or recursively, other packages.

  An import package is more commonly referred to with the single word "package", but this guide will use the expanded term when more clarity is needed to prevent confusion with a Distribution Package which is also commonly called a "package".

Package Index
: A repository of distributions with a web interface to automate package discovery and consumption.

4.6.6 Bullet Lists

Simple

- A simple list.
- There are no margins between list items.
- Simple lists do not contain multiple paragraphs. That's a complex list.
- In the case of a nested list
  - There are no margins between elements
    - Still no margins
      - Still no margins

Complex

- A bullet list
  - Nested bullet list.
  - Nested item 2.

- Item 2.

  Paragraph 2 of item 2.

  - Nested bullet list.
  - Nested item 2.
    - Third level.
    - Item 2.
  - Nested item 3.
- `inline literall`
- `inline literall`
- `inline literall`
- This item has multiple paragraphs.

  This item has multiple paragraphs.
- This item has multiple paragraphs.

  This item has multiple paragraphs.

Second list level

- here is a list in a second-level section.
- yahoo
- yahoo
  - yahoo
  - here is an inner bullet oh
    - one more `with an inline literally`. yahoo

      heh heh. child. try to beat this embed:

```
import sys

import sphinx.application

# Make project_info importable regardless of how Sphinx is
 ↪invoked. `python -m
# sphinx`/sphinx-autobuild add the current working
 ↪directory to sys.path
# automatically when run from this directory, but
 ↪installed console scripts
# like `sphinx-build` (e.g. via `make latexpdf`) don't -
 ↪their sys.path[0] is
# their own bin directory instead.
sys.path.insert(0, os.path.dirname(os.path.abspath(__file_
```

(continues on next page)

(continued from previous page)

```
↪_)))
```

  - and another. yahoo
  - yahoo
  - `hi`

- how about an admonition?

  Note

  This is a note nested in a list.

- and hehe

But deeper down the rabbit hole

- I kept saying that, “deeper down the rabbit hole”. yahoo
  - I cackle at night yahoo.
- I’m so lonely here in GZ guangzhou
- A man of python destiny, hopes and dreams. yahoo
  - yahoo
    * yahoo `hi`
    * `destiny`

4.6.7 Hlists

- First item
- Second item
- Third item
- Forth item
- Fifth item
- Sixths item

Hlist with images

-
-

4.6.8 Numbered List

1. One,
2. Two.
3. Three with long text. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed feugiat sagittis neque quis eleifend. Duis rutrum lectus sit amet mattis suscipit.

- A) Using bullets and letters. (A)
- B) Using bullets and letters. (B)
- C) Using bullets and letters. (C)

Mimicking ReadTheDocs' PR previews

This is simulating how ReadTheDocs puts admonitions on top of previews.

4.7 Really Long Page Title because we should test sidebar wrapping

This page basically exists to show-off the cool table of contents sidebar.

4.7.1 Heading 1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.2 Heading 2

Let's make sure multiple permalinks work: *one*, *two*, *three*.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.3 Heading 3

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.4 Heading 4

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.5 Heading 5

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.6 Heading 6

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.7 Heading 7

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

Heading 7.1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

Heading 7.2

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

Heading 7.3

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

Heading 7.4

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.8 Heading 8

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.9 Heading 9

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.10 Heading 10

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.11 Heading 11

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.12 Heading 12

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.13 Heading 13

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.14 Heading 14

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.15 Heading 15 – this one is intentionally really long

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.16 Heading 16 – this one is also lots of words that would wrap content

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.17 Heading 17

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.18 Heading 18

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.19 Heading 19

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.20 Heading 20

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.21 Heading 21

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.22 Heading 22

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.23 Heading 23

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.24 Heading 24

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.25 Heading 25

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.26 Heading 26

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.27 Heading 27

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.28 Heading 28

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.29 Heading 29

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.30 Heading 30

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.31 Heading 31

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.32 Heading 32

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.33 Heading 33

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.34 Heading 34

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.35 Heading 35

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.36 Heading 36

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.37 Heading 37

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.38 Heading 38

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.39 Heading 39

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.40 Heading 40

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.41 Heading 41

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.42 Heading 42

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.7.43 Heading 43

Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque sit temporibus cupiditate in ducimus illum assumenda dolor, dignissimos laboriosam voluptate dolorem dolore eum repellendus minima, nisi sequi? Eveniet, dignissimos asperiores!

4.8 Structural Elements

Attention

This warning box is repeated across several pages. It is maintained in a single dedicated file 'thingtoinclude.rst' and embedded where needed using the directive '.. include:: thingtoinclude.rst'

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec lorem neque, interdum in ipsum nec, finibus dictum velit. Ut eu efficitur arcu, id aliquam erat. In sit amet diam gravida, imperdiet tellus eu, gravida nisl. Praesent aliquet odio eget libero elementum, quis rhoncus tellus tincidunt. Suspendisse quis volutpat ipsum. Sed lobortis scelerisque tristique. Aenean condimentum risus tellus, quis accumsan ipsum laoreet ut. Integer porttitor maximus suscipit. Mauris in posuere sapien. Aliquam accumsan feugiat ligula, nec fringilla libero commodo sed. Proin et erat pharetra.

Etiam turpis ante, luctus sed velit tristique, finibus volutpat dui. Nam sagittis vel ante nec malesuada. Praesent dignissim mi nec ornare elementum. Nunc eu augue vel sem dignissim cursus sed et nulla. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque dictum dui sem, non placerat tortor rhoncus in. Sed placerat nulla at rhoncus iaculis.

4.8.1 Document Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed condimentum nulla vel neque venenatis, nec placerat lorem placerat. Cras purus eros, gravida vitae tincidunt id, vehicula nec nulla. Fusce aliquet auctor cursus. Phasellus ex neque, vestibulum non est vitae, viverra fringilla tortor. Donec vestibulum convallis justo, a faucibus lorem vulputate vel. Aliquam cursus odio eu felis sodales aliquet. Aliquam erat volutpat. Maecenas eget dictum mauris. Suspendisse arcu eros, condimentum eget risus sed, luctus efficitur arcu. Cras ut dictum mi. Nulla congue interdum lorem, semper semper enim commodo nec.

Document Subsection

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam efficitur in eros et blandit. Nunc maximus, nisl at auctor vestibulum, justo ex sollicitudin ligula, id faucibus urna orci tristique nisl. Duis auctor rutrum orci, in ornare lacus condimentum quis. Quisque arcu velit, facilisis quis interdum ac, hendrerit auctor mauris. Curabitur urna nibh, porttitor at ante sit amet, vestibulum interdum dolor. Duis dictum elit orci, tincidunt imperdiet sem pellentesque et. In vehicula pellentesque varius. Phasellus a turpis sollicitudin, bibendum massa et, imperdiet neque. Integer quis sapien in magna rutrum bibendum. Integer cursus ex sed magna vehicula finibus. Proin tempus orci quis dolor tempus, nec condimentum odio vestibulum. Etiam efficitur sollicitudin libero, tincidunt volutpat ligula interdum sed.

Document Subsubsection

Donec non rutrum lorem. Aenean sagittis metus at pharetra fringilla. Nunc sapien dolor, cursus sed nisi at, pretium tristique lectus. Sed pellentesque leo lectus, et convallis ipsum euismod a. Integer at leo vitae felis pretium aliquam fringilla quis odio. Sed pharetra enim accumsan feugiat pretium. Maecenas at pharetra tortor. Morbi semper eget mi vel finibus. Cras rutrum nulla eros, id feugiat arcu pellentesque ut. Sed finibus tortor ac nisi ultrices viverra. Duis feugiat malesuada sapien, at commodo ante porttitor ac. Curabitur posuere mauris mi, vel ornare orci scelerisque sit amet. Suspendisse nec fringilla dui.

Document Paragraph

Pellentesque nec est in odio ultrices elementum. Vestibulum et hendrerit sapien, quis vulputate turpis. Suspendisse potenti. Curabitur tristique sit amet lectus non viverra. Phasellus rutrum dapibus turpis sed imperdiet. Mauris maximus viverra ante. Donec eu egestas mauris. Morbi vulputate tincidunt euismod. Integer vel porttitor neque. Donec at lacus suscipit, lacinia lectus vel, sagittis lectus.

4.9 Structural Elements 2

Attention

This warning box is repeated across several pages. It is maintained in a single dedicated file 'thingtoinclude.rst' and embedded where needed using the directive '.. include:: thingtoinclude.rst'

Etiam turpis ante, luctus sed velit tristique, finibus volutpat dui. Nam sagittis vel ante nec malesuada. Praesent dignissim mi nec ornare elementum. Nunc eu augue vel sem dignissim cursus sed et nulla. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque dictum dui sem, non placerat tortor rhoncus in. Sed placerat nulla at rhoncus iaculis.

4.9.1 Document Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed condimentum nulla vel neque venenatis, nec placerat lorem placerat. Cras purus eros, gravida vitae tincidunt id, vehicula nec nulla. Fusce aliquet auctor cursus. Phasellus ex neque, vestibulum non est vitae, viverra fringilla tortor. Donec vestibulum convallis justo, a faucibus lorem vulputate vel. Aliquam cursus odio eu felis sodales aliquet. Aliquam erat volutpat. Maecenas eget dictum mauris. Suspendisse arcu eros, condimentum eget risus sed, luctus efficitur arcu. Cras ut dictum mi. Nulla congue interdum lorem, semper semper enim commodo nec.

Document Subsection

Fig. 5: This is a caption for a figure. Text should wrap around the caption.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam efficitur in eros et blandit. Nunc maximus, nisl at auctor vestibulum, justo ex sollicitudin ligula, id faucibus urna orci tristique nisl. Duis auctor rutrum orci, in ornare lacus condimentum quis. Quisque arcu velit, facilisis quis interdum ac, hendrerit auctor mauris. Curabitur urna nibh, porttitor at ante sit amet, vestibulum interdum dolor. Duis dictum elit orci, tincidunt imperdiet sem pellentesque et. In vehicula pellentesque varius. Phasellus a turpis sollicitudin, bibendum massa et, imperdiet neque. Integer quis sapien in magna rutrum bibendum. Integer cursus ex sed magna vehicula finibus. Proin tempus orci quis dolor tempus, nec condimentum odio vestibulum. Etiam efficitur sollicitudin libero, tincidunt volutpat ligula interdum sed. Praesent congue sagittis nisl et suscipit. Vivamus sagittis risus et egestas commodo.Cras venenatis arcu in pharetra interdum. Donec quis metus porttitor tellus cursus lobortis. Quisque et orci magna. Fusce rhoncus mi mi, at vehicula massa rhoncus quis. Mauris augue leo, pretium eget molestie vitae, efficitur nec nulla. In hac habitasse platea dictumst. Sed sit amet imperdiet purus.

4.10 Tables

4.10.1 Grid Tables

Here's a grid table followed by a simple table:

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	Cells may span columns.		
body row 3	Cells may span rows.	• Table cells • contain • body elements.	
body row 4			
body row 5	Cells may also be empty: `-->`		

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

Giant Tables

Header 1	Heade 2	Heade 3	Header 1	Heade 2	Heade 3	Header 1	Heade 2	Heade 3	Header 1	Heade 2	Header 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3
body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3	body row 1	column 2	column 3

4.10.2 Table containing code

Version	Installing
Pradyun's pip fork and installer	`pip install "pip @ git+https://github.com/pradyunsg/pip#20.3.3" "installer @ git+https://github.com/pradyunsg/installer"`
PyPI	`pip install pip installer`

4.10.3 List Tables

Table 1: List tables can have captions like this one.

List table	Head 1	Header 2	Header 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.
Stub Row 1	Row 1	Column 2	Column 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.
Stub Row 2	Row 2	Column 2	Column 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.
Stub Row 3	Row 3	Column 2	Column 3 long. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sit amet mauris arcu.

Table 2: This is a list table with images in it.

Fig. 6: This is a short caption for a figure.	Fig. 7: This is a long caption for a figure. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec porttitor dolor in odio posuere, vitae ornare libero mattis. In lobortis justo vestibulum nibh aliquet, non.

4.11 Typography

This is quite important, for a website where the majority of the content is going to be prose.

Notice the font family being used for the prose, as well as the font family being used for the heading. Think about the spacing between the lines, as well as the spacing between various paragraphs. Also keep the font weight in mind, and consider if/how you want antialiasing and font-smoothing to work.

Multiple paragraphs are a common occurance, because you often need more than a single paragraph to describe a thing. The rest of this paragraph is gonna be the famous lorem-ipsum: Lorem ipsum **dolor** sit amet consectetur adipisicing elit. Accusamus, sunt voluptatum tenetur

libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

4.11.1 Headings

This next bit will explore how the various headings look. Think about how the content separation should work, and how the various headings should interact with the main content.

Heading 3

Heading 4

Heading 5

Heading 6

4.11.2 Heading 2 with content

Lorem ipsum **dolor** sit amet consectetur adipisicing elit.

Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Heading 3 with content

Lorem ipsum **dolor** sit amet consectetur adipisicing elit.

Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Heading 4 with content

Lorem ipsum **dolor** sit amet consectetur adipisicing elit.

Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Heading 5 with content

Lorem ipsum **dolor** sit amet consectetur adipisicing elit.

Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Heading 6 with content

Lorem ipsum **dolor** sit amet consectetur adipisicing elit.

Accusamus, sunt voluptatum tenetur libero nulla esse veritatis accusantium earum commodi hic voluptatem officia culpa optio atque. Quaerat sed quibusdam ratione nam.

Chapter 5

Installation

1. Install the theme in your Sphinx project.

   ```
   pip install iati-sphinx-theme
   ```

2. In your project's `conf.py` set the `html_theme`.

   ```
   html_theme = "iati_sphinx_theme"
   ```

Chapter 6

Configuration

This theme has multiple options, which can be configured using the `html_theme_options` object in your `conf.py` file.

```
html_theme_options = {
  "github_repository": "https://github.com/organisation/repository",
  "header_title_text": "IATI Sphinx Theme",
  "header_eyebrow_text": "IATI Tools: Documentation",
  "languages": ["en", "fr", "es"],
  "plausible_domain": "example.com",
  "project_title": "IATI Sphinx Theme: Documentation",
  "tool_nav_items": {
    "IATI Tool": "https://tool.iatistandard.org/"
  },
}
```

There is more information on each option below.

6.1 github_repository

This should be a link to the Github repository for the documentation site, and is used to link to the source code in the footer of the site.

6.2 header_title_text

The title text to display in the header. Defaults to the value of `project` in `conf.py`.

6.3 header_eyebrow_text

The text above the title in the header. Defaults to “IATI Tools: Documentation”.

6.4 languages

A list of languages which the documentation is available in, used to populate the language switcher component. Defaults to English only.

6.5 plausible_domain

To integrate with Plausible Analytics, add the `plausible_domain` option in your project's `conf.py` file.

If your docs site is a subdomain for the site it is documenting, use the top level domain for cross-subdomain tracking. For example, for the Sphinx site `docs.example.com`, use `example.com` as your `plausible_domain`.

```
html_theme_options = {
  "plausible_domain": "example.com"
}
```

6.6 project_title

The text to use in the breadcrumb and tool navigation components.

6.7 tool_nav_items

Items to include in the header tool nav bar. Defaults to a single item that links back to the site's homepage.

Chapter 7

Page ordering

Pages appear - in the sidebar, and everywhere else described below - in the order they're listed inside `toctree::` directives. Order is **not** derived from alphabetical order, file names, or folder structure; it's simply the order documents are listed in the toctree source.

```
.. toctree::

   installation
   configuration
   faq
```

Here, `installation` comes before `configuration` because it's listed first - regardless of what the files are named or where they live on disk.

7.1 Grouping pages under a caption

A page can contain more than one `toctree::` directive. Each one can have its own `:caption:`, which becomes a labelled group in the sidebar, in the order the `toctree::` directives themselves appear on the page. This theme's own docs use exactly this pattern:

```
.. toctree::
   :caption: Examples

   examples/videos

.. toctree::
   :caption: Development

   pdf-styling
   kitchen-sink/index
```

7.2 Where else page order matters

The same toctree order drives more than just the sidebar:

- **PDF chapters** - since `docs/conf.py`'s `latex_documents` starts from `index`, the PDF build walks the same toctree to decide chapter order. A page moved earlier in the toctree moves earlier in the PDF too.
- **Previous/Next links** - when the `show_relations` theme option is enabled, the footer's Previous/Next links are computed by flattening the toctree, so they follow the same order.

Two things toctree order does **not** affect:

- **Breadcrumbs** - these always show `Home` and the current page's title, regardless of how deeply a page is nested in a toctree.
- **Search results** - these are ranked by Sphinx's search index, not by toctree position.

7.3 A `:glob:` gotcha

The `:glob:` option lets a toctree entry be a wildcard pattern (for example `kitchen-sink/*`), and *only* wildcard entries are sorted alphabetically. If you write out explicit filenames under `:glob:` rather than a wildcard, as this theme's own kitchen sink index does, they still follow plain source order - it's easy to assume `:glob:` alphabetises everything, but it only applies to genuine wildcards.

Chapter 8

Custom roles

8.1 `iati-reference`

When referencing elements of the standard, use the `iati-reference` role. For example:

```
:iati-reference:`iati-activities/iati-activity/iati-identifier`
```

The code above will apply the following styles:

iati-activities/iati-activity/iati-identifier

Chapter 9

Translation

The IATI Sphinx Theme is translatable. The following languages are currently supported:

- English (en)
- French (fr)
- Spanish (es)

9.1 Built-in strings

To enable translation of built-in strings, you must add the theme to the `locale_dirs` list in your `conf.py`.

```
import os
import iati_sphinx_theme

local_dirs = [
  "locale",
  os.path.join(os.path.dirname(iati_sphinx_theme.__file__), "locale")
]
```

9.2 User-defined strings

User-defined strings must be translated by the user of the theme. These are configured in your `conf.py` file under `html_theme_options`. In order to translate these, complete the following steps in the same location as the *conf.py* file:

1. Mark strings as translatable using `sphinx.locale.get_translation`, usually renamed to `_()`.

```
import os
from sphinx.locale import get_translation

MESSAGE_CATALOG_NAME = "iati-sphinx-theme"
```

(continues on next page)

(continued from previous page)

```
_ = get_translation(MESSAGE_CATALOG_NAME)

html_theme_options = {
    "header_title_text": _("Title"),
    "tool_name": _("IATI Example Tool"),
}

def setup(app):
    locale_path = os.path.join(os.path.abspath(os.path.dirname(__
↪file__)), "locale")
    app.add_message_catalog(MESSAGE_CATALOG_NAME, locale_path)
```

2. Extract translatable strings to `locale/iati-sphinx-theme.pot`.

```
pybabel extract conf.py -o locale/iati-sphinx-theme.pot
```

3. Create or update `.po` files for desired languages.

```
# To add a new language
pybabel init -i locale/iati-sphinx-theme.pot -d locale --domain=iati-
↪sphinx-theme -l es

# To update an existing language
pybabel update -i locale/iati-sphinx-theme.pot -d locale --
↪domain=iati-sphinx-theme -l es
```

4. Compile the `.mo` files. This must be done before Sphinx builds the project, for example using Read the Docs' pre_build job, or by compiling and committing the files to version control.

```
pybabel compile -d locale --domain=iati-sphinx-theme
```

5. Continue with your project's usual translation workflow.

Python Module Index

u

urllib.request, 15