Curriculum

Contents

Chapter 1. Technical communication at Vistula

Technical communication is the process of defining, creating and delivering information products for the safe, efficient and effective use of products (technical systems, software, services).

Technical communication

Technical communicators may put the information they capture into paper documents, web pages, computer-based training, digitally stored text, audio, video, and other media.

Today, effective technical communication is more important than ever. We live in an age in which whole industries are built around the development, retention and application of information.A technical writer is a professional writer that communicates complex information. Essentially, technical writers break down complex technical products into easy-to-understand guides that help the end-user understand how to use the products and services.

Vistula course

Technical writer, Information Developer, Documentation Specialist, UX Writer, Content Designer, Documentation Engineer, Content Writer, User Assistance Specialist, Information engineer, Information Architect - are the positions you can choose between after the course of “Technical Communication” at Vistula University.

The students are provided with a professional set of tools.

The course is realizied with the help of the partners:

Curriculum of the course

Table 1. Subjects

Subject	Hours
Technical Communication	20
Technical text	25
WEB technologies	10
Basic technologies of Technical Communication	25
Techniques and tools applied in technical documentation	50
Project, process and team management	20
ITCQF certification	20

Chapter 2. A profession of Technical Writer

A **technical writer** is a professional information communicator whose task is to transfer information between two or more parties, through any medium that best facilitates the transfer and comprehension of the information.

Responsibilities

- Technical content creation and content review
- Formatting and publishing
- Acquisition and organization of information
- Estimation of documentation quality
- Help for other teams - marketing, training, QA
- Automation of processes, implementation of tools
- Creation and maintainance of the information architecture
- Determination of the clearest and most logical way to present information

Specializations

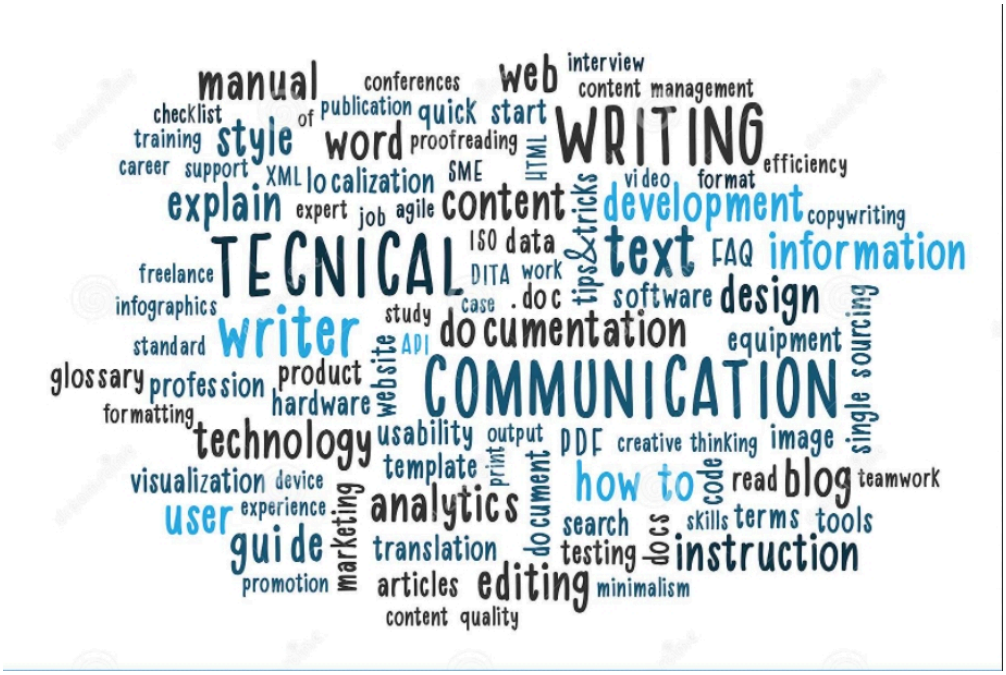
- Technical Writer
- Information Developer
- Documentation Specialist
- Information Architect
- UX Writer
- Content Designer

- Documentation Engineer
- Content Writer
- User Assistance Specialist
- Technical Translator

Courses

- Google Technical Writing Course (Free)
- Udemy Technical Writing Courses (Paid)
- Hashnode Technical Writing Bootcamp (Free)
- Game design documentation (Free) RUS
- School of Technical Writing (Free/Paid)
- 7 Day Technical Writing Email Course (Free)
- Online Technical Writing Classes (SkillShare)
- Technical Writing Tutorial (Free)

- Technical Writing: Quick Start Guides Online Class
- Wikiversity. Technical writing courses



Conferences

- soap!
- Meet Content
- Write the docs
- tecom Europe
- Evolution of TC 2021

Communities

- Techwriter.pl
- hashnode
- Society for Technical Communication
- Tech Writer's Tribe
- Season of Docs

Social networks

- Technical Writer Forum
- Projektowanie słowem: UX writing

- DITA
- Tworzenie dokumentacji
- Technical writers
- Software Technical Writer’s Group
- Microcopy & UX Writing
- UX writers Polska
- Technical writers united
- Lokalizacja gier i oprogramowania

Literature

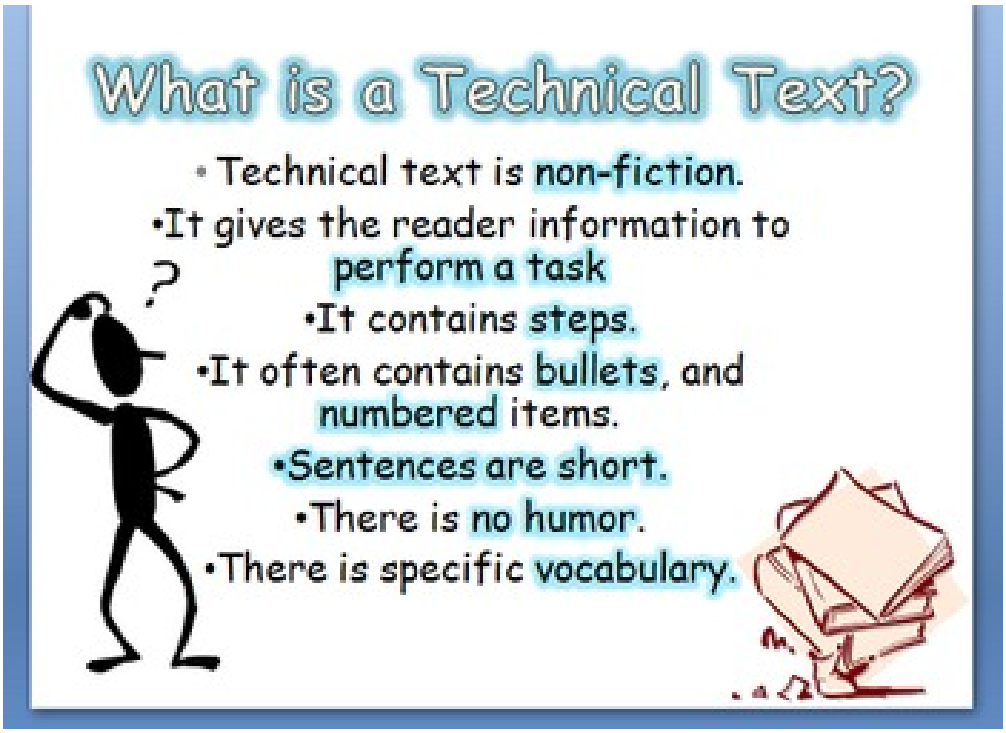
- The Insider's Guide to Technical Writing (Kindle Edition)
- Technical Writing For Dummies (Kindle Edition)
- Technical Writing Process: The simple, five-step guide that anyone can use to create technical documents such as user guides, manuals, and procedures
- A library of techwriter.pl
- Technical Writer Career Guide - Free Ebook

Sites

- Technical Writing for Beginners
- I’d rather be writing
- CyberText
- Heroic Technical Writers
- ClickHelp
- Technical writing is easy
- Tech Comm Tools
- Tech Writer koduje
- Scriptorium
- 3di

Chapter 3. Technical text

Technical text has certain characteristics which distinguish it from other types of texts. The purpose of technical texts is to educate the reader in a particular subject or skill through in-depth study and practice.



A typical technical text should be:

- **clear**
- **straightforward**
- **precise**
- **easily understood**
- **with denotative meanings**
- **detailed**
- **very structured**
- **skimmable**
- **with a problem-solving focus**

Types of documents

All the technical documents can be divided into three main groups: Product documentation, Training materials, and Process documentation.

Product documentation

This category of technical communication deliverables deals with user assistance related to products, including installation, usage, maintenance, customization, etc.

1. **User interface texts**

   UI strings, in-app copies, UX copies

   User interface text appears on UI surfaces. This text includes control labels and static text:

   - Control labels identify controls and are placed directly on or next to the controls.
   - Static text, which is so called because it is not part of an interactive control, provides users with detailed instructions or explanations so they can make informed decisions.

2. **Online help**

Online help is a topic-based document delivered together with software or operating systems. The document is designed to be displayed on the screen for immediate assistance when using IT products.

The purpose of online help is to provide the user with quick assistance directly from the application/module to which it refers.

Online help should consist of topics covering different aspects of a product itself and its possible usage patterns.

3. **User guides**

   A user guide (or user manual) is a document designed to provide assistance for a particular aspect of a product or solution.

   A user guide may serve different purposes and explain the product from different perspectives. Therefore, depending on usage context and identified audience needs, many different types of guides can be created, like:

   - **Administration guides** – for assisting users in system management
   - **How-to guides** – for guiding the users through a procedure to achieve a specific goal
   - **Implementation guides** – for explaining the procedure of enabling a solution
   - **Installation guides** – for guiding through product installation process
   - **Quick start guides** – for assisting in first-time use of a product
   - **Technical guides** – for covering low-level procedures

   Since a user guide is designed to be read as a whole, it typically has a book-like structure and is divided into regular chapters supported by images and diagrams.

4. **Mobile apps help**

   This type of documentation is used in the case of software running on mobile devices: smartphones, tablets, e-book readers, etc.

   The purpose of mobile application help is to provide users with instant assistance concerning the software they are using. This type of documentation makes it possible to adapt the way content is presented to the specific character of the user's device, mainly with respect to screen size and navigation possibilities.

   Since this type of documentation is presented on specific devices, the structure of the help should be adjusted to their limited field of view and navigation options. In the case of smaller applications, the documentation may be limited to only a couple of questions and answers or a single quick start topic.

5. **Knowledge base**

A knowledge base is a web-based system for gathering information on a specific subject, including frequently asked questions (FAQ), installation procedures, quick start articles, usage instructions, etc.

The main purpose of this type of documentation is to gather all available knowledge on a specific product in one place accessible for end users.

This type of documentation is very flexible in terms of structure, and can be implemented according to the specific needs of the intended audience. Depending on the needs, a knowledge base can be divided into different types of content, like FAQs, articles, forums, tutorials.

6. **Release notes**

   This type of documentation provides an overview of changes in a particular version of a software product. It does not explain features in such detail as other documentation types.

   The purpose of the release notes document is to communicate changes introduced in a specific version of the product to end users. The changes may include feature additions and removals, as well as bug fixes and known issues.

   The structure of release notes is typically designed individually in each organization depending on the requirements and the type of information to be conveyed. The most popular structure consists of a short introduction followed by a bulleted list of modifications and/or known issues, together with the description of their impact on the product that each change entails.

7. **API documentation**

   API documentation is a deliverable providing information on how to use and interact with an application programming interface (API). It is a concise guide to an API's endpoints, their expected parameters and their data types, as well as possible return values and ormats. API documentation also frequently includes end-to-end tutorials and detailed example scenarios.

   There are two key objectives of API documentation:

   - To explain the functionality and business value of an API, along with possible use cases
   - To provide a quick reference for those working with an API

   Since API documentation is primarily a source of reference information, it needs to be easily searchable and is therefore web-based.

   This type of documentation is typically produced from code, either completely automatically or semi-automatically.

8. **AR/VR**

   Virtual reality (VR) implies a complete immersion experience that shuts out the physical world. Using VR devices, users can be transported into a number of real-world and imagined environments.

Augmented reality (AR) adds digital elements to a live view, often by using the camera on a smartphone. It's simpler than VR and doesn't require such complex software and hardware in most cases.

9. **Chatbots**

   A chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. A chatbot is a type of software that can automate conversations and interact with people through messaging platforms. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse or pass the industry standard Turing test.

   Chatbots are used in dialog systems for various purposes including customer service, request routing, or information gathering. While some chatbot applications use extensive word-classification processes, natural language processors, and sophisticated AI, others simply scan for general keywords and generate responses using common phrases obtained from an associated library or database.

   Most chatbots are accessed on-line via website popups or through virtual assistants. They can be classified into usage categories that include: commerce (e-commerce via chat), education, entertainment, finance, health, news, and productivity.

10. **Infographics**

    Infographics (a clipped compound of "information" and "graphics") are graphic visual representations of information, data, or knowledge intended to present information quickly and clearly. They can improve cognition by utilizing graphics to enhance the human visual system's ability to see patterns and trends. Similar pursuits are information visualization, data visualization, statistical graphics, information design, or information architecture. Infographics have evolved in recent years to be for mass communication, and thus are designed with fewer assumptions about the readers' knowledge base than other types of visualizations.

Training materials

The technical communication deliverables included in this category are used for facilitating the learning process.

1. **Presentations**

   Presentations are training and/or informative materials displaying information in the form of a slide show.

They are used as reference when conducting training sessions or when presenting information on a particular subject. This type of materials helps the trainer or presenter maintain a fixed structure of the session and helps the attendees visualize, understand and memorize the information to be learned.

2. **Classroom trainings**

   Classroom training materials are created for teaching groups of users gathered in one room.

   They are designed to teach attendees how to use a product by explaining theoretical aspects and conducting practical exercises with the supervision of a trainer. This kind of training often offers live usage of a product, which provides possibilities of serving both education and marketing purposes.

3. **eLearnings**

   E-learning is a form of education via the Internet or from physical media. It makes it possible to complete a course, training session, or even university education without the need for physical presence in a classroom.

   The purpose of e-learning is to provide the attendees with training possibilities whenever classroom session is not possible or inconvenient to be organized. This type of training also helps achieve better cost-efficiency, since it does not require the organizers to involve trainers, to book a venue or to ship environments. Moreover, e-learning is designed to be the most accessible form of training – if the technological requirements are met, it can be completed at any time and any place chosen by the trainee.

4. **Webinars**

   A webinar is an Internet seminar conducted with the use of a webcast technology providing bi-directional communication between the trainer and trainees.

   Webinars are used for training purposes whenever a classroom session is not possible and e-learning limitations make it difficult to explain a subject in a sufficient manner. The interaction with a trainer possible in webinars provides the attendees with the opportunity to clarify any doubts they might have in reference to the subject of the training. This type of communication is also commonly used for demo purposes, as well as for business contacts.

   This type of training may be provided in the form of an online lecture with audio/video, often accompanied by a presentation.

Process documentation

Project documentation is a deliverable required to be created and approved before work on a project can be started. Examples of such documentation include project proposals, requirement specifications, feasibility studies, business cases, etc.

1. **Project documentation**

   Project documentation is a deliverable required to be created and approved before work on a project can be started. Examples of such documentation include project proposals, requirement specifications, feasibility studies, business cases, etc.

   The purpose of the document is to identify the scope of work required to complete a project, to explain why the project is necessary, and to propose ways of accomplishing it.

   Project documentation is similar to user guides and has a book-like structure. The main sections that should be present in the document include problem statement (motivation), objectives (what should be done), problem solution (how to do it), and project management (who should do it and when it should be done).
2. **Policies and procedures**

   This type of documentation refers to a set of rules, principles and guidelines created or implemented by an organization in order to achieve its goals.

   Policies and procedures are created in order to systematize the way an organization should be functioning.

   This type of documentation may follow the structure of a short user guide.
3. **Reports**

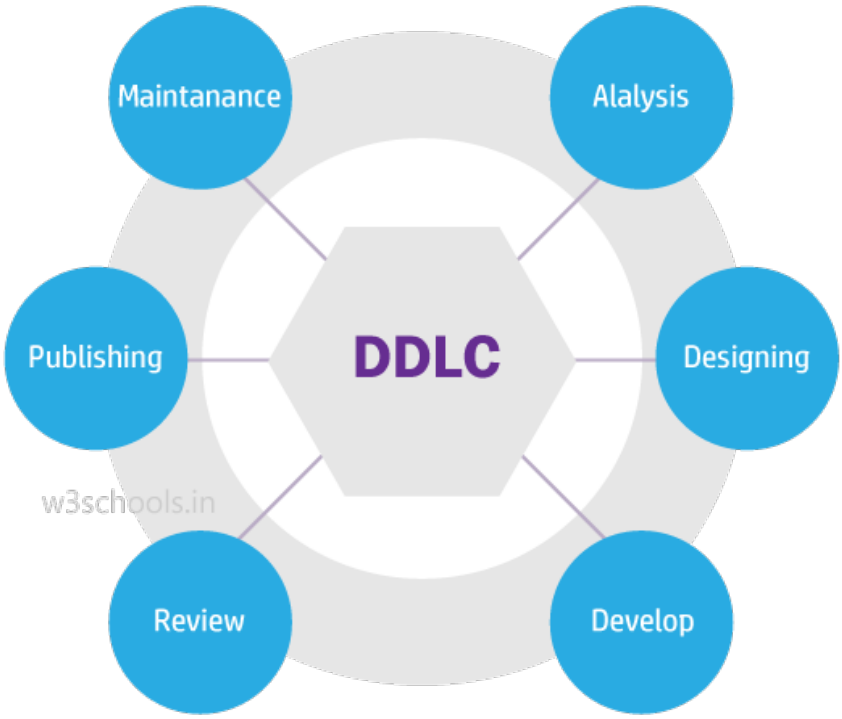
   A report is an analytical document written to provide data on a particular subject to a specific audience.

   Reports are most commonly requested by managers, business partners, existing and prospective customers, etc.

   The structure of this type of document depends primarily on the purpose and target audience of the report.

Document development life cycle

Document Development Life Cycle (DDLC) can be defined as the practice of developing a document that involves a systematic process that continues in cyclic order.



Requirement Analysis

In this stage, technical writers or content developers must collect specific information regarding the product from product requisite, SME, online help or seniors, and clients. It is because technical writers have to prepare the content as per the audience. It has an essential subsection: Audience analysis where professional writers and documentation experts have to explore who will use the product associated with the writing, user's need of building the product, as well as assess of skill along with the expertise of the target audience before writing the document.

Usually, technical writers assemble such information from Subject Matter Expert (SME). This saves the writer's time to understand the product. Since every company has SMEs for various departments, technical writers and content creators can approach them. Another potential target to get info regarding the product and its working mechanism is from software developers. Time, cost, and resource estimation are also done in this phase.

- Identify the audience (determine if the reader will be a technical user or a novice user)
- Identify the type of document to create
- Identify the cost (based on the number of screens in case of a Website guide)
- Identify the SMEs
- Identify various sources of related documentation

- Request for access to the relevant resources
- Identify the technical and functional reviewers

Designing Phase

In this designing stage, technical writers need to design the document or content by using proper layout, format, style and strategy. Professional writers use various authoring tools and bulk document processing applications such as RoboHelp, Madcap Flare, MS Office, oXygen. Also, capturing screen images and product design tools like Snagit are used. Camtasia is another application used for video capturing and editing. XML and DITA are also used for creating better formatting of documents. These are some tools and technologies technical writer needs to know.

- Gather related documentation
- Install the required tools
- Create a strategy to publish work on time
- Create the Table of Content (TOC)
- Create the Template and Style Guide

Developing the content

The content is written as per the product features, requirements, and understanding made in the first phase. Once the product is analyzed, the product is run, drafting is done as per format and template.

- Create content
- Create figures
- Create the index, appendix, glossary as requested

Editing / Proofreading

Next comes the editing phase, where the document is tested, as per the client/user's necessity, requirements, and product features. Technical writing editors, peer reviewers, or the content review expert (who might be the head of the professional writing department), will test and check the complete documentation. Here, technical writers verify the technical part, figures, grammar mistakes, and document format. Proofreading is a part of it where content experts check the entire draft to remove bugs from the documentation drafting.

- Review the document with the appointed SME
- Peer review
- Technical review
- Functional review

Publishing

Here, in this stage, technical writers bring out the document, release it with the product or as online help, and take a print of the entire document. The print is taken to check if the alignment is proper or not, and reading the hard copy also helps in getting a clear picture of the format and a few other documentation errors. Hyperlinks are added to the content in case of online release.

Maintenance

In this stage, if there is an update required to the document after the initial release, technical writers or content developers add updates, alter or modify the documentation. As the release of new products into the market, the online documentation or the document released earlier is updated.

- Constant deliverables maintenance
- Content reuse analysis
- Redundant data elimination
- Template revision
- New product feature inclusion

Important aspects of writing

The process of technical writing presupposes taking into account numerous aspects in order to get a text of high quality.

Work with information

There 5 ways to get information for a Technical Writer.

Project materials

- Specifications and requirements
- Estimates and analyses
- Product strategies
- Customer interviews
- Training materials
- Mockups
- Test plans

Product

- Previous product version
- Trials
- Mockups
- Test versions (test environment)

Existing documentation

- Previos version
- Similar product
- Similar type of documentation
- The same technology, but a different project

- Developer's documentation
- Knowledge base
- Documentation for implementation departments
- Documentation for support departments
- Standards

Competition

- Competition analysis
- Open documentation portals
- Libraries
- Knowledge bases
- Product community forums
- Open social media channels

Eksperts (Subject Matter Experts – SME)

- Specialists in their fields
- Person with broad subject knowledge
- Experienced practitioner

SME

A technical writing job supposes to communicate with members of other teams, for example, with developers, designers, SMEs, and the like; and all of them are just people. They can be busy, they can forget something or they even can dislike you because of some subjective reasons. So, how to handle unresponsive SMEs if you need to interview them?

When you set a date for a business meeting, keep in mind the following tips:

Build a strong relationship with your SME. The stronger your relationship is with your SME, the more likely you are to develop successful content.

Use examples.SMEs are NOT instructional designers or course creators. Presenting them with similar end-products of what you'd like your course to look like will give them a much better idea of the content you need them to provide.

Do your homework — prepare relevant questions, the answers to those questions should not be easily found elsewhere. You should use questions as a base for discussion, don't ask questions just one by one but discuss with them good and bad ideas, and suggest what you can improve.

Think about ways of communication that SME prefers. Maybe, it will be more suitable for them to communicate via phone, email, slack and so on.

Help SMEs with their tasks, it can be something like consulting about a presentation, proofreading a report for them, etc. You'll show that you're not just another team member and they will memorize you.

Style guides

A style guide is a **set of standards for writing and designing contents**. A style guide for technical writing defines the style that should be used in technical communication, such as in user manuals, online help, and procedural writing. A style guide helps you to write documentation in a clearer way, and to keep a consistent tone of voice and style.

- Google
- Microsoft
- Atlassian
- Apple
- Salesforce

Standards (STE)

Simplified Technical English(STE) was developed to help the readers of English-language documentation understand what they read, particularly when these readers are non-native English speakers.

ASD-STE100

Simplified Technical English STE writing rules

Wikipedia: Simplified Technical English

Customer personas

A customer persona (also known as a buyer persona) is a semi-fictional archetype that represents the key traits of a large segment of your audience, based on the data you've collected from user research and web analytics. It gives you insight into what your prospective customers are thinking and doing as they weigh potential options that address the problem they want to solve.

The 5 best guides for creating a customer persona

The example of the persona that was created for the purpose of this project.

Customer journey map

A customer journey map is a visual representation of the process a customer or prospect goes through to achieve a goal with your company. With the help of a customer journey map, you can get a sense of your customers' motivations — their needs and pain points.

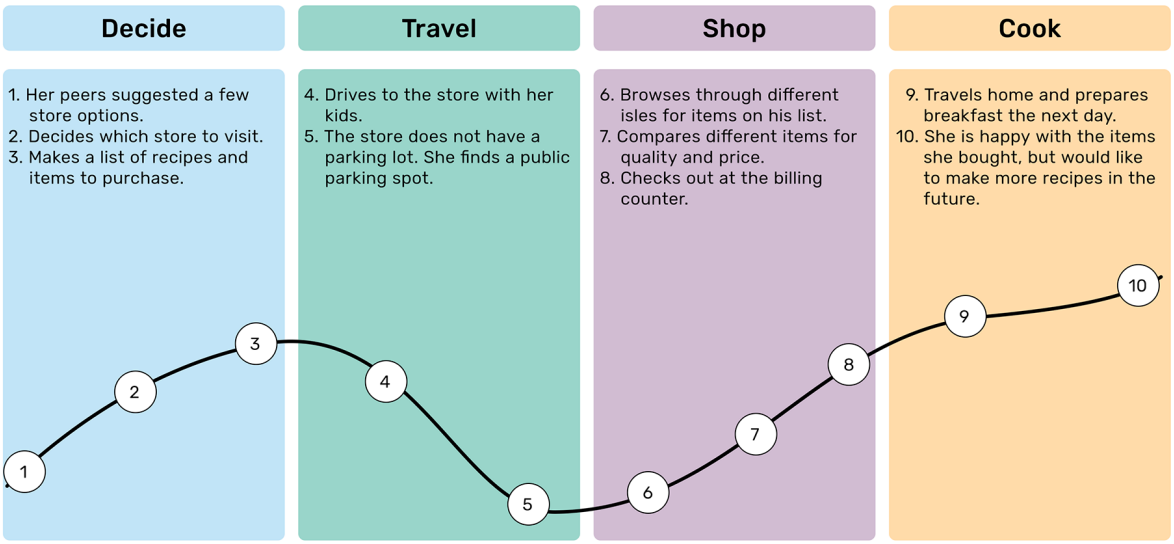
How to Create an Effective Customer Journey Map (Examples + Template)

Scenario

Leslie needs new breakfast recipe ideas and asks her peers for suggestions. She wants her kids to eat healthy.

Expectations

- Find variety of organic products.
- Prepare fresh and healthy breakfast for her kids.



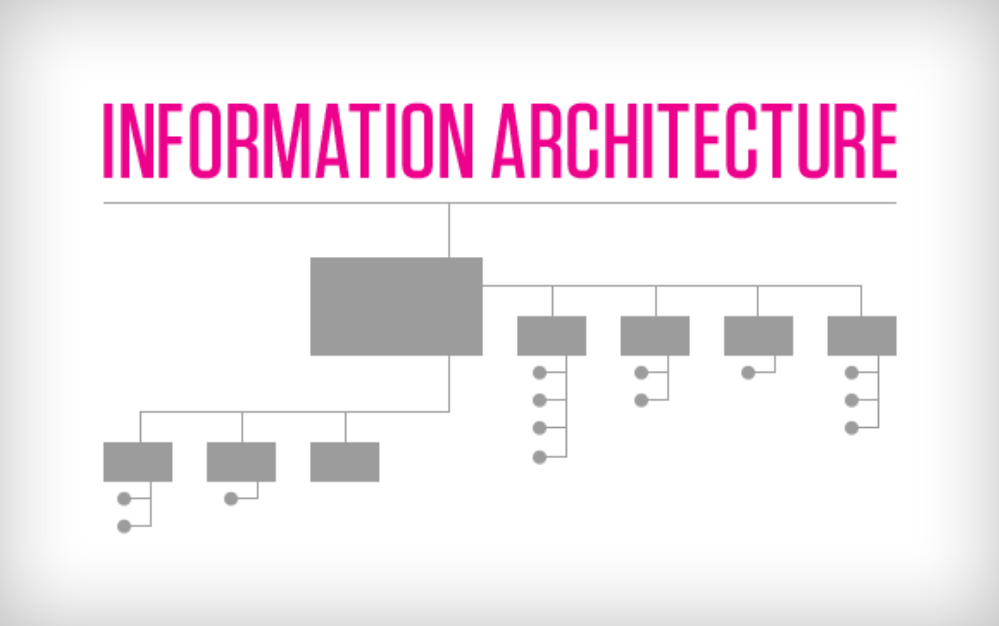
Content strategy

Content strategy is the ongoing process of translating business objectives and goals into a plan that uses content as a primary means of achieving those goals.

What is Content Strategy? (With Examples)

Information architecture

Information architecture is the creation of a structure for a website, application, or other project, that allows us to understand where we are as users, and where the information we want is in relation to our position. Information architecture results in the creation of site maps, hierarchies, categorizations, navigation, and metadata.



Complete Beginner's Guide to Information Architecture

Taxonomy

Taxonomy represents the foundation upon which information architecture stands, and all well-rounded developers should have at least a basic understanding of taxonomy to ensure that they can create organized, logical applications.

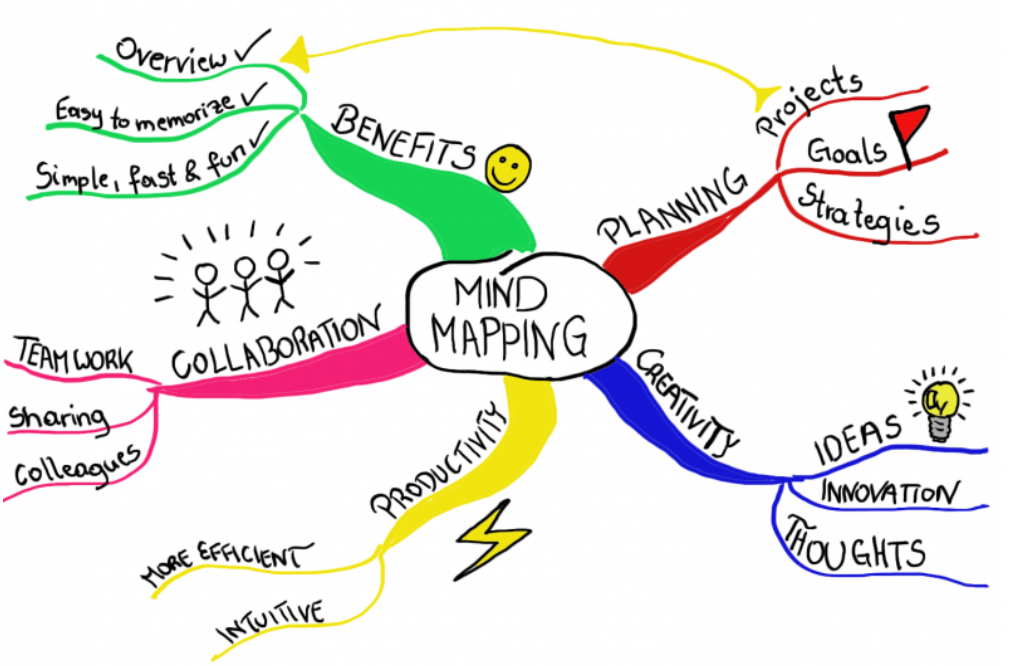
Understanding information taxonomy helps build better apps

Mind mapping

A Mind Map is an easy way to brainstorm thoughts organically without worrying about order and structure. It allows you to visually structure your ideas to help with analysis and recall.

A Mind Map is a diagram for representing tasks, words, concepts, or items linked to and arranged around a central concept or subject using a non-linear graphical layout that allows the user to build an intuitive framework around a central concept. A Mind Map can turn a long list of monotonous information into a colorful, memorable and highly organized diagram that works in line with your brain's natural way of doing things.

MindMapping


MIND MAPPING
BENEFITS
Overview
Easy to memorize
Simple, fast & fun
PLANNING
Projects
Goals
Strategies
COLLABORATION
TEAM WORK
Sharing
Colleagues
CREATIVITY
IDEAS
INNOVATION
THOUGHTS
PRODUCTIVITY
MORE EFFICIENT
INTUITIVE

Chapter 4. WEB technologies

According to Microsoft, web technologies include the following: Mark-up languages, such as **HTML and CSS**; Programming languages; Web server and server technologies; Databases, and Business applications.

HTML

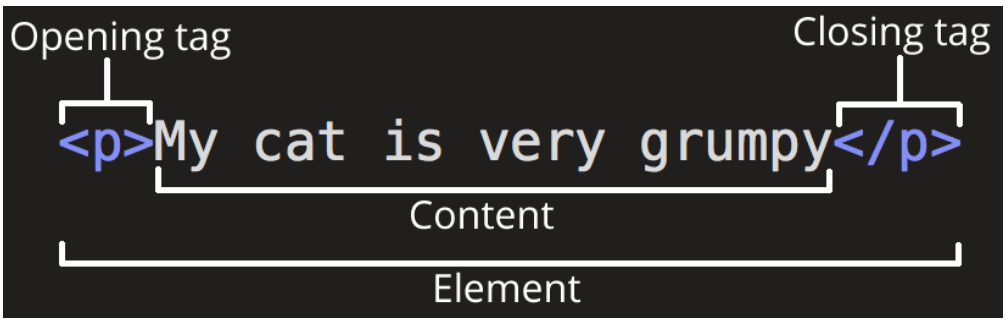
HTML (Hyper Text Markup Language) is the code that is used to structure a web page and its content.

HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.

The main parts of the **element** are as follows:

1. **The opening tag:** This consists of the name of the element (in this case, p), wrapped in opening and closing **angle brackets**. This states where the element begins or starts to take effect — in this case where the paragraph begins.
2. **The closing tag:** This is the same as the opening tag, except that it includes a *forward slash* before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.
3. **The content:** This is the content of the element, which in this case, is just text.
4. **The element:** The opening tag, the closing tag, and the content together comprise the element.



Elements can also have attributes that look like the following:

Attributes contain extra information about the element that you don't want to appear in the actual content. Here, `class` is the attribute *name* and `editor-note` is the attribute *value*. The `class` attribute allows you to give the element a non-unique identifier that can be used to target it (and any other elements with the same `class` value) with style information and other things.

An attribute should always have the following:

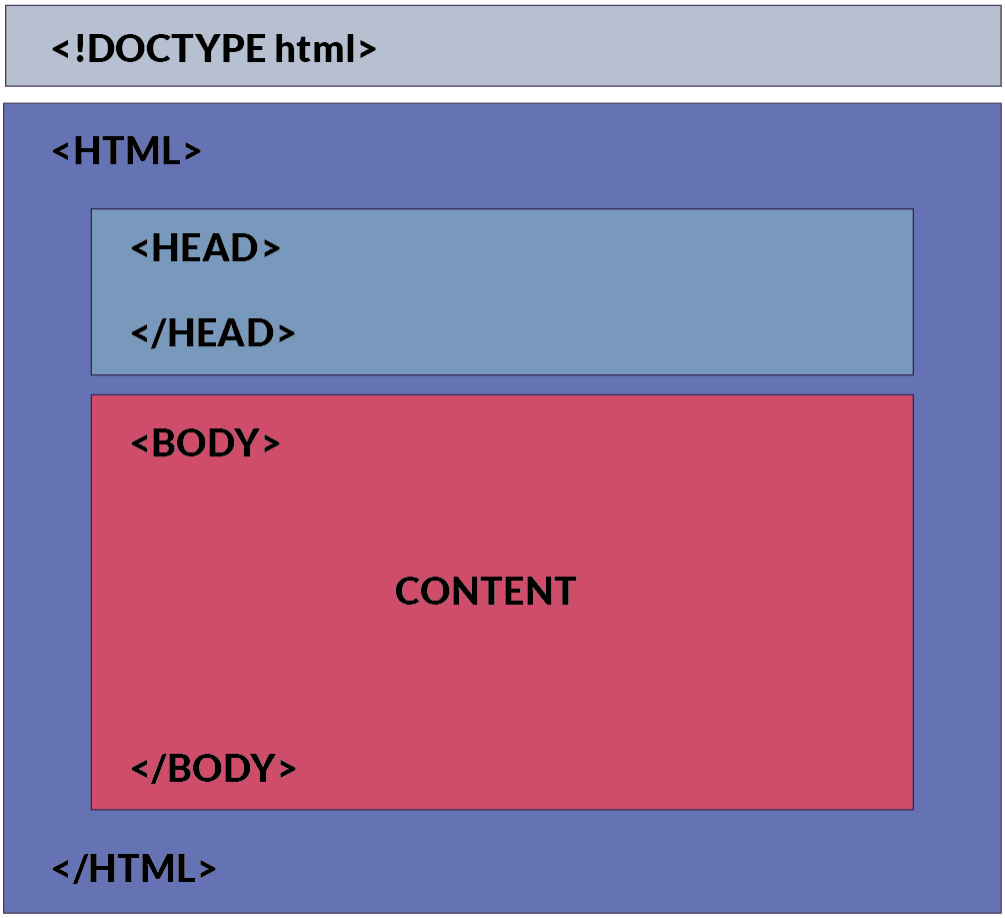
1. A space between it and the element name (or the previous attribute, if the element already has one or more attributes).
2. The attribute name followed by an equal sign.
3. The attribute value wrapped by opening and closing quotation marks.

Anatomy of an HTML document

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <title>My first page</title>

  </head>

  <body>

    <img src="images/Vistula_logo.png" alt="Vistula">

  </body>

</html>
```

Here, we have the following:

- `<!DOCTYPE html>` — doctype. It is a required preamble, The Document Type Declaration. If a declaration is not included, various browsers will revert to “quirks mode” for rendering.
- `<html></html>` — the `<html>` element. This element wraps all the content on the entire page and is sometimes known as the root element.
- `<head></head>` — the `<head>` element. This element acts as a container for all the stuff you want to include on the HTML page that *isn't* the content you are showing to your page's viewers.
- `<meta charset="utf-8">` — This element sets the character set your document should use to UTF-8 which includes most characters from the vast majority of written languages.
- `<title></title>` — the `<title>` element. This sets the title of your page, which is the title that appears in the browser tab the page is loaded in.
- `<body></body>` — the `<body>` element. This contains *all* the content that you want to show to web users when they visit your page, whether that's text, images, videos, games, playable audio tracks, or whatever else.



Basic body elements

1. **Headings**

   Heading elements allow you to specify that certain parts of your content are headings — or subheadings. HTML contains 6 heading levels, `<h1>`–`<h6>`.

   ```
   <h1>My main title</h1>
   <h2>My top level heading</h2>
   <h3>My subheading</h3>
   <h4>My sub-subheading</h4>
   <h5>My minor title</h5>
   <h6>My insignificant title</h6>
   ```

2. **Paragraphs**

   `<p>` elements are for containing paragraphs of text.

   ```
   <p>This is a single paragraph</p>
   ```

3. **Lists**

   The most common list types are ordered and unordered lists:

a. **Unordered lists** are for lists where the order of the items doesn't matter, such as a shopping list. These are wrapped in a `<ul>` element.

```
<ul>
    <li>Milk</li>
    <li>Cheese
        <ul>
            <li>Blue cheese</li>
            <li>Feta</li>
        </ul>
    </li>
</ul>
```

b. **Ordered lists** are for lists where the order of the items does matter, such as a recipe. These are wrapped in an `<ol>` element.

```
<ol>
  <li>Mix flour, baking powder, sugar, and salt.</li>
  <li>In another bowl, mix eggs, milk, and oil.</li>
  <li>Stir both mixtures together.</li>
  <li>Fill muffin tray 3/4 full.</li>
  <li>Bake for 20 minutes.</li>
</ol>
```

Each item inside the lists is put inside an `<li>` (list item) element.

4. **Links**

   To add a link, use a simple element — `<a>` — "a" being the short form for "anchor".

```
<a href="https://techwritersblog.com/">Technical writers blog</a>
```

5. **Images**

   The usage of the `<img>` element:

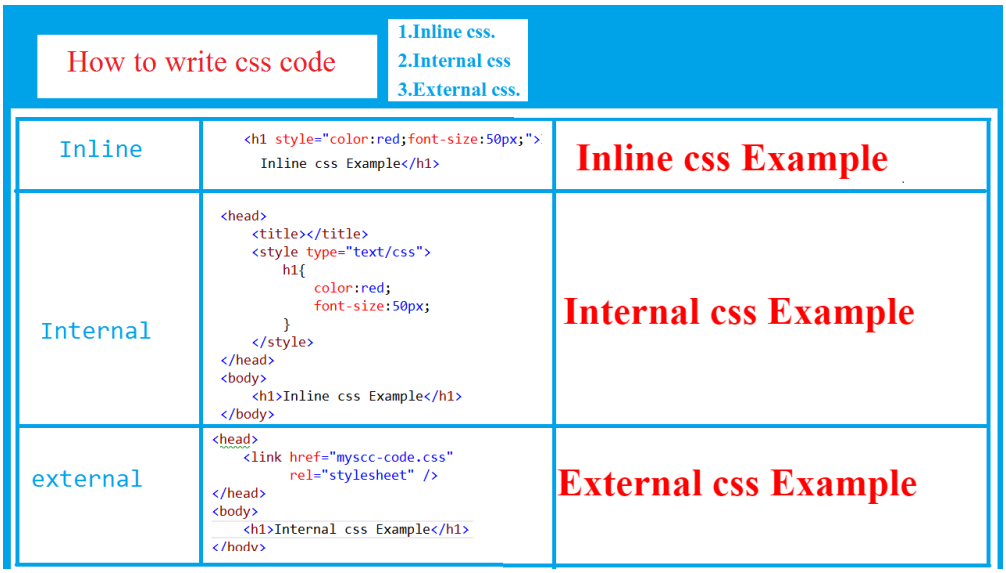
   - The `src` attribute is **required**, and contains the path to the image you want to embed.
   - The `alt` attribute holds a text description of the image, which isn't mandatory but is **incredibly useful** for accessibility — screen readers read this description out to their users so they know what the image means.

```
<img src="images/Vistula_logo.png" alt="Vistula">
```

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

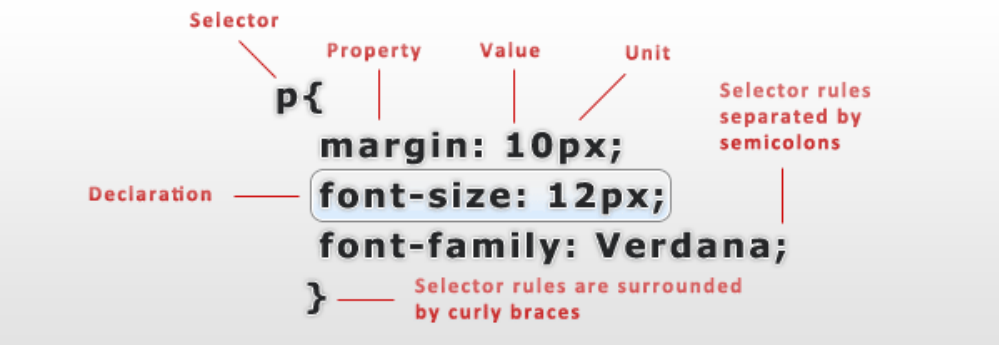


Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties.

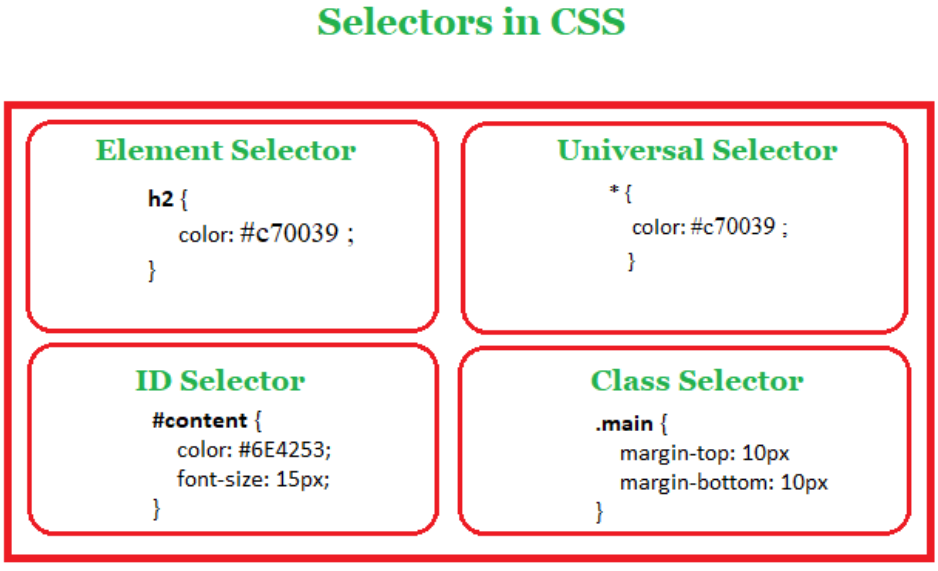
A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.



In CSS, selectors declare which part of the markup a style applies to by matching tags and attributes in the markup itself.

Selectors may apply to the following:

- all elements of a specific type, for example the second-level headers h2
- elements specified by attribute, in particular:
    1. id: an identifier unique within the document, identified with a hash prefix e.g. #id
    2. class: an identifier that can annotate multiple elements in a document, identified with a period prefix e.g. .classname
- elements depending on how they are placed relative to others in the document tree.

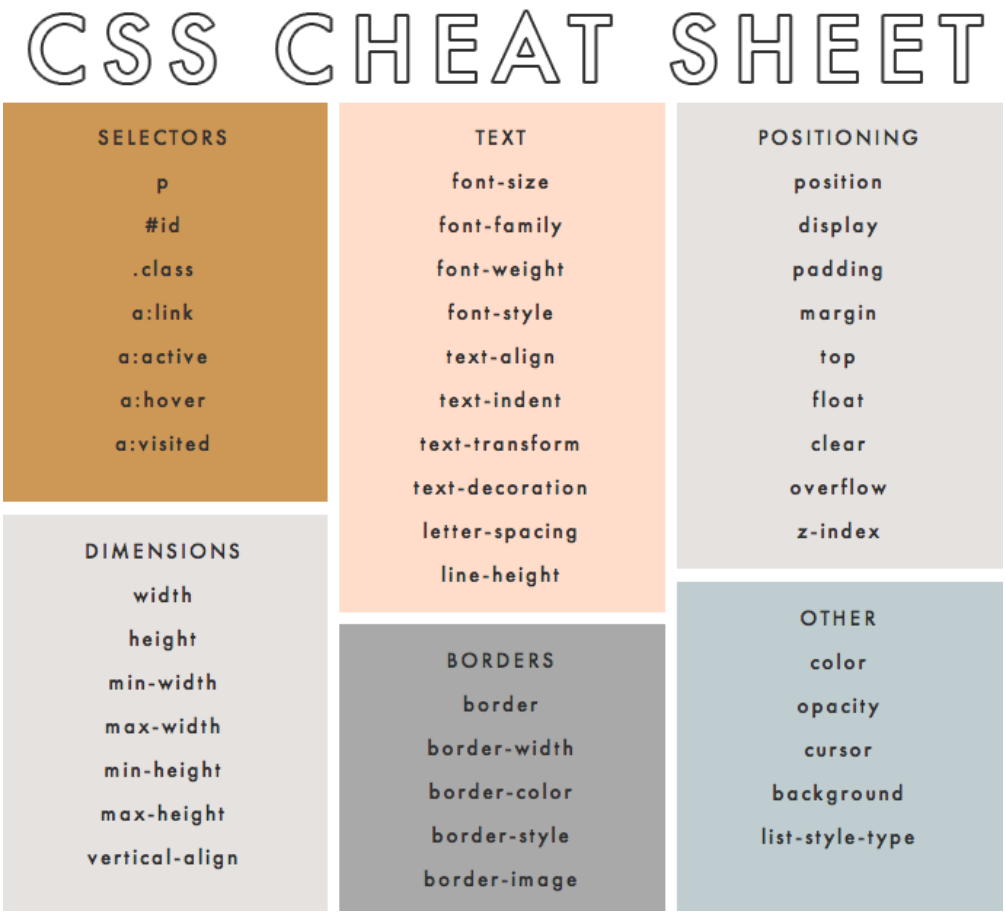


Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters, hyphens, and underscores. A class may apply to any number of instances of any elements. An ID may only be applied to a single element.

A declaration block consists of a list of declarations in braces. Each declaration itself consists of a property, a colon (:), and a value. If there are multiple declarations in a block, a semi-colon (;) must be inserted to separate each declaration. An optional semi-colon after the last (or single) declaration may be used.

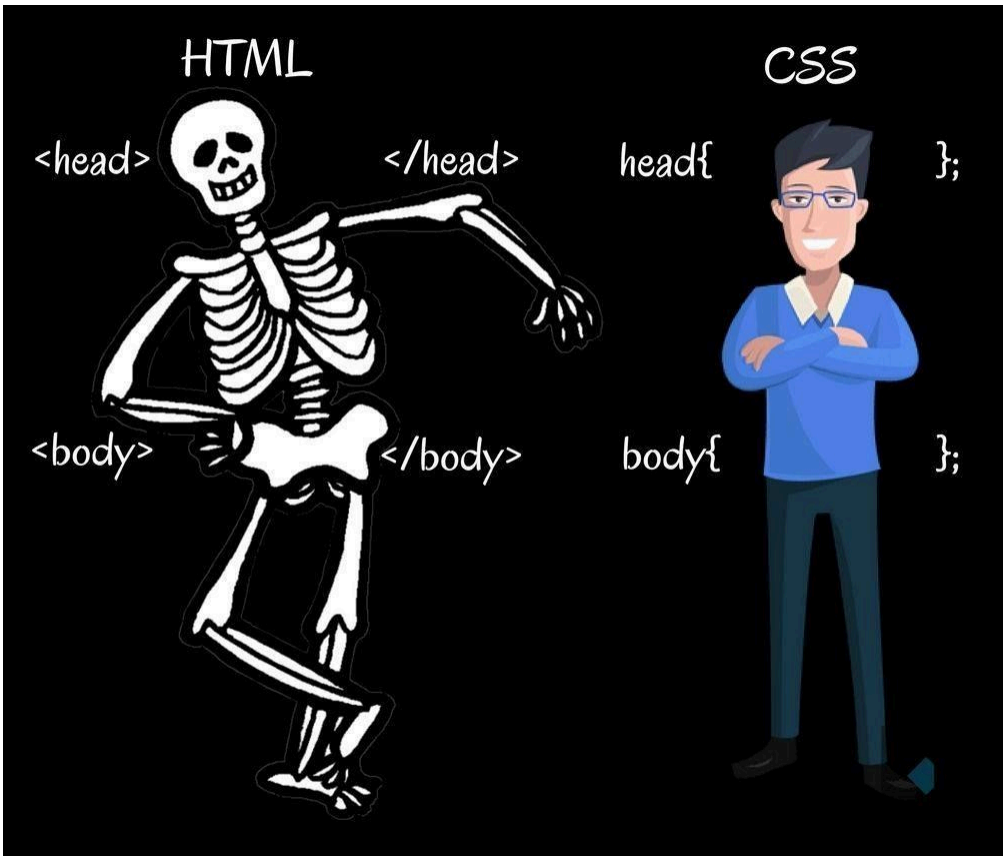
Properties are specified in the CSS standard. Each property has a set of possible values. Some properties can affect any type of element, and others apply only to particular groups of elements.

Values may be keywords, such as “center” or “inherit”, or numerical values, such as 200px (200 pixels). Color values can be specified with keywords (for example, “red”), hexadecimal values (for example, #FF0000, also abbreviated as #F00), RGB values on a 0 to 255 scale (for example, rgb(255, 0, 0)), RGBA values that specify both color and alpha transparency (for example, rgba(255, 0, 0, 0.8)), or HSL or HSLA values (for example, hsl(000, 100%, 50%), hsla(000, 100%, 50%, 80%)).



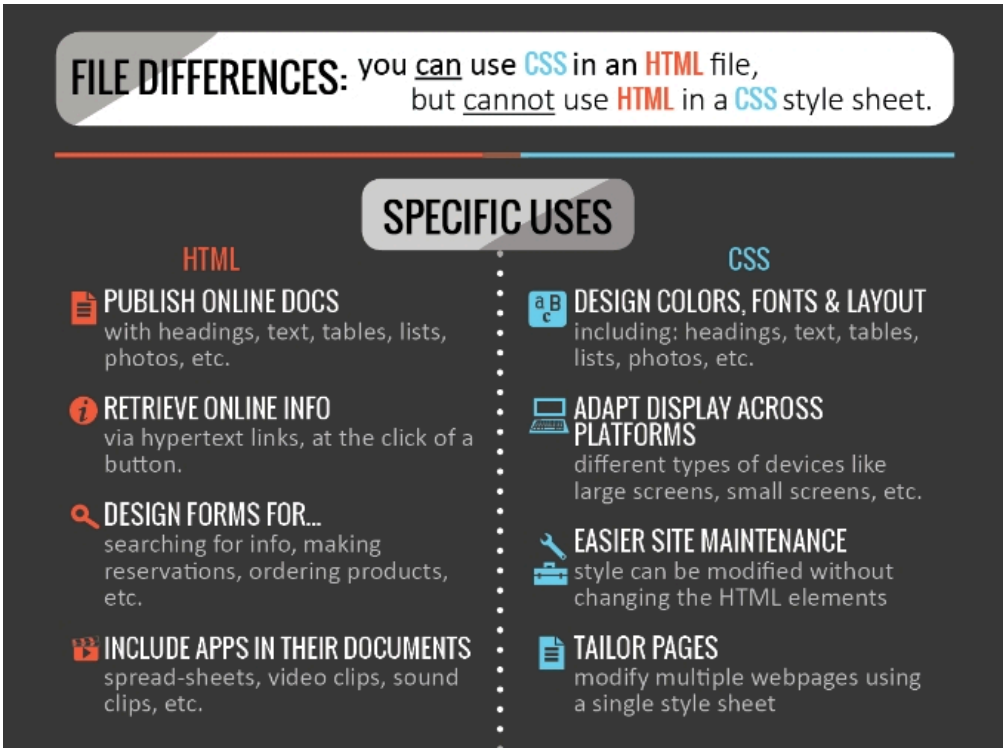
Difference between HTML and CSS

HTML is responsible for structure, CSS for style.

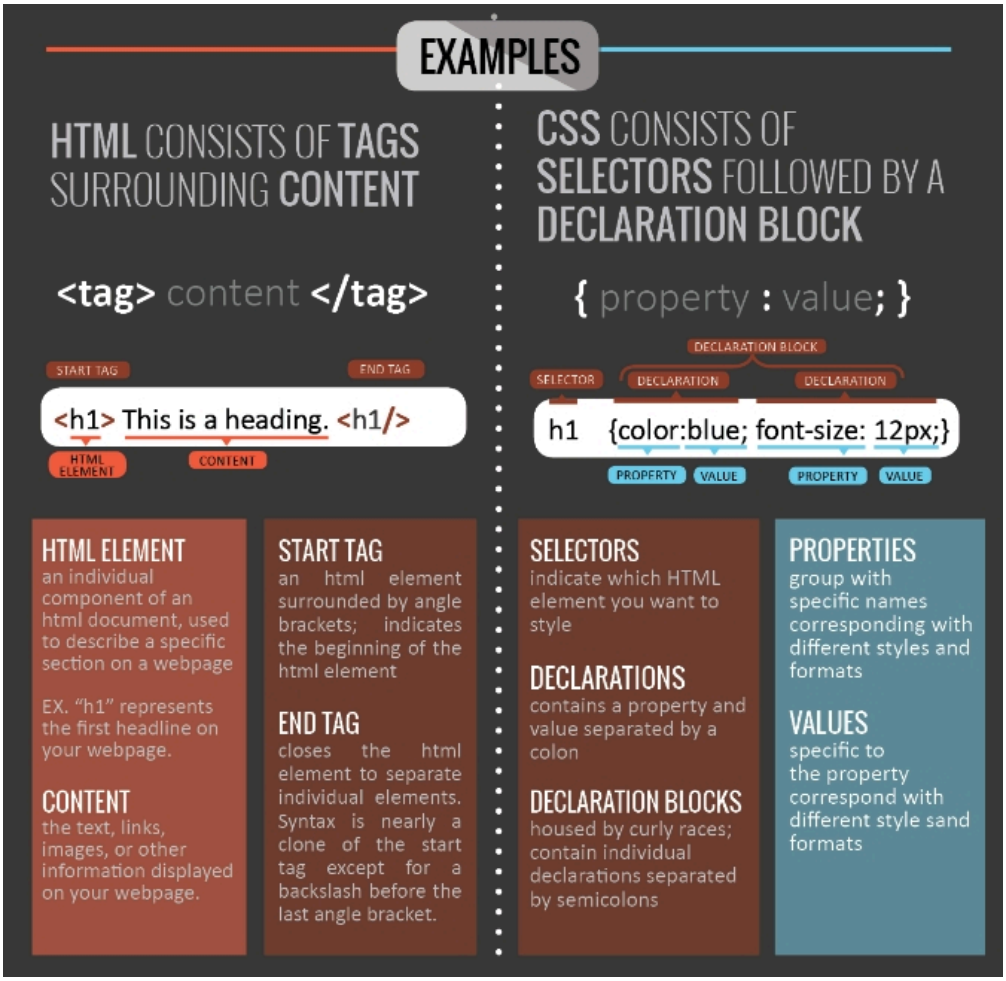


HTML stands for Hyper Text Markup Language. CSS - Cascading Style Sheets.

The purpose of HTML is to publish online docs. CSS - design.



HTML consists of tags. CSS consists of selectors.



But the general difference looks in the following way:


HTML
CSS

Chapter 5. Basic technologies

A technical writer should be communicative, have excellent language skills, perfect research and exploration skills, the ability to create versatile contents, and the knowledge of basic technologies that could be beneficial in developing effective technical writing.

In addition to solid research, language, writing, and revision skills, a technical writer may have skills in:

- Business analysis
- Computer scripting
- Content management
- Content design
- Illustration/graphic design
- Indexing
- Information architecture
- Information design
- Localization/technical translation
- Training
- E-learning
- User interfaces
- Video editing
- Website design/management
- Hypertext Markup Language (HTML)
- Usability testing
- Problem solving
- Version control
- Markup language
- Static site generators

Input and output formats

In an information system, input is the raw data that is processed to produce output.

Table 2. The most frequently used formats

Input formats	Output formats
Microsoft Word	PDF
LaTex	HTML5, WebHelp
HTML	Help (.chm - Windows)
XML (Adobe Frame maker, DITA, MadCap Flare)	Tooltip
JSON (API)	Wiki

Table 2. The most frequently used formats (continued)

Input formats	Output formats
Doxygen	Excel
Javadoc, Pydoc	Graphical tools
Markdown	CAT tools

Input formats

1. **Microsoft Word**

   Microsoft Word or MS Word (often called Word) is a graphical word processing program that users can type with. It is made by the computer company Microsoft. Its purpose is to allow users to type and save documents.

   Similar to other word processors, it has helpful tools to make documents.

   - Spelling and grammar checker, word count (this also counts letters and lines)
   - Speech recognitionInserts pictures in documents
   - Choice of typefaces
   - Special codesWeb pages, graphs, etc.
   - Tables
   - Displays synonyms of words and can read out the text
   - Prints in different ways

   Microsoft Word is a part of Microsoft Office, but can also be bought separately.

2. **LaTex**

   LaTeX is a software system for document preparation. When writing, the writer uses plain text as opposed to the formatted text found in “What You See Is What You Get” word processors like Microsoft Word, LibreOffice Writer and Apple Pages. The writer uses markup tagging conventions to define the general structure of a document (such as article, book, and letter), to stylise text throughout a document (such as bold and italics), and to add citations and cross-references. A TeX distribution such as TeX Live or MiKTeX is used to produce an output file (such as PDF or DVI) suitable for printing or digital distribution.

   LaTeX is widely used in academia for the communication and publication of scientific documents in many fields, including mathematics, statistics, computer science, engineering, physics, economics, linguistics, quantitative psychology, philosophy, and political science.

3. **HTML**

HTML stands for “Hypertext Markup Language.” HTML is the language used to create webpages. “Hypertext” refers to the hyperlinks that an HTML page may contain. “Markup language” refers to the way tags are used to define the page layout and elements within the page.

4. **XML (Adobe Frame maker, DITA, MadCap Flare)**

   Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

   The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

5. **JSON (API)**

   JSON (JavaScript Object Notation) is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays (or other serializable values). It is a very common data format, with a diverse range of applications, one example being web applications that communicate with a server.

   JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data.

6. **Doxygen, Javadoc, Pydoc**

   Doxygen is a documentation generator and static analysis tool for software source trees. When used as a documentation generator, Doxygen extracts information from specially-formatted comments within the code. When used for analysis, Doxygen uses its parse tree to generate diagrams and charts of the code structure. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code.

   Javadoc is a documentation generator created by Sun Microsystems for the Java language (now owned by Oracle Corporation) for generating API documentation in HTML format from Java source code. The HTML format is used for adding the convenience of being able to hyperlink related documents together.

   Pydoc is the standard documentation module for the programming language Python. Similar to the functionality of Perldoc within Perl and Javadoc within Java, Pydoc allows Python programmers to access Python’s documentation help files, generate text and HTML pages with documentation specifics, and find the appropriate module for a particular job.

7. **Markdown**

Markdown is a lightweight markup language for creating formatted text using a plain-text editor. John Gruber and Aaron Swartz created Markdown in 2004 as a markup language that is appealing to human readers in its source code form. Markdown is widely used in blogging, instant messaging, online forums, collaborative software, documentation pages, and readme files.

Output formats

1. **PDF**

   Portable Document Format (PDF), standardized as ISO 32000, is a file format developed by Adobe in 1993 to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems. Based on the PostScript language, each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, vector graphics, raster images and other information needed to display it.

   PDF files may contain a variety of content besides flat text and graphics including logical structuring elements, interactive elements such as annotations and form-fields, layers, rich media (including video content), and three-dimensional objects using U3D or PRC, and various other data formats. The PDF specification also provides for encryption and digital signatures, file attachments, and metadata to enable workflows requiring these features.
2. **HTML5, WebHelp**

   Webhelp is a chunked HTML output format in the DocBook xslt stylesheets that was introduced in version 1.76.1. The documentation for web help also provides an example of web help and is part of the DocBook xsl distribution.
3. **Help (.chm - Windows)**

   Microsoft Compiled HTML Help is a Microsoft proprietary online help format, consisting of a collection of HTML pages, an index and other navigation tools. The files are compressed and deployed in a binary format with the extension.CHM, for Compiled HTML. The format is often used for software documentation.
4. **Tooltip**

   The tooltip, also known as infotip or hint, is a common graphical user interface element in which, when hovering over a screen element or component, a text box displays information about that element (such as a description of a button's function, or what an abbreviation stands for). The tooltip is displayed continuously as long as the user hovers over the element.

   On desktop, it is used in conjunction with a cursor, usually a pointer, whereby the tooltip appears when a user hovers the pointer over an item without clicking it.
5. **Wiki**

Simple editing is one of the major benefits of using a wiki. Users can edit pages without knowing HTML, and still use many formatting features of HTML. Most wikis define a set of formatting rules to convert plain text into HTML. Some wikis also allow some HTML “tags” within a page. (Some wikis use raw HTML instead of special formatting rules.)

6. **Excel**

   Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications (VBA). Excel forms part of the Microsoft Office suite of software.

   Microsoft Excel has the basic features of all spreadsheets, using a grid of cells arranged in numbered rows and letter-named columns to organize data manipulations like arithmetic operations. It has a battery of supplied functions to answer statistical, engineering, and financial needs. In addition, it can display data as line graphs, histograms and charts, and with a very limited three-dimensional graphical display. It allows sectioning of data to view its dependencies on various factors for different perspectives (using pivot tables and the scenario manager). A PivotTable is a powerful tool that can save time when it comes to data analysis.

7. **Graphical tools**

   Graphical tools can provide comprehensive and easily understandable ways to present results of statistical analyses, particularly when a large amount of data is involved.

8. **CAT tools**

   The “CAT” in CAT tool stands for “Computer Aided Translation” or “Computer Assisted Translation” but, it doesn’t mean that a computer is actually completing the translation for you. CAT tools are different than “machine translation” – they assist a human translator in doing their work more quickly and in managing their translation projects. CAT tools typically contain a translation memory, which stores previous source and target translations for easy reference while working. Term bases are also an integral part of translation tools, giving translators the ability to develop their own bilingual glossaries in their subject areas.

Markdown

Markdown is a lightweight markup language for creating formatted text using a plain-text editor. Markdown is widely used in blogging, instant messaging, online forums, collaborative software, documentation pages, and readme files.

Headers

To create a header, preface the phrase with a hash mark (#). You place the same number of hash marks as the size of the header you want. For example, for a header one, you'd use one hash mark, while for a header three, you'd use three.

```
## ex. 2 Customer

### ex. 3 Project manager

#### ex. 4 Product owner

##### ex. 5 Developer

###### ex. 6 Technical writer
```

Paragraphs

To create paragraphs, use a blank line to separate one or more lines of text. You should not indent paragraphs with spaces or tabs.

```
**ex. Technical writing is not just about understanding technical information and recording it in a document.**


**Technical writing takes high-level information and processes it into digestible content for a specific audience.**
```

Line Breaks

To create a line break , end a line with two or more spaces, and then type return.

```
**ex. A technical writer is a professional writer that communications complex information.**

*So what exactly does a technical writer do?*

**They create technical documentation that includes things like instruction manuals, user manuals, journal articles,

 quick reference guides, and white papers.**
```

Emphasis

- **Bold**

  To bold text, add two asterisks (*) before and after a word or phrase. To bold the middle of a word for emphasis, add two asterisks without spaces around the letters.

```
ex. **Attention to detail with a creative eye.**
```

- **Italic**

  To italicize text, add one asterisk before and after a word or phrase. To italicize the middle of a word for emphasis, add one asterisk without spaces around the letters.

```
ex. *Passionate about learning and developing.*
```

Blockquotes

To create a blockquote, add a > in front of a paragraph.

```
ex. Mark Zuckerberg believes:

> Some people dream of success... while others wake up and work hard at it.
```

Lists

- **Numbered Lists**

  To create an ordered list, add line items with numbers followed by periods. The numbers don't have to be in numerical order, but the list should start with the number one.

```
1. Jira

2. Confluence

3. Brackets

4. Visual Studio Code

5. MadCap Flare

6. Snagit

7. Camtasia

8. MemoQ

9. Oxygen

10. GitHub Desktop
```

- **Bullet Lists**

  To create an unordered list, add asterisks (*) in front of line items.

```
* Technical communication

* Web technologies
```

```
    * HTML
    * CSS
* Technical texts
```

Images

To add an image, add an exclamation mark (!), followed by alt text in brackets, and the path or URL to the image asset in parentheses. You can optionally add a title after the URL in the parentheses.

```
![Technical
 communication](https://www.vistula.edu.pl/wp-content/uploads/2017/07/studia_podyplomowe_komunikacja_techniczna-565x278
.jpg)
```

Links

- **Linking Images**

  To add a link to an image, enclose the Markdown for the image in brackets, and then add the link in parentheses.

  ```
  ![Technical writer](assets/images/writer.jpg)
  ```

- **Linking URL**

  To create a link, enclose the link text in brackets and then follow it immediately with the URL in parentheses.

  ```
  [Visit GitHub!](www.github.com)
  ```

- **Linking to another file**

  ```
  [Additional content](Visual Studio Code.md)
  ```

Tables

To add a table, use three or more hyphens (---) to create each column's header, and use pipes (|) to separate each column. You can optionally add pipes on either end of the table.

```
| header 1 | header 2 |
| -------- | -------- |
| course 1 | course 2 |
| lesson 1 | lesson 2 |
```

Code

To indicate a span of code, wrap it with backtick quotes (`). Unlike a pre-formatted code block, a code span indicates code within a normal paragraph.

ex. Specify

```
`x=7`
```

when x equals seven.

To include a literal backtick character within a code span, you can use multiple backticks as the opening and closing delimiters.

```
```
ex. x=7;
    y=17;
    z=x+y;
```
```

Graphical elements

Using graphics is an essential part of creating technical documentation. They help you convey complex information in a more understandable and clear way for the readers.

Raster images

Raster images, also known as bitmaps, are comprised of individual pixels of color. Each color pixel contributes to the overall image.

Raster images might be compared to pointillist paintings, which are composed with a series of individually-colored dots of paint. Each paint dot in a pointillist painting might represent a single pixel in a raster image. When viewed as an individual dot, it's just a color; but when viewed as a whole, the colored dots make up a vivid and detailed painting. The pixels in a raster image work in the same manner, which provides for rich details and pixel-by-pixel editing.

Raster images are capable of rendering complex, multi-colored visuals, including soft color gradients. Digital cameras create raster images, and all the photographs you see in print and online are raster images.

There are different types of raster files: JPG, GIF, and PNG are examples, and each file type has its own nuances.

Raster images are ideal for photo editing and creating digital paintings in programs such as Photoshop and GIMP , and they can be compressed for storage and web optimized images.

How you can use a given raster image depends on its size and quality. Quality is often dictated by how many pixels are contained in an inch, expressed as pixels-per-inch or ppi; as well as the overall dimensions of the image, also expressed as pixels (for example, 5,000 pixels wide by 2,500 pixels high).

- **.jpeg**

  JPEG (ang. Joint Photographic Experts Group) – algorytm stratnej kompresji grafiki rastrowej, wykorzystany w formacie plików graficznych o tej samej nazwie. Motywacją do powstania tego standardu było ujednolicenie algorytmów kompresji obrazów monochromatycznych i kolorowych.

  Format plików JPEG/JFIF obok formatów GIF i PNG jest najczęściej stosowanym formatem grafiki na stronach WWW.

- **.gif**

  The Graphics Interchange Format is a bitmap image format. It has come into widespread usage on the World Wide Web due to its wide support and portability between applications and operating systems.

  The format supports up to 8 bits per pixel for each image, allowing a single image to reference its own palette of up to 256 different colors chosen from the 24-bit RGB color space. It also supports animations and allows a separate palette of up to 256 colors for each frame. These palette limitations make GIF less suitable for reproducing color photographs and other images with color gradients, but well-suited for simpler images such as graphics or logos with solid areas of color.

- **.png**

  Portable Network Graphics is a raster-graphics file format that supports lossless data compression. PNG was developed as an improved, non-patented replacement for Graphics Interchange Format (GIF).

  PNG supports palette-based images (with palettes of 24-bit RGB or 32-bit RGBA colors), grayscale images (with or without alpha channel for transparency), and full-color non-palette-based RGB or RGBA images. The PNG working group designed the format for transferring images on the Internet, not for professional-quality print graphics; therefore non-RGB color spaces such as CMYK are not supported. A PNG file contains a single image in an extensible structure of chunks, encoding the basic pixels and other information such as textual comments and integrity checks documented in RFC 2083.

Vector images

Unlike raster graphics, which are comprised of colored pixels arranged to display an image, vector graphics are made up of paths, each with a mathematical formula (vector) that tells the path how it is shaped and what color it is bordered with or filled by.

Since mathematical formulas dictate how the image is rendered, vector images retain their appearance regardless of size. They can be scaled infinitely. Vector images can be created and edited in programs such as Illustrator, CorelDraw, and InkScape (don't worry, these visual editors do the math for you).

Though vectors can be used to imitate photographs, they're best-suited for designs that use simple, solid colors. Vector images are comprised of shapes, and each shape has its own color; thus, vectors cannot achieve the color gradients, shadows, and shading that raster images can (it is possible to mimic them, but it requires rasterizing part of the image – which means it would not be a true vector). True vector graphics are comprised of line art, sometimes called wireframes, that are filled with color.

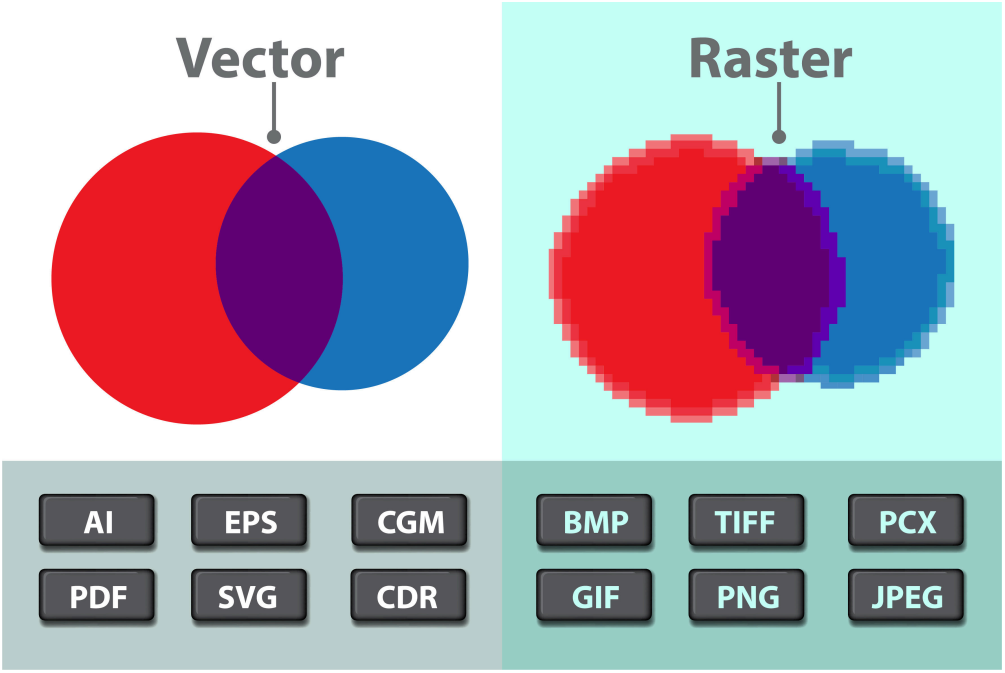
Because vectors can be infinitely scaled without loss of quality, they're excellent for logos, illustrations, engravings, etchings, product artwork, signage, and embroidery. Vectors should not be used for digital paintings or photo editing; however, they're perfect for projects such as printing stickers that do not include photos.

It's important to note that, with the exception of the SVG format, vectors must be rasterized before they can be used on the web.

- **.svg**

  SVGs are vector graphics. Rather than defining the color of each pixel like you would in a bitmap (JPEG, PNG, GIF, BMP etc.), vector graphics define lines and shapes, e.g. draw a black line from co-ordinate 0,0 to 100,100. This has a number of advantages: vectors are easy to modify, generally

require smaller files and are scalable to any dimension without losing quality — which makes them ideal for responsive web design. Bitmaps remain the best choice for photographs or very complex images (note that SVGs can include embedded bitmaps).

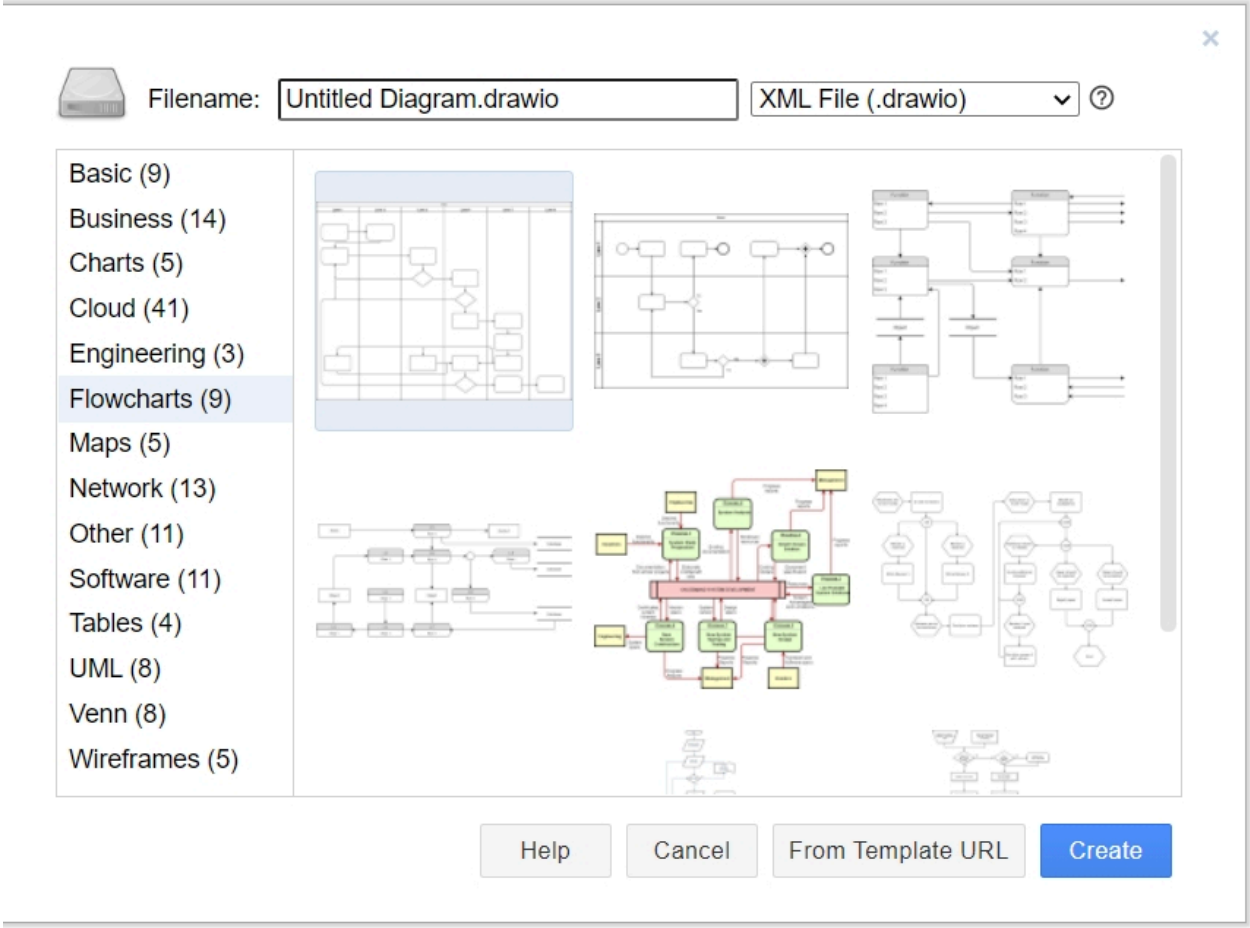


draw.io (app.diagrams.net)

The draw.io platform is a free-to-use online diagram app and editor. With this software package, you can create high-quality designs, custom flow charts, complex network diagrams, and Unified Modelling Language (UML) system layouts.

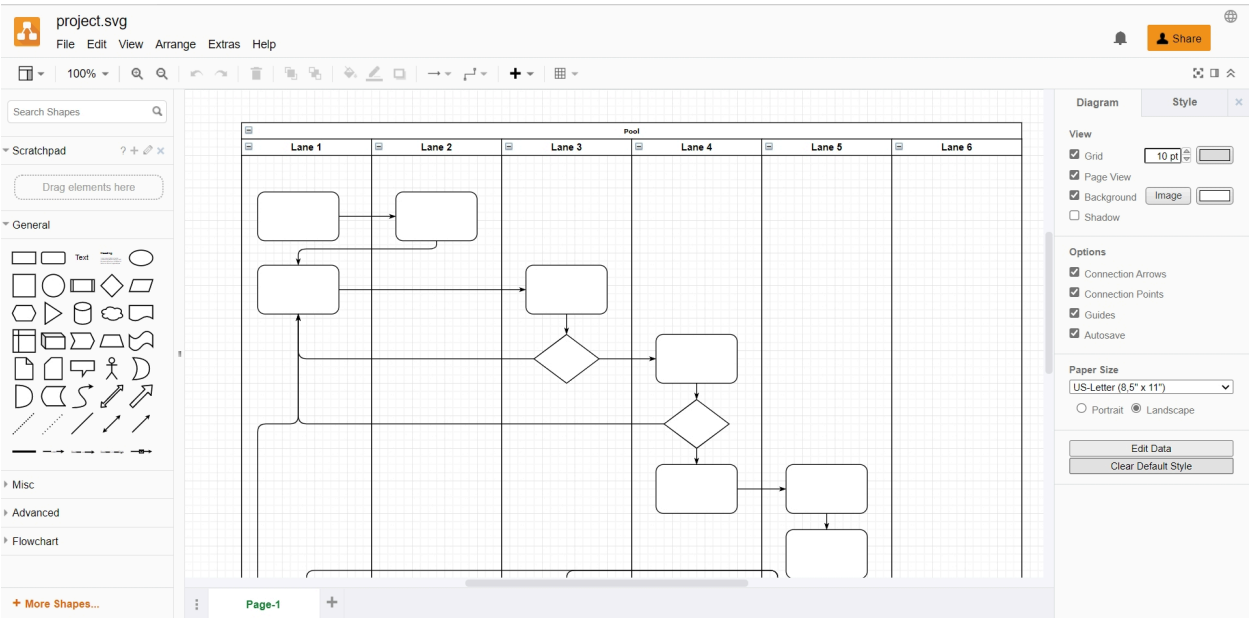
draw.io offers a beginner-friendly vector graphics processing environment. The big advantage of vector graphics over traditional image processing is that you won't lose any quality if you re-open and edit a vector file.

Vector graphics are useful in all sorts of different professional settings, from creating building blueprints to designing project workflows to share with your team. The draw.io package comes with dozens of different pre-made layouts that you can pick from when working on a new design. Not having to start your projects from scratch will make your design process more efficient.



The draw.io interface is nicely laid out with helpful shortcuts that will help you quickly locate popular shapes, functions, and settings. By default, on the left-hand side of the draw.io edit screen, you'll see a panel of useful shapes. One nice feature that speeds up the editing process is that when you put your cursor over a shape (but don't click it), a larger version will appear on your screen, letting you decide whether the shape is suitable for your project.

Over on the right-hand side of the screen, you'll see a panel populated by contextual menus. At first, in this space, you'll see generic options such as changing your diagram size or adding a grid. However, when you select an object, the content changes. For example, if some text is highlighted, you'll see font options, and selecting a shape causes a menu with color options to appear.



GitHub and GitHub Pages

GitHub can be considered as the most popular collaboration platform that is used in developing software.Therefore, it is quite important for anyone who is involved in the process of developing software to learn the basics.

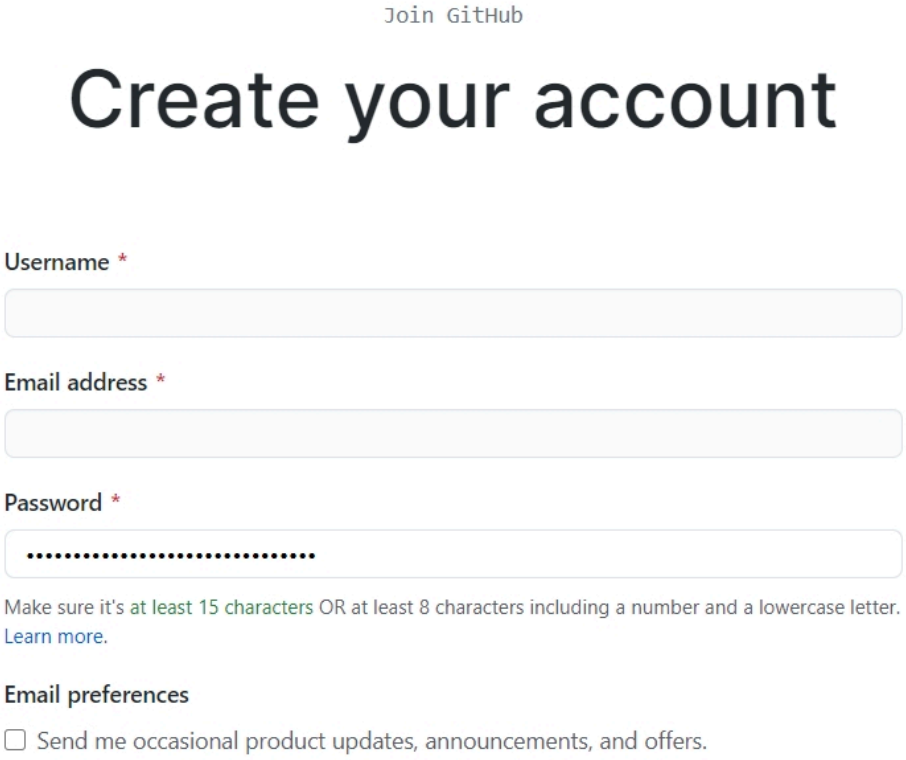
GitHub and GIT

GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project.**Git** is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

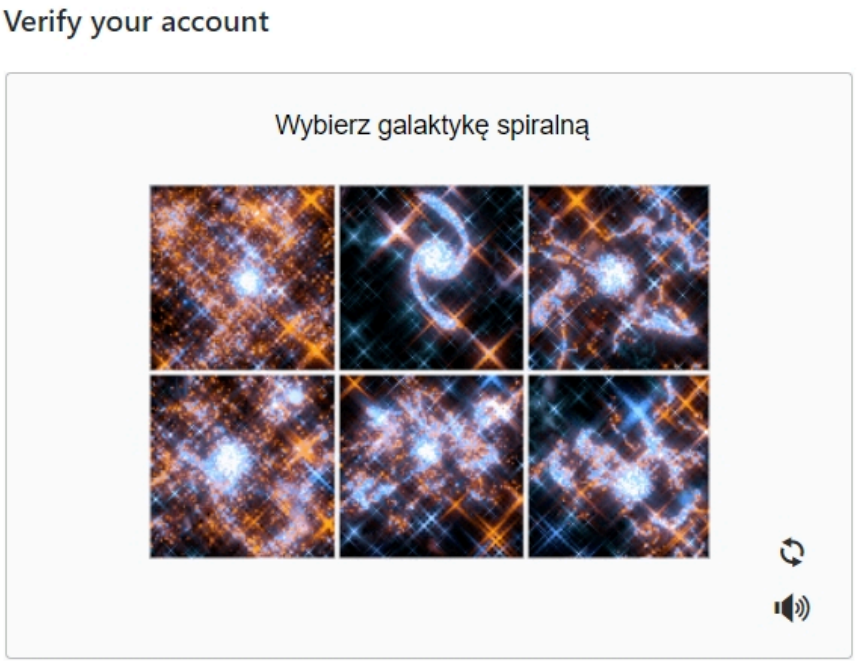
How to create an account on GitHub

1. Go to GitHub in a web browser.
2. Enter your personal details. In addition to creating a **username** and entering an **email address**, you'll also have to create a **password**.

   Your password must be at least 15 characters in length or at least 8 characters with at least one number and lowercase letter.



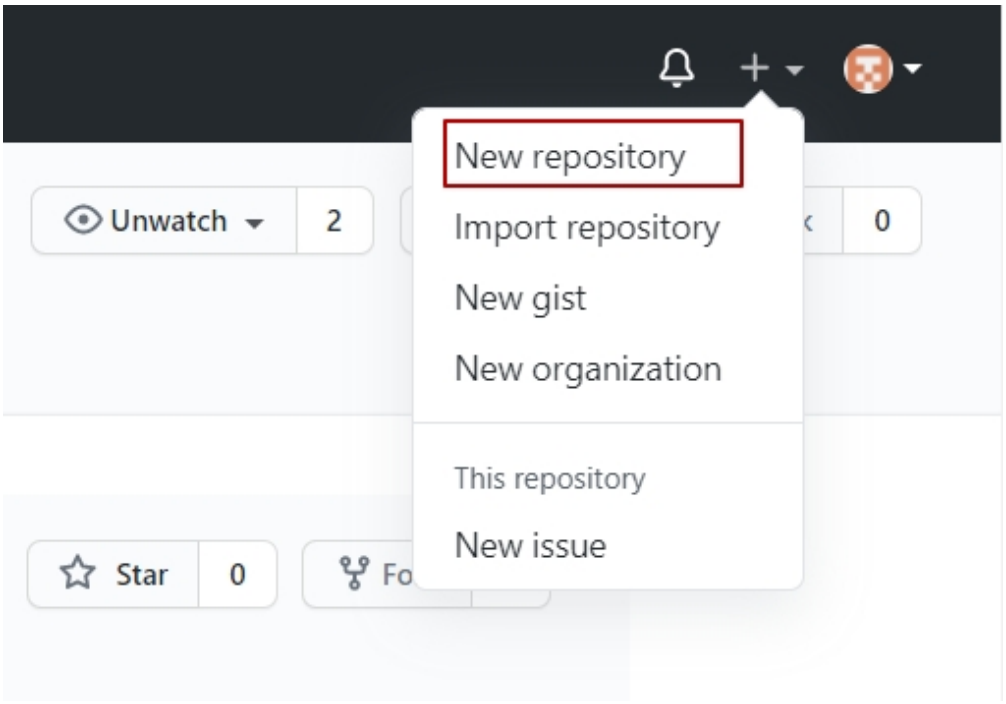
3. Click the **Create an account** button.
4. Complete the **CAPTCHA** puzzle. The instructions vary by puzzle, so just follow the on-screen instructions to confirm that you are a human.



5. Click the **Choose** button for your desired plan. Once you select a plan, GitHub will send an email confirmation message to the address you entered.
6. Click the **Verify email** address button in the message from GitHub. This confirms your email address and returns you to the sign-up process.
7. Review your plan selection and click **Continue**. You can also choose whether you want to receive updates from GitHub via email by checking or unchecking the “Send me updates” box.
8. Select your preferences and click **Submit**. GitHub displays a quick survey that can help you tailor your experience to match what you’re looking for. Once you make your selection, you’ll be taken to a screen that allows you to set up your first repository.

How to create a repository

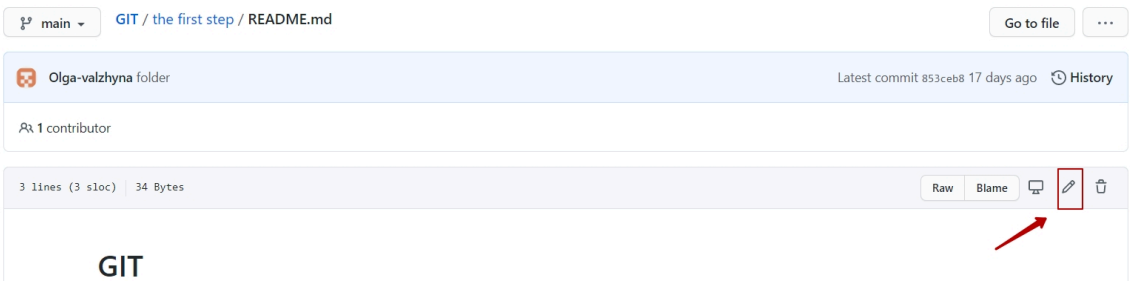
1. Go to the GitHub website, look in the upper right corner, and click the + sign and then click “New repository.”



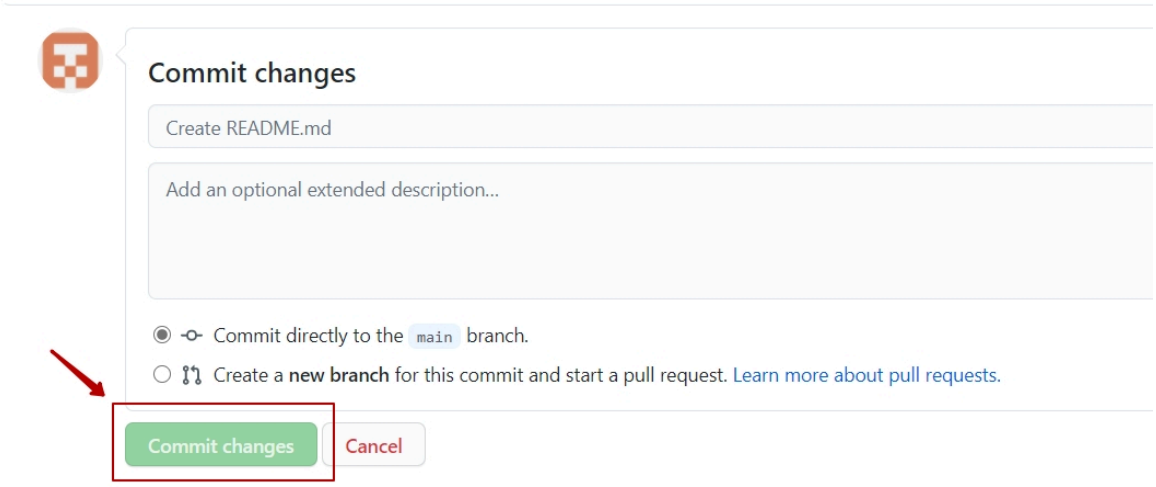
2. Name the repository, and add a quick description.
3. Decide whether you want this to be a public or a private repository.
4. Click **Initialize this repository with a README** if you want to include the README file.
5. Click **Create repository**

How to make changes to your README file

1. Go to your repository.
2. Click the name of the file to bring up that file (for example, click **README.md** to go to the readme file).
3. Click the pencil icon in the upper right corner of the file and make some changes.

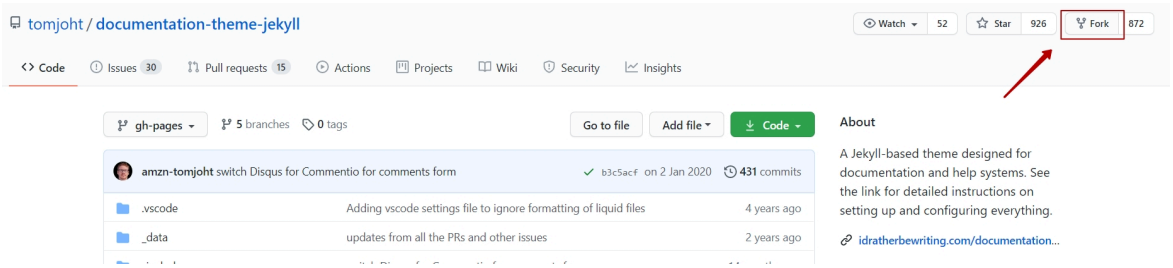


4. Write a short message in the box that describes the changes you made (and an extended description if you want).
5. Click the **Commit changes** button.



How to fork a repository

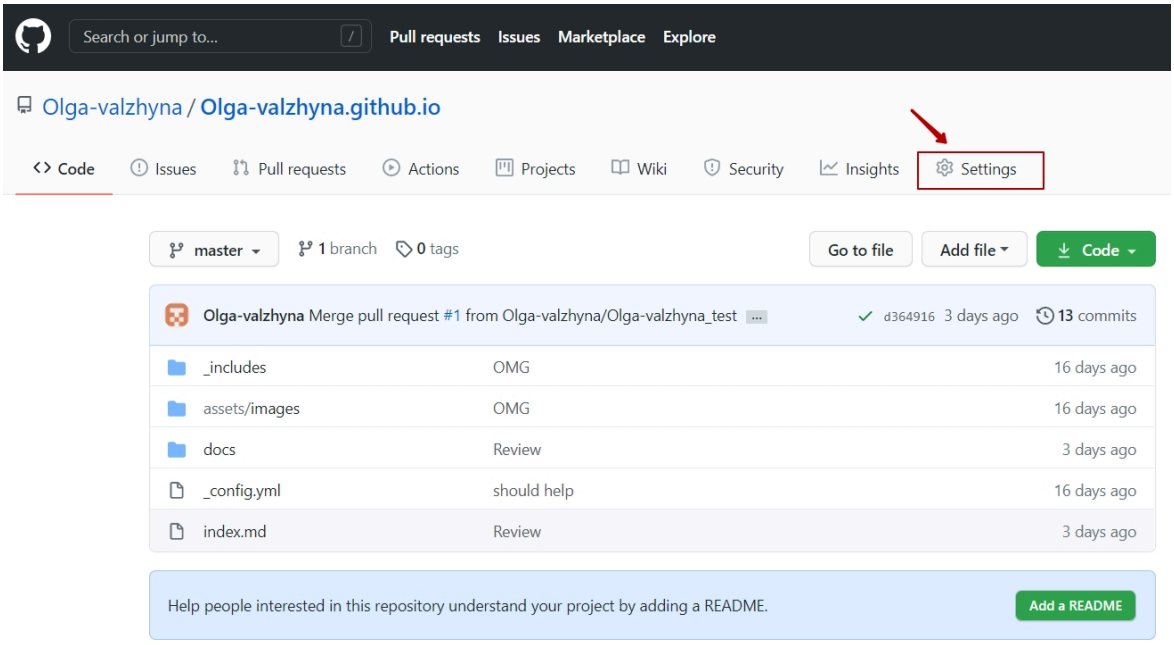
1. Search for the repository that you need in the search bar.
2. Select the one you need among the repositories.
3. Press **Fork button** to initiate the git forking process.



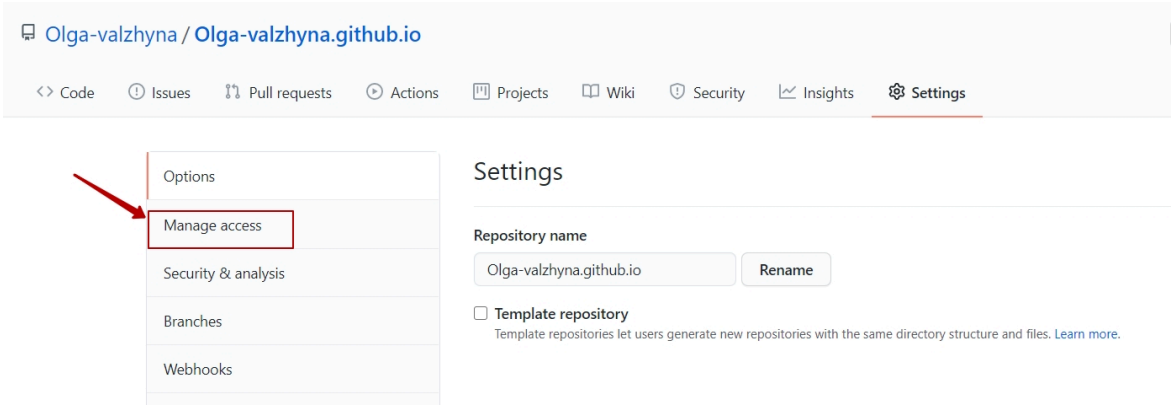
4. The repository will be forked to your account instantly. This can be seen by your username after forking the repository.
5. If you want to store the repo on your local machine, on the right side of the screen, below the **Contributors** tab, you'll see a green button that says **Clone or Download**. Go ahead and click that.

How to invite collaborators to your repository

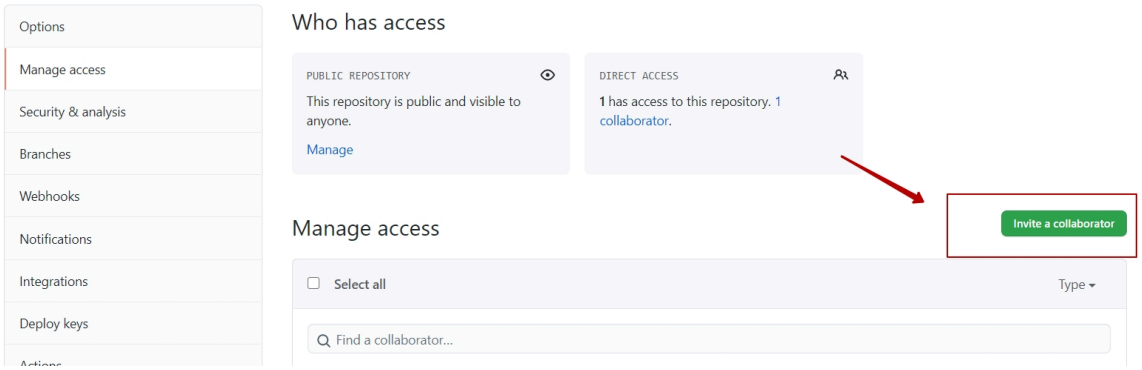
1. Under your repository name, click **Settings**.



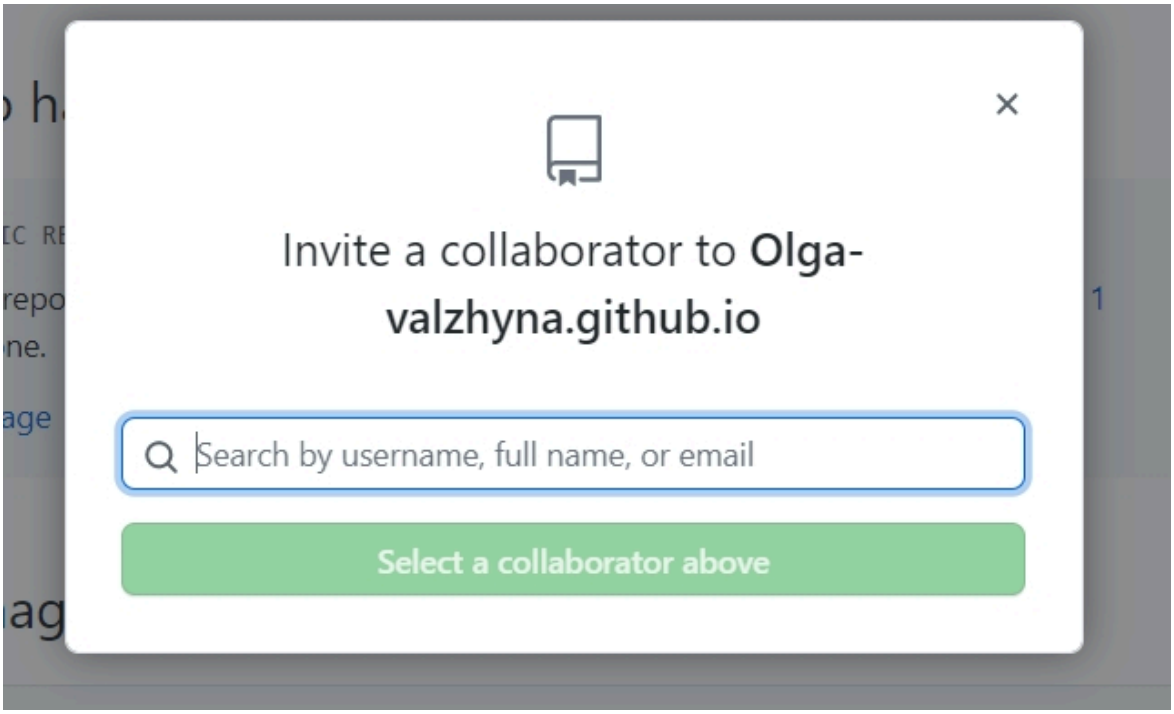
2. In the left sidebar, click **Manage access**.



3. Click **Invite a collaborator**.



4. In the search field, start typing the name of the person you want to invite, then click a name in the list of matches.

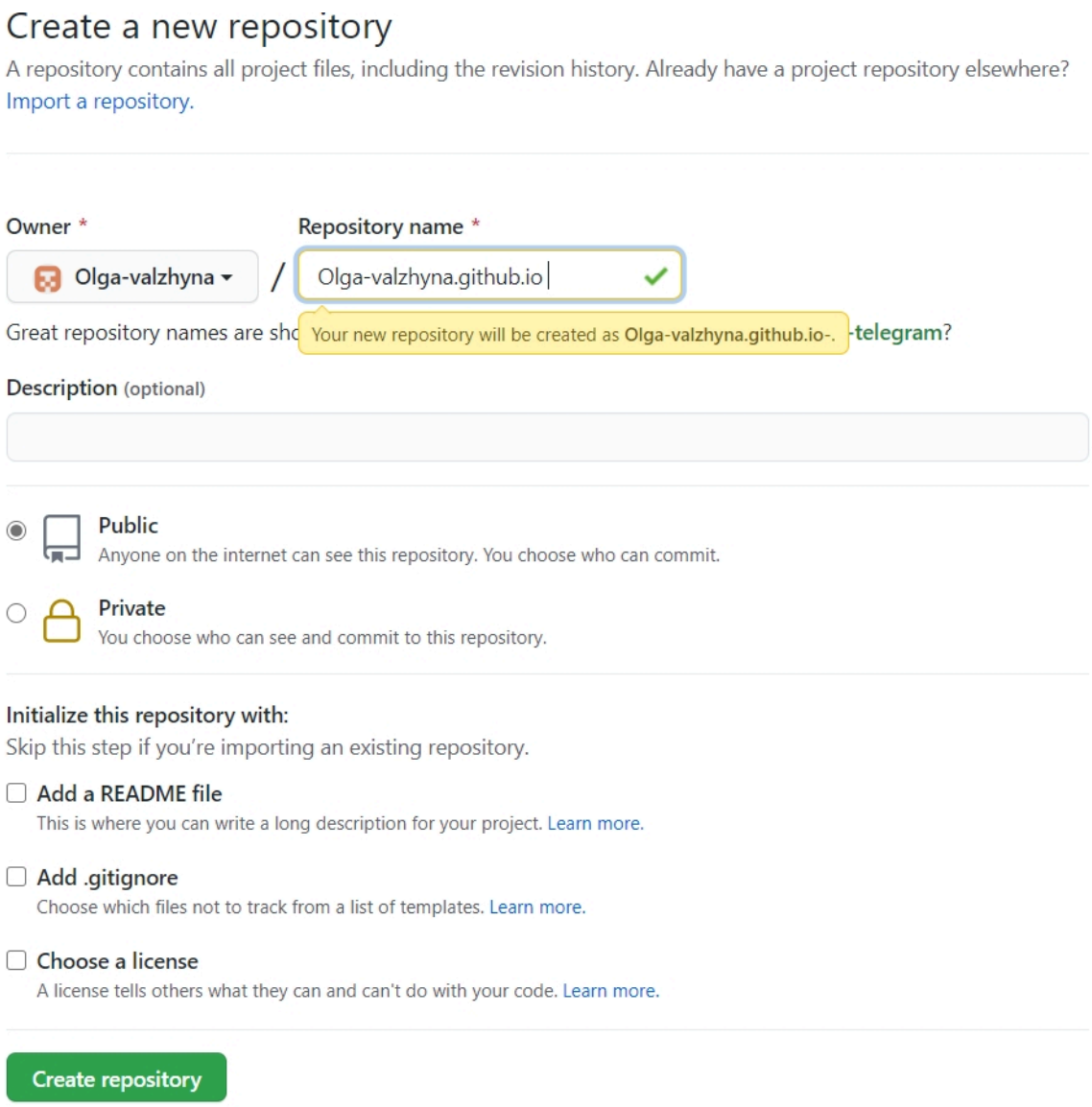


5. Click Add NAME to REPOSITORY.
6. The users will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.

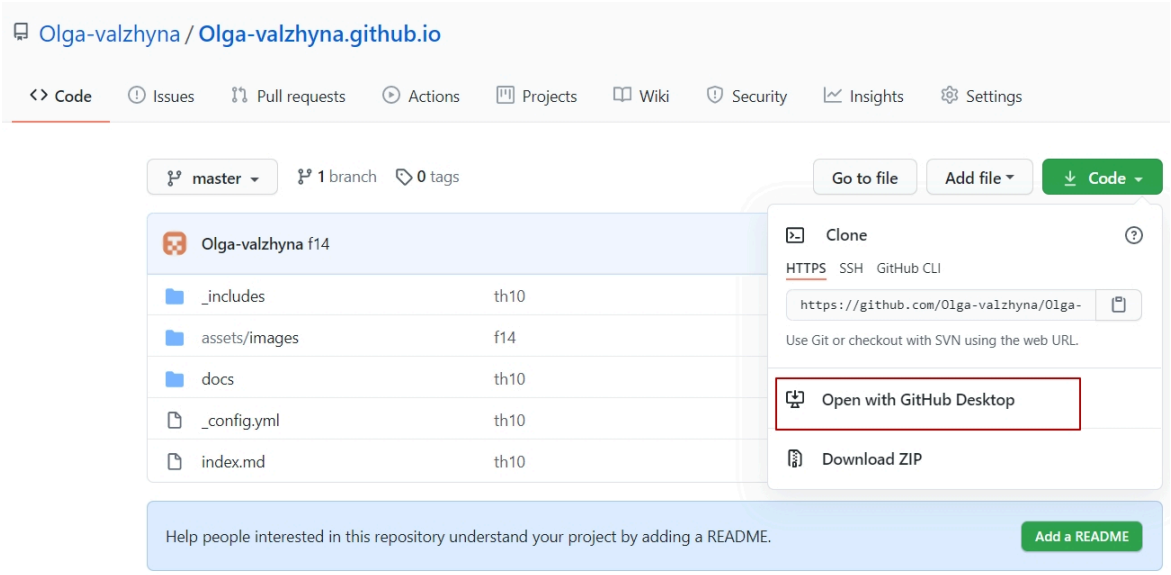
GitHub Pages

GitHub Pages is a service that hosts static websites, which can generate websites directly from source Markdown files hosted on a GitHub repository. If you know how to use GitHub and you need to create a simple webpage, you can't do better than GitHub Pages. It is an example of the most basic way of hosting a website for free on the Internet. Just create a new repository on GitHub, commit the Markdown files, and enable the GitHub Pages feature. In order to start working with GitHub Pages, follow the instruction.

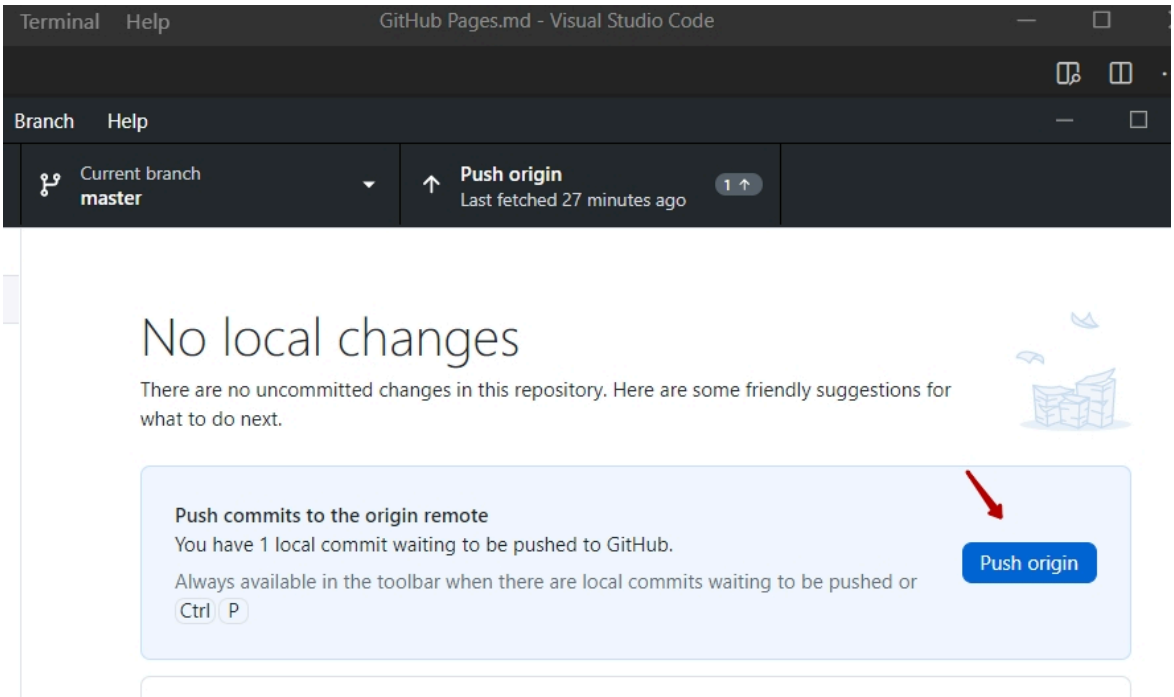
1. Create a new public repository named **username.github.io**, where username is your username (or organization name) on **GitHub**.



2. Clone the new repository with the help of **GitHub Desktop** to your local folder.



3. Enter the project folder and add an **.md** file, using **Studio Visual Code**(or any other text editor).
4. Check changes in **GitHub Desktop**. Commit and push them.



5. Go to **https://username.github.io**.

Just-the-Docs and Jekyll

Just the Docs gives your documentation a jumpstart with a responsive Jekyll theme that is easily customizable and hosted on GitHub Pages.

Just the Docs

Just the Docs is built for a static site generator, that is called Jekyll. Just the Docs requires no special plugins and can run on GitHub Pages *(on page 52)*' standard Jekyll compiler.

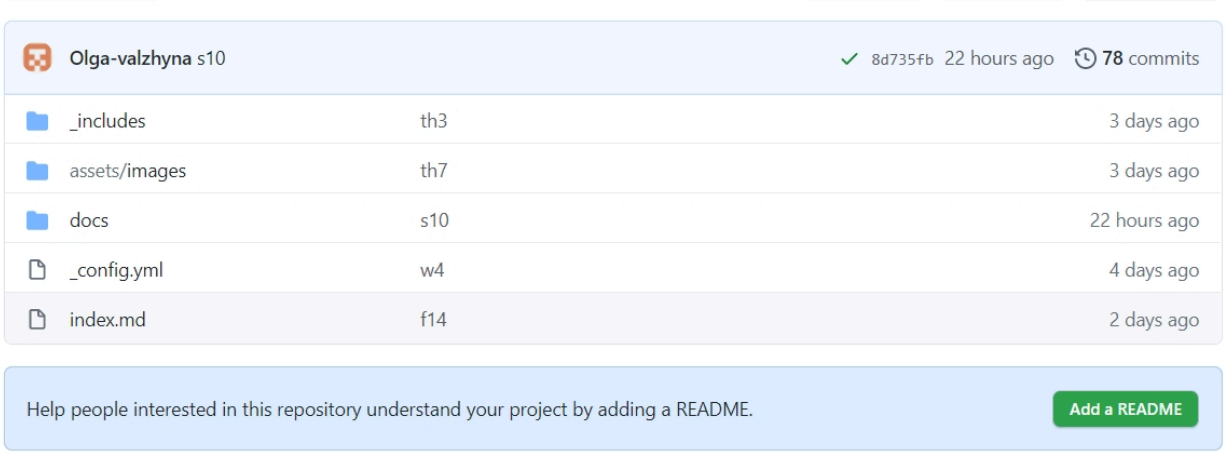
Jekyll

Jekyll is a simple, blog-aware, static site generator perfect for personal projects, or organization sites. Jekyll takes your content, renders Markdown templates, and spits out a complete, static website. As Jekyll is a static site generator, it does not use databases to generate the pages dynamically. Instead of using databases, Jekyll supports loading content from YAML, JSON, CSV, and TSV files. Jekyll is the engine behind GitHub Pages, a GitHub feature that allows users to host websites based on their GitHub repositories for no additional cost.

Username.github.io

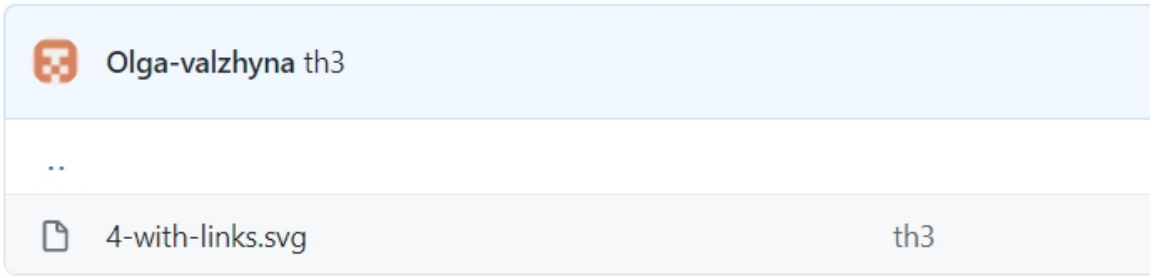
Create a new repository named **Username.github.io**(where "username" is your actual GitHub user name). If you scroll down on the settings page, you'll see the **GitHub Pages** section near the bottom. Click the **Choose a theme** button to start the process of creating your site.

In order to create a site with the help of **Just the Docs** your **Username.github.io** should contain the following folders and files.



1. **_includes**

   It's a folder with SVG files (in case you have them. If not, you don't need such a folder).



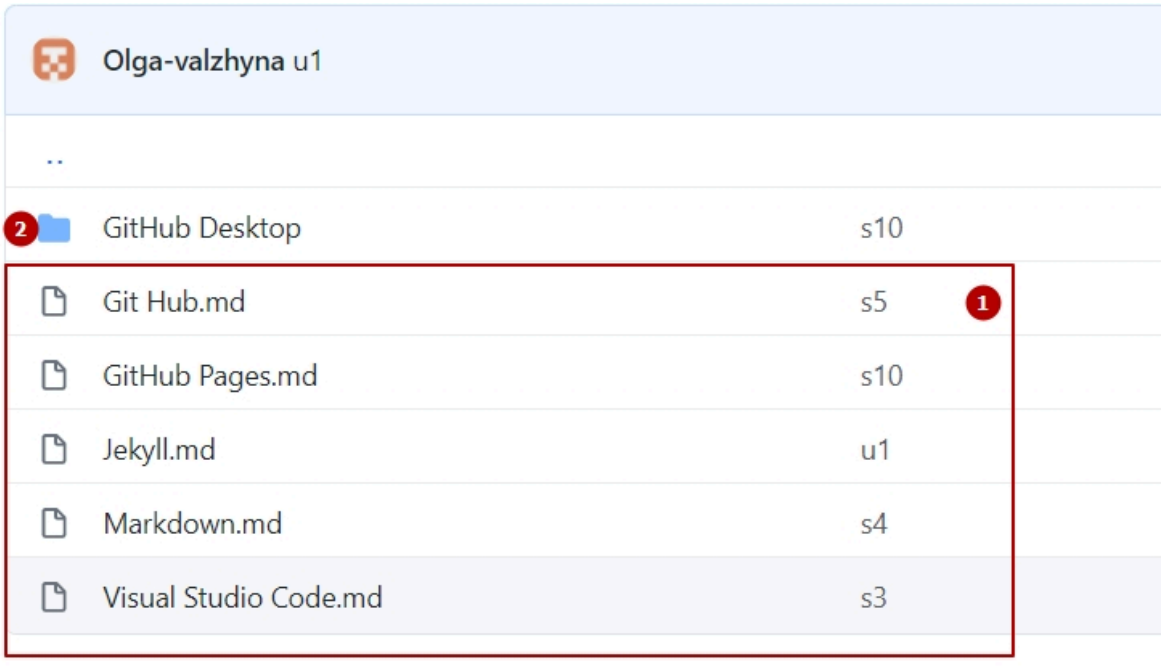
2. **assets/images**

   This folder contains JPGs, PNGs and PDFs.

3. **docs**

   All the documents are presented here.

And depending on the location of the document their structure is different.



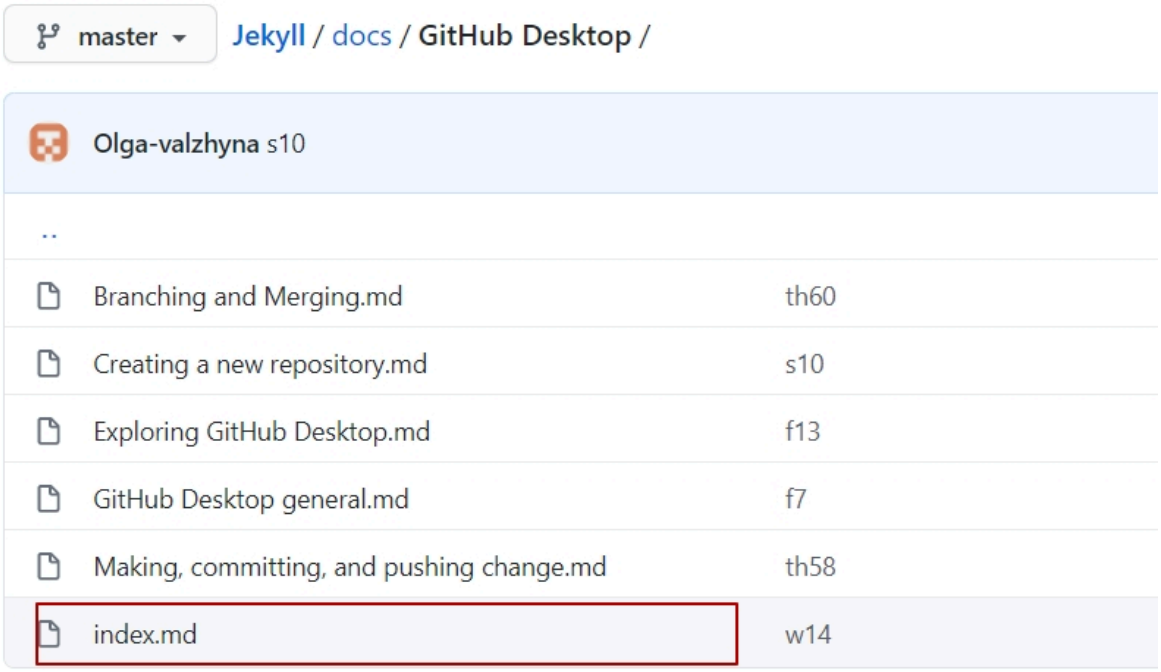
a. 1 docs/file.md

```
--
layout: default
title: Title
nav_order: 2
---
# Title
Markdown text goes here.
```

b. 2 docs/chapter/file.md

```
---
layout: default
title: Title
parent: Chapter title
nav_order: 1
---
# Title
Markdown text goes here.
```

Within each chapter there should be a separate **index.md** file



docs/chapter/index.md

```
---

layout: default

title: Chapter title

nav_order: 3 (the order in the main Table of Contents)

has_children: true

---

# Title

Markdown text goes here.
```

4. **_config.yml**

   This file is copied from **Just the Docs**

```
name: _____________

title: Page title

description: Page description

remote_theme: username/just-the-docs (username = your account name)

color_scheme: light/dark

aux_links:

"Source repository on GitHub":

- "//github.com/username/username.github.io“

back_to_top: true

back_to_top_text: "Back to top“

footer_content: "_____________"
```

5. **index.md**

```
---

layout: default

title: Introduction (is presented in Table of Contents)

nav_order: 1 (index.md is the first page in your Table of Contents)

description: “Description of this chapter"

permalink: /

---

# Introduction

Markdown text goes here.
```

Chapter 6. Tools

Using the right tools for technical writing makes the life of a technical writer easy. There are a lot of different tools available for variable purposes — authoring, publishing, screen captures, drawing, image manipulation, and more.

Programs used during the course

1. **Jira**

   https://www.atlassian.com/software/jira
2. **Confluence**

   https://www.atlassian.com/software/confluence
3. **Snagit**

   https://www.techsmith.com/screen-capture.html
4. **Camtasia**

   https://www.techsmith.com/video-editor.html
5. **Brackets**

   https://brackets.download/
6. **Visual Studio Code**

   https://code.visualstudio.com/Download
7. **GitHub Desktop**

https://desktop.github.com/

8. **MadCap Flare**

   https://www.madcapsoftware.com/products/flare/

9. **Oxygen**

   https://www.oxygenxml.com/xml_editor/download_oxygenxml_editor.html

10. **memoQ**

    https://www.memoq.com/

Spelling and grammar

- Grammarly
- DeepL

Readability

Hemmingway App

Diagrams

Gliffy for Chrome

Screenshots

- Snipping tool
- Screenpresso
- FireShot

Availability

- Validators
- Narzędziownia Polskiej Akademii Dostępności

Tasks management

- Trello
- To-Do (MS Planner)
- Google Keep/Tasks
- Todoist

Video/Animations

- Adobe CS (Premiere, After effects)
- Cinema 4D
- Mocha AE
- iSpring Suite

Screen recording

- Camtasia/Snagit
- Powerpoint

GIFs

- Camtasia/Snagit
- Converters online, for example convertio.co/mp4-gif/

MadCap Flare

MadCap Flare — a native XML content authoring application, which offers single-source authoring in an environment that does not require to know any coding.

Workflow

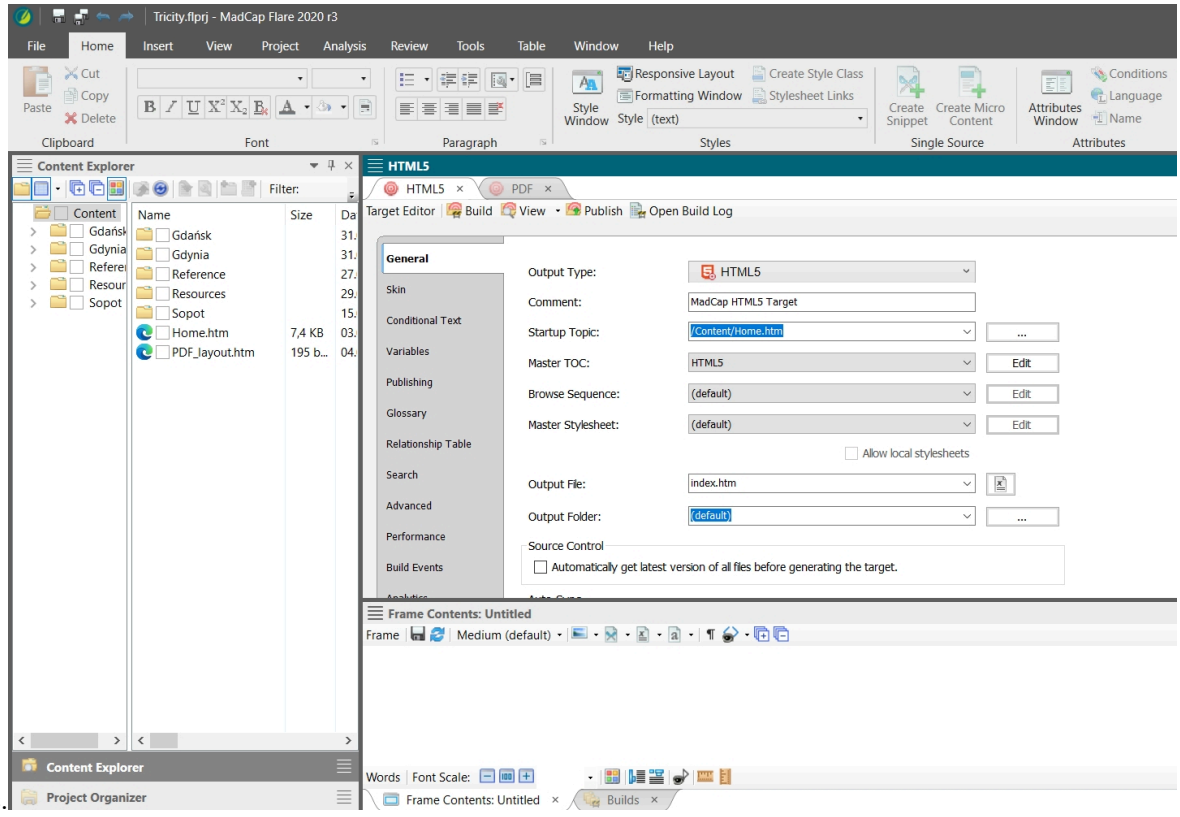
From a single Flare project you can produce output in many different formats, including documentation for the web, desktop, print, and mobile.

Process

When you break down the authoring process in Flare, you will discover that it can be quite simple. Following are the basic steps that you need to follow for creating and developing a project in Flare.

1. **Start Projects** Create a project from scratch, or start a project by importing existing content from a variety of sources.
2. **Add Content and Features** Add content and features, such as topics, text, a table of contents,cross-references, navigation, page layouts, and all of the other elements necessary to help your end users.
3. **Design** Through the use of features such as stylesheets, skins, page layouts, master pages, and more, you can design a look and feel for your output.

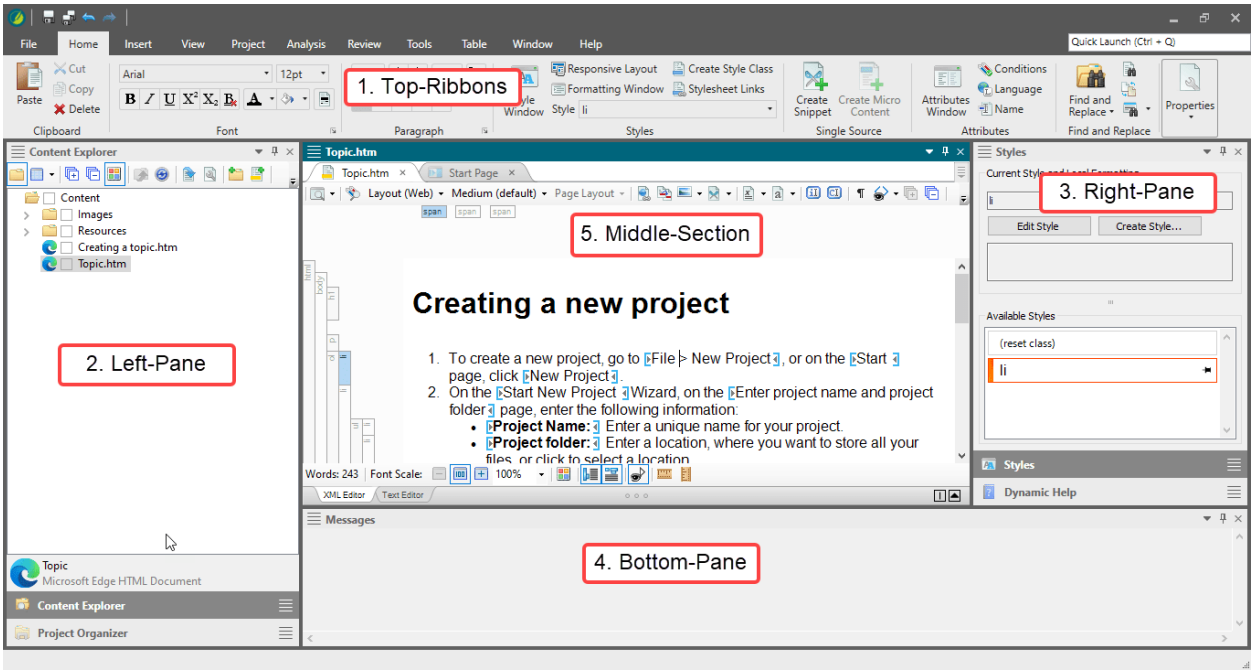
4. **Develop Targets** Decide the type(s) of output formats that you want to generate and develop targets accordingly to meet your needs.
5. **Build and Distribute Output** Generate output from a target, then make the files accessible to your end users.



Interface

Flare uses a modern Multiple Document Interface (MDI) that provides you many options to work the way that you want.

It has the following main sections:



1. **Top–Ribbons**: This section contains all the features and menus.
2. **Left–Pane**: This section contains many explorers to access all the madcap flare files. You will find all your content files in this section. This section has the following tabs:
    - **Content Explorer**: In general you will find all your styles and content related files in this tab; for example, topics, stylesheets, master pages, and many more.
    - **Project Explorer**: You will find all your output related files in this tab; for example, target files, table of contents, condition tags, variables, and many more.
3. **Right–Pane**: This pane has the following tabs:
    - Find and Replace
    - Styles
    - Dynamic help
4. **Bottom–Pane**: You will see your build progress window and messages window in this pane.
5. **Middle-section**: You can see many of the flare files editors in this section; such as XML editor, TOC editor, Stylesheet editor. When you start the madcap flare application, you will find a start page appearing in this window, which you can use to perform some high-level tasks and access information.

Note:

Video - Official Webinar: An Overview of MadCap Flare

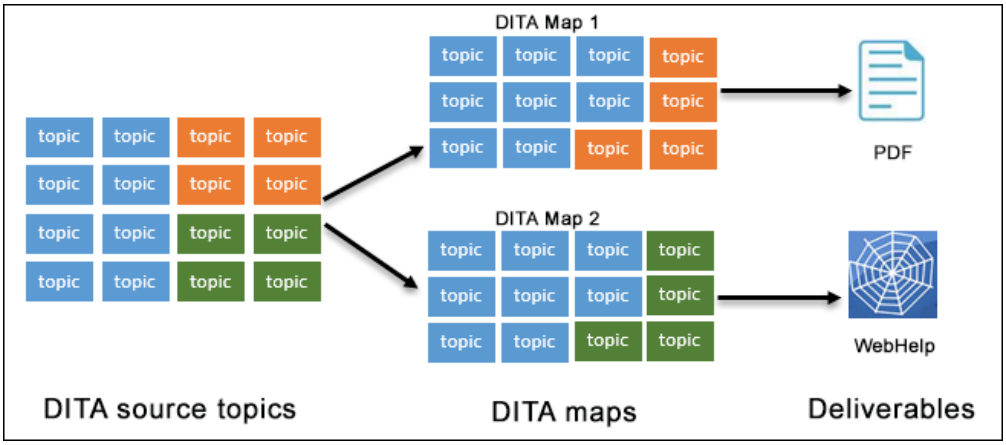
DITA and Oxygen

The Darwin Information Typing Architecture (DITA) specification defines a set of document types for authoring and organizing topic-oriented information, as well as a set of mechanisms for combining, extending, and constraining document types. It is an open standard that is defined and maintained by the OASIS DITA Technical Committee.

DITA

The name DITA derives from the following components:

- **Darwin**: it uses the principles of specialization and inheritance, which is in some ways analogous to the naturalist Charles Darwin's concept of evolutionary adaptation,
- **Information Typing**: which means each topic has a defined primary objective (procedure, glossary entry, troubleshooting information) and structure,
- **Architecture**: DITA is an extensible set of structures.



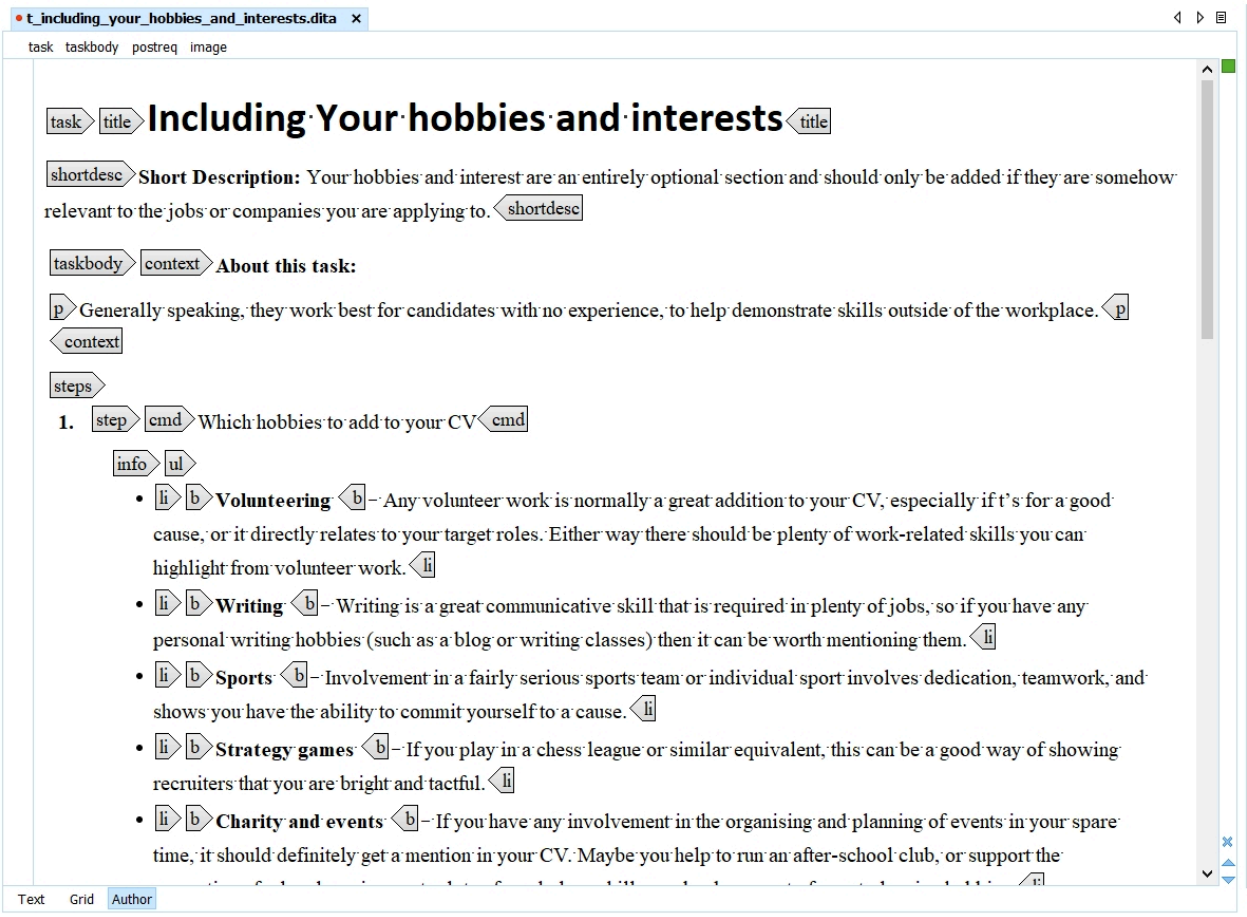
Information typing

The latest version of DITA (DITA 1.3) includes five specialized topic types: **Task, Concept, Reference, Glossary Entry, and Troubleshooting**. Each of these five topic types is a specialization of a generic Topic type, which contains a title element, a prolog element for metadata, and a body element. The body element contains paragraph, table, and list elements, similar to HTML.

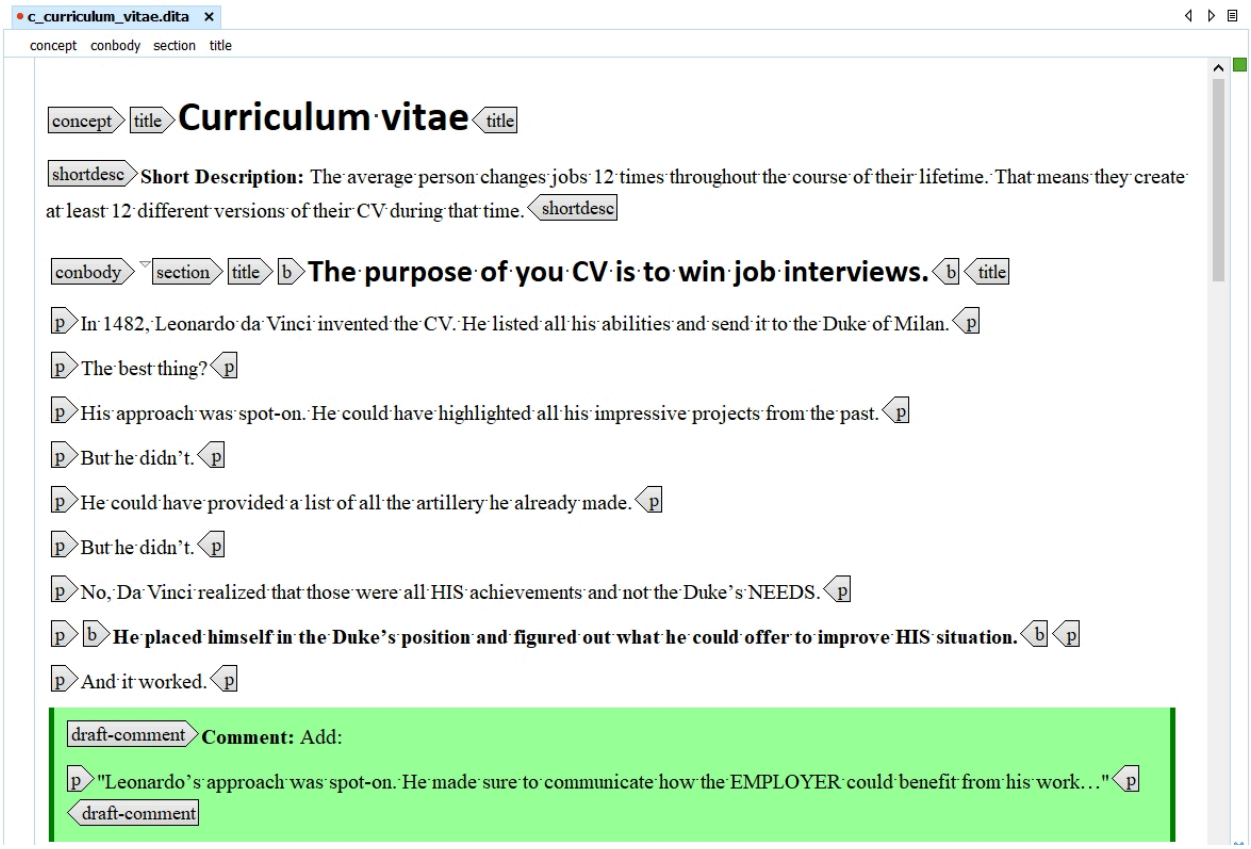
DITA content is created as topics, each an individual XML file. Typically, each topic covers a specific subject with a singular purpose, for example, a conceptual topic that provides an overview, or a procedural topic that explains how to accomplish a task. Content should be structured to resemble the file structure in which it is contained.

A **Task** topic is intended for a procedure that describes how to accomplish a task. It lists a series of steps that users follow to produce an intended outcome. The steps are contained in a taskbody element, which

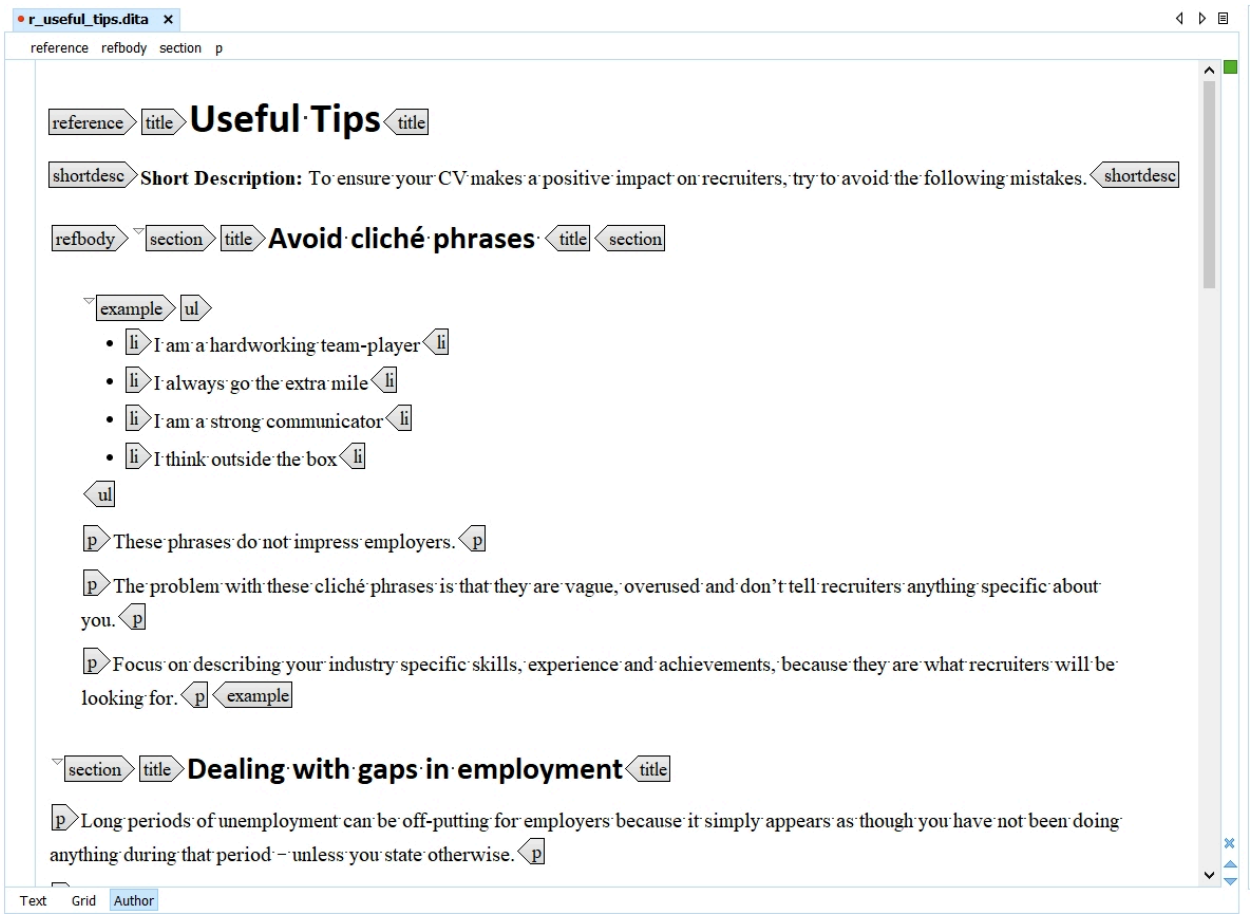
is a specialization of the generic body element. The steps element is a specialization of an ordered list element.



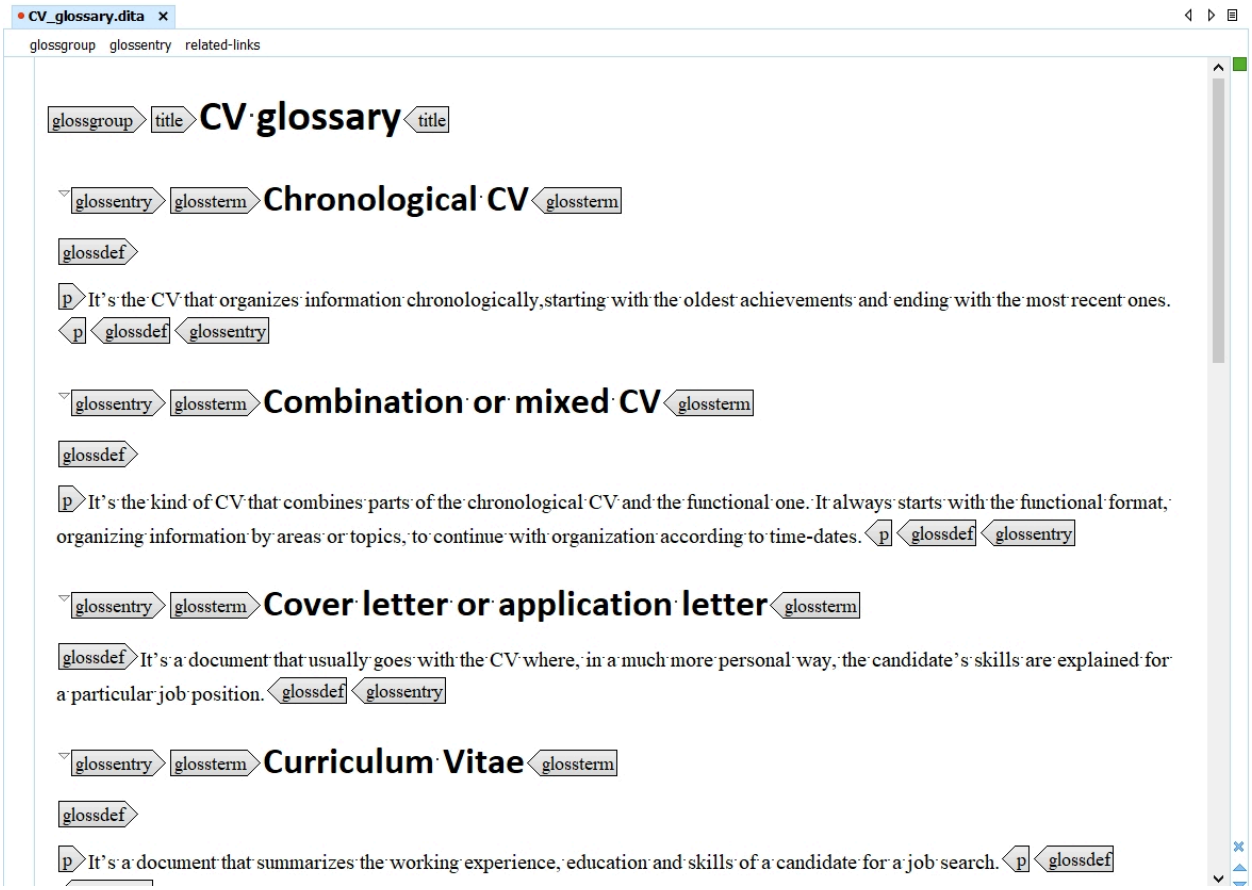
Concept information is more objective, containing definitions, rules, and guidelines.



A **Reference** topic is for topics that describe command syntax, programming instructions, and other reference material, and usually contains detailed, factual material.

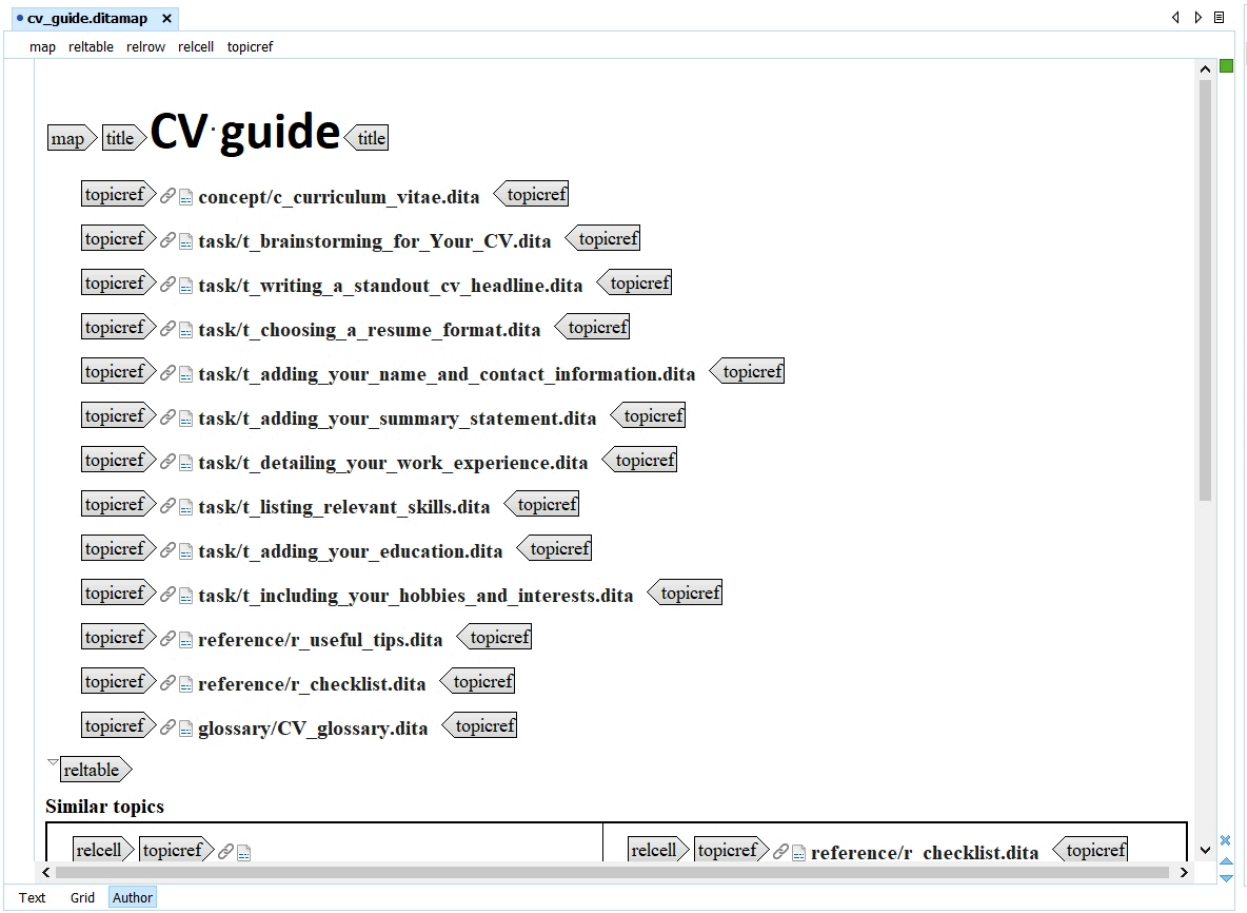


A **Glossary Entry** topic is used for defining a single sense of a given term. In addition to identifying the term and providing a definition, this topic type might also have basic terminology information, along with any acronyms or acronym expansions that may apply to the term.



Maps

A DITA map is a container for topics used to transform a collection of content into a publication. It gives the topics' sequence and structure. A map can include relationship tables (reltables) that define hyperlinks between topics. Maps can be nested. Maps can reference topics or other maps, and can contain a variety of content types and metadata.



The **Troubleshooting** topic describes a condition that the reader may want to correct, followed by one or more descriptions of its cause and suggested remedies.

Metadata

DITA includes extensive metadata elements and attributes, both at topic level and within elements. Conditional text allows filtering or styling content based on attributes for audience, platform, product, and other properties. The conditional processing profile **(.ditaval file)** is used to identify which values are to be used for conditional processing.

oXygen XML Editor

The Oxygen XML Editor is a multi-platform XML editor, XSLT/XQuery debugger and profiler with Unicode support. It is a Java application, so it can run in Windows, Mac OS X, and Linux.

Oxygen XML offers three views designed for editing XML documents. These views are text, grid, and author.

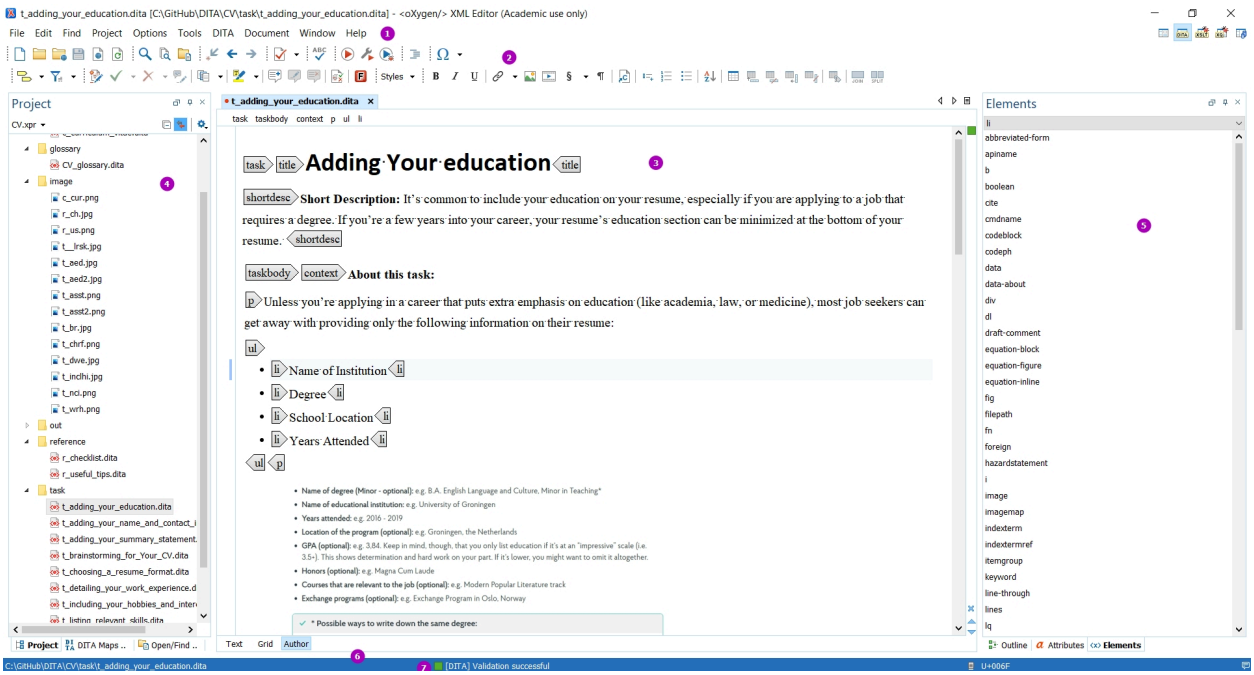
- The **text view** is the default view for editing an XML document. As the name suggests, this view shows the XML text as text.
- The **grid view** shows the XML document in a spreadsheet-like fashion. The left-most column shows the elements, including comments and processing instructions, at the root level. The next column shows attributes of root elements, and every unique first child of the root XML element. If the root element has six children all named “section”, then the grid view will show only one section element and a notation that there are six of them. This iteration continues for the next column.
- The **author view** is based on providing a CSS file for the document that specifies the data type for each element in the document’s schema. Oxygen XML comes with document CSS files for formats like DITA, DocBook, and TEI. XML tags and attributes in this view can be completely disabled or can be shown in various combinations.

oXygen interface

An introduction to the main components of the Oxygen interface.

When you open Oxygen XML Editor, you will see the editor in Text Editing Mode, as below. If instead you see the Welcome dialogue box, uncheck Show at startup in the lower left corner and click Close.

Begin by familiarizing yourself with the different sections of the Oxygen interface.



1. The **Menus** at the top are similar to other Windows programs, although the options are different.
2. The **Toolbars** below the **Menus** provide easy access to common functions. They are also fully configurable by right-clicking in an empty part of the menu bars. The image above contains several menus in addition to the default menus.
3. The **Editor Pane** in the center of the interface allows you to work with files in three different **Editing Modes**, which you select at the bottom of the pane. Text mode is shown, where you can see the coding directly. Author mode provides something similar to a WYSIWYG interface, where you can see an approximation of the final appearance of page, but it is less convenient to work in when editing or proofing documents.
4. The **Project View** in the left-hand pane is one of several different views that you can dock in place. Project View lets you see all of the files and folders in the current xml-project. By right-clicking a file or folder, you can also check validation and run transformation scenarios on the selected files.
5. The **Model View** on the right is, again, one of several different helper views that can be docked in place or hidden from view. The Model View shows the definition of the XML element at the cursor placement. Additional views can be seen by clicking the tabs on the right.
6. The **Results View** displays the messages generated as a result of user actions such as validations, transformations, search operations, and others. Each message is a link to the location related to the event that triggered the message. Double-clicking a message opens the file containing the location and positions the cursor at the location offset.
7. The blue **Status Bar** at the bottom of the window includes the following information, in the order it is displayed from left to right:

- The path of the current document.
- Information about the most recent operation, such as formatting or validating.
- The Unicode value for the character directly to the right of the current cursor position
- The status of the current document. The status of Modified is displayed for documents that have not yet been saved. Otherwise, this section is left blank.
- In Text editing mode, the current line and character position is displayed.

Localization and memoQ

Many people think that localization is mainly connected with translation. But technical writing may be closely related to it as well. Localization, in this case, is the adaptation of your documentation to the culture of the targeting country.

Localization

Localization is the adaptation of your documentation to the culture of the targeting country. It is a bridge between the cultures. If your product is going to enter the market in another country, you should write in a manner that will enable the translators to make your texts suitable for a foreign culture. This is an additional responsibility for technical writers. It might seem surprising, but localization takes place not only when we talk about cultures as different as Europe and Asia. Localization takes place as well in the documents that come from the USA to Great Britain and vise versa, or Spanish speaking countries, etc. The language might be the same, but due to local particularities, it needs adaptation.

Localization in documentation may touch only the most obvious things like date format, currency, measurements, and other small inconsistencies. At the same time, localization may have a deeper character adapting more global things like the questions of rhetoric.

So, what should a technical writer bear in mind writing documentation for a foreign market?

- **Additional space.** As a rule, companies want the documents in different languages to be identical and to look the same. It is easier for the readers to navigate and find the page or passage they need. This is a challenging thing to make the documents identical. You should be ready that your text in another language may become longer. Sometimes, this difference may be crucial. The text simply will not fit the layout. The best idea is to use additional space in the original document so that the new text does not ruin the whole document.
- **Wise font choice.** Fonts are not universal for all the languages. Not all fonts are suitable for Asian languages, for example. The same can be said about many other languages. So, to avoid problems here, you should choose your fonts carefully. Make sure that your fonts are suitable for all the languages that are going to be used in your documents. Besides, you should limit the number of fonts you use for documentation. This will contribute to simplicity, as well.

- **Careful use of abbreviations and acronyms.** They both can be a stumbling point of translators and technical communicators. A correct way out is to use the full version when they are going to be mentioned in the text for the first time. Then you can use the short version.
- **Separate use of text and visuals.** When visuals contain text, that might be a problem. Technical writers or translators will have to prepare new ones with the proper text. Again that might affect the layout if the size of the visuals is wrong. It is better to avoid it and separate the visual materials from the text.
- **Attention to detail.** One should keep in mind that the smallest details may affect the image of the product. For example, the choice of colors. In different cultures, they are perceived differently and have different meanings.

Poor localization spoils the image of the product on the market and creates a feeling of awkwardness. It all starts with misunderstanding between the cultures. That is a great deal of work to represent the product in the correct way, and if you are going to create documentation that will be translated into other languages, you can improve the results greatly if you know what may cause problems.

CAT tools

The **"CAT"** in CAT tool stands for **"Computer Aided Translation"** or **"Computer Assisted Translation"** but, as you might already know, it doesn't mean that a computer is actually completing the translation for you. CAT tools are different than "machine translation" – they assist a human translator in doing their

work more quickly and in managing their translation projects. CAT tools typically contain a translation memory, which stores previous source and target translations for easy reference while working. Term bases are also an integral part of translation tools, giving translators the ability to develop their own bilingual glossaries in their subject areas.

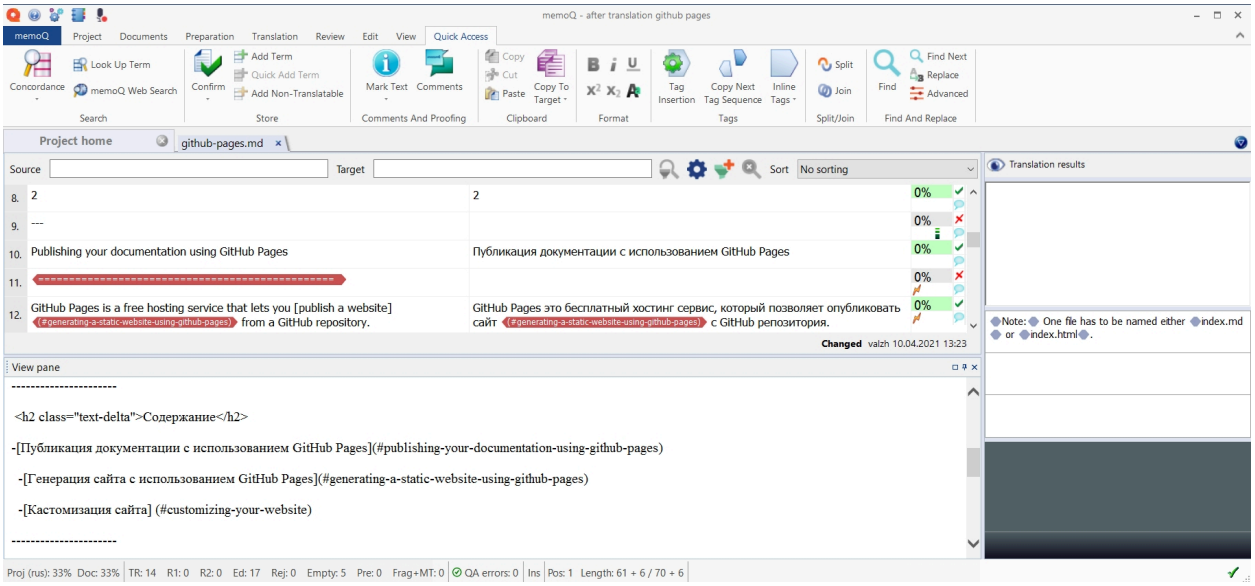
CAT tools can include a wide range of different features. Some can work with different types of documents, such as Powerpoint presentations, without needing to convert the text to a different file format. Some provide access to online terminology databases or help the translator to better manage translation memories. Some are software-based and some operate entirely in the cloud.

There is an incredible variety of CAT tools available on the market. As a translator, you could go crazy trying to find the perfect CAT tool, but really there is no single perfect CAT tool. Your choice will depend on a number of factors.

memoQ

memoQ is a CAT tool. It is a complete translation software solution for translating, managing terminology, editing, and running LQA. It has some useful extras such as the translation preview pane that allows you

to see the segment you are translating in-context. memoQ has a free version that is suitable for personal use or small projects, and the pro version is available with a perpetual license.



Chapter 7. Project management

Project management is the application of knowledge, skills, techniques, processes, expertise, and tools in order to meet project goals and requirements within a specific timeframe.

Projects are, essentially, just a means towards a specific business goal. They are a system of tasks and resources, all working together towards a common outcome that supports a certain business direction.

While projects will naturally vary between companies and industries, all should contain four core elements that make up the basics of project management: **scope, resources, time and money**.

Scope

The scope of a project refers to its size, goals and the requirements needed to achieve the desired outcome. Scope is perhaps the most important project element to clearly nail down before starting work. Changes in scope and unclear goals can all contribute to an exhaustion of project resources, missed deadlines or, even worse, the failure of the project altogether.

Resources

This refers to all human and material resources at your disposal to achieve your goal. This can be employees, freelancers, tools, software, etc. Having a clear view of your resources allows project managers to leverage them in the most efficient way possible the project goals.

Time

This includes both overall time for a project, from conception through to completion, as well as time requirements for specific tasks. Having clear breakdown of your required tasks, their lead times, dependencies and the “critical path” to project completion allows project managers to manage their resources smarter and more efficiently.

Money

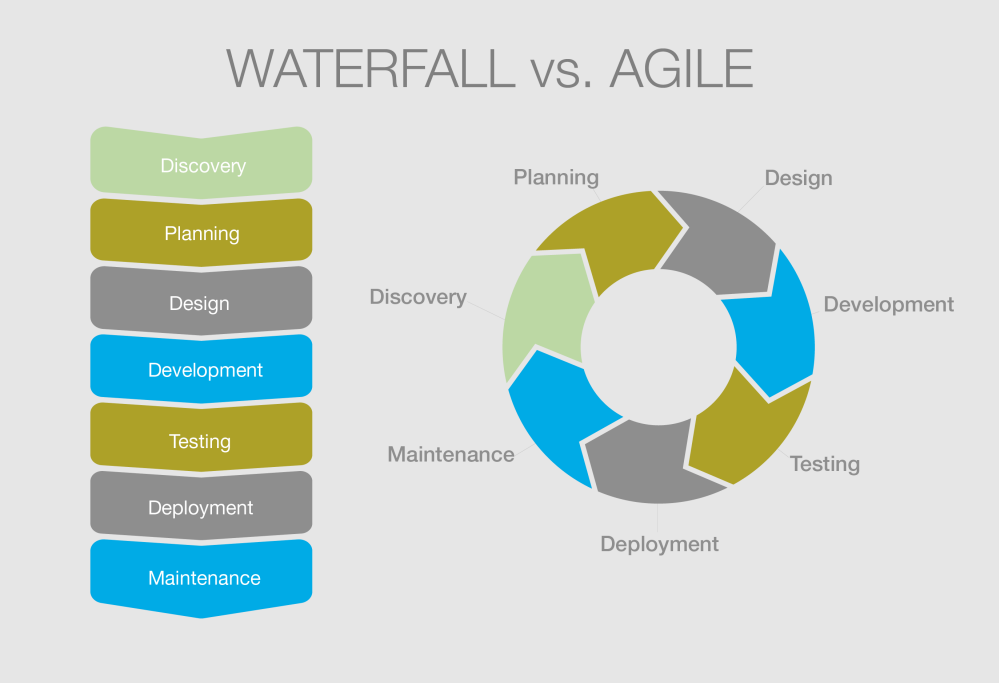
You should have a firm grasp on the cost of the project, and the potential profit it will generate for your business. Executing a simple cost-benefit analysis for your project will help you determine if the scope is too large, too small or if it’s even worthwhile at all.

Waterfall and Agile

Agile and Waterfall are both Software Development Lifecycle (SDLC) methodologies that have been widely adopted in the IT industry.

Waterfall methodology, also known as the linear sequential lifecycle model, is defined by its linear, structured approach to project management. It is made up of a series of steps that are completed in sequential order within the software development life cycle (SDLC).

In contrast to waterfall development, agile is defined by its iterative approach to project management. Instead of drafting lengthy project requirements at the onset, an agile team breaks out the product into specific features, and they tackle each one under a specific time constraint, known as a sprint.



Waterfall

The concept of the waterfall model incorporates a step-wise/sequential approach to project development. The project's progress flows progressively downward through all the phases, similar to a waterfall.

The "waterfall model" is also known as the "traditional" approach to project development and is known for taking things slow and steady. Each phase has to be finalized before moving on to the subsequent one.

On considering the waterfall methodology, you are aiming for a successful outcome as a result of meticulous planning at each phase in the process. Waterfall methodology mainly emphasizes on accuracy.

Pros

- The static and predictable workflow nature of this model makes it simple to create timelines, cost estimation, and stick to schedules.
- The waterfall model requires clear documentation of each phase in the process.
- The waterfall methodology is suitable for the projects which require various software components.
- Projects progress can be easily measured and evaluated.
- The team doesn't need any prior knowledge to start working on a project.
- Having a well-documented project and estimated timelines make it simple and easy to give timely updates to senior management.
- It's quite easy to manage due to an easy review process and distinct deliverables.
- It facilitates the speedy delivery of the product.
- The project undertaken is entirely reliant on the team with less client intervention.
- A quality assurance test is performed before the completion of each phase.

Cons

- Each phase in the process is dependent on others. Change in one step leads to change in other phases as well. Water methodology leads to higher time consumption.
- The process is not suitable for projects that have frequently changing requirements.
- Small changes in the middle of the process may lead to quite a lot of issues.
- The stakeholders/customers can't see the working of a project until completion of all phases before coding.
- Documentation consumes more time.
- Customers and stakeholders don't have any idea regarding what they want until they take a view of the working project. The waterfall model manages requirements gathering at the initial phase, so there are chances of missing the essential aspects down the line.
- There is a high tendency to neglect the testing phase, which can be risky for a project.

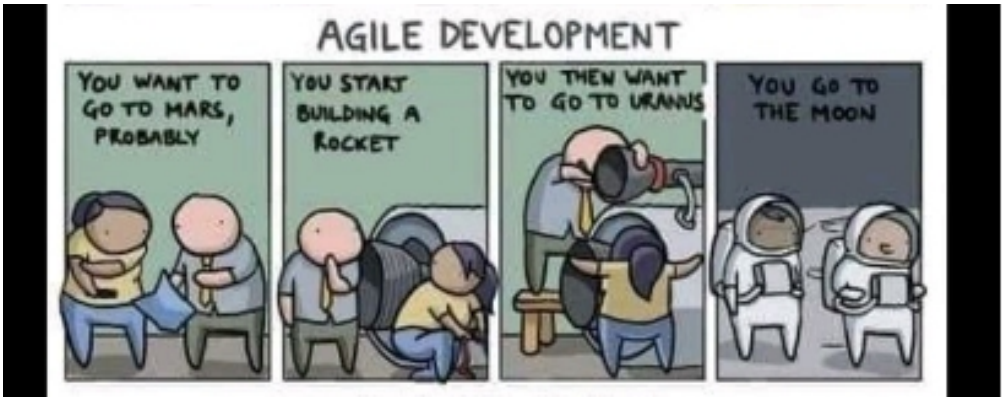
Agile

Agile methodology is defined as a modern, iterative, and adaptable approach to managing a project. It permits you to breakdown a large project into easier and convenient tasks which are accomplished in short iterations. Every iteration is referred to as a single development cycle and is reviewed by the concerned team.

Agile methodology allows the entire team to adapt to changes quickly with fewer alterations, which in turn end in the successful completion of the project despite the odds encountered. This methodology relies on customer involvement all through the development process.

Pros

- Customer satisfaction can be quickly done through rapid and continuous delivery.
- Agile provides better adaptability to changes with quick response.
- Special attention to design and technical details enhance agility.
- Agile facilitates regular cooperation between the developers and the stakeholders.
- People interaction is emphasized rather than any process and tools.
- The Agile methodology supports a continuous development pace.
- You will get instant feedback.
- You can fix issues quickly.



Cons

- In the case of large and complex projects, it's quite challenging to assess the effort required during the initial phase of the project.
- Agile provides the least priority to designing and documentation, which can lead to problems. Organizations can easily step off the project if the customer is not transparent with the outcome.
- There is no pre-defined plan. The agile technique works at its best when the designers are capable of being agile.
- Senior programmers are proficient in taking decisions required for the development process. So there is no place for amateur programmers lacking experienced resources.

Scrum and Kanban

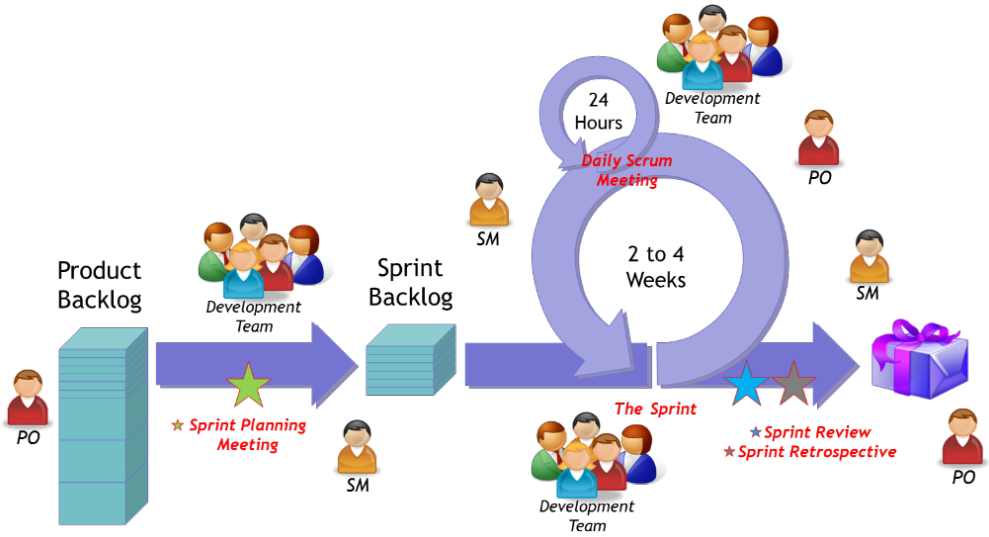
Scrum and Kanban are two flavours of Agile software development - two deceptively simple but surprisingly powerful approaches to software development.

SCRUM		KANBAN
A Scrum Master, Product Owner and Team Members make up a Scrum Team.	Team Roles	No set roles are defined. Roles are not required to be cross functional.
Columns are labeled to reflect periods in the work flow from beginning through the team's definition of done.	Work Boards	Columns are likewise labeled to show work flow states, but also publish the max number of stories allowed in column at once.
Scrum processes place heavy emphasis on schedule with a prioritized list of story points. This iterative process enables accurate estimations of work flow and effective management of multiple projects.	Scheduling/ Cadence	There are no required time boxes or iterations. While the Kanban method is iterative in nature, the continual improvement is expected to occur in an evolutionary fashion as work is continually completed.

Scrum

Scrum is a framework utilizing an agile mindset for developing, delivering, and sustaining complex products,with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies.

Scrum denotes five time-based events for managing product delivery iteratively and incrementally, while maximizing opportunities for feedback. These are:



- **Sprint Planning.** An eight-hour session where the team decides what to deliver in the coming sprint (from the product backlog) and how to go about it.
- **Sprint.** A timeframe of a month or less where the team delivers what was agreed in the sprint planning session.
- **Daily Scrum.** A 15-minute timebox (commonly referred to as daily stand up) where the team meets daily during the sprint to inspect progress and identify blockers.

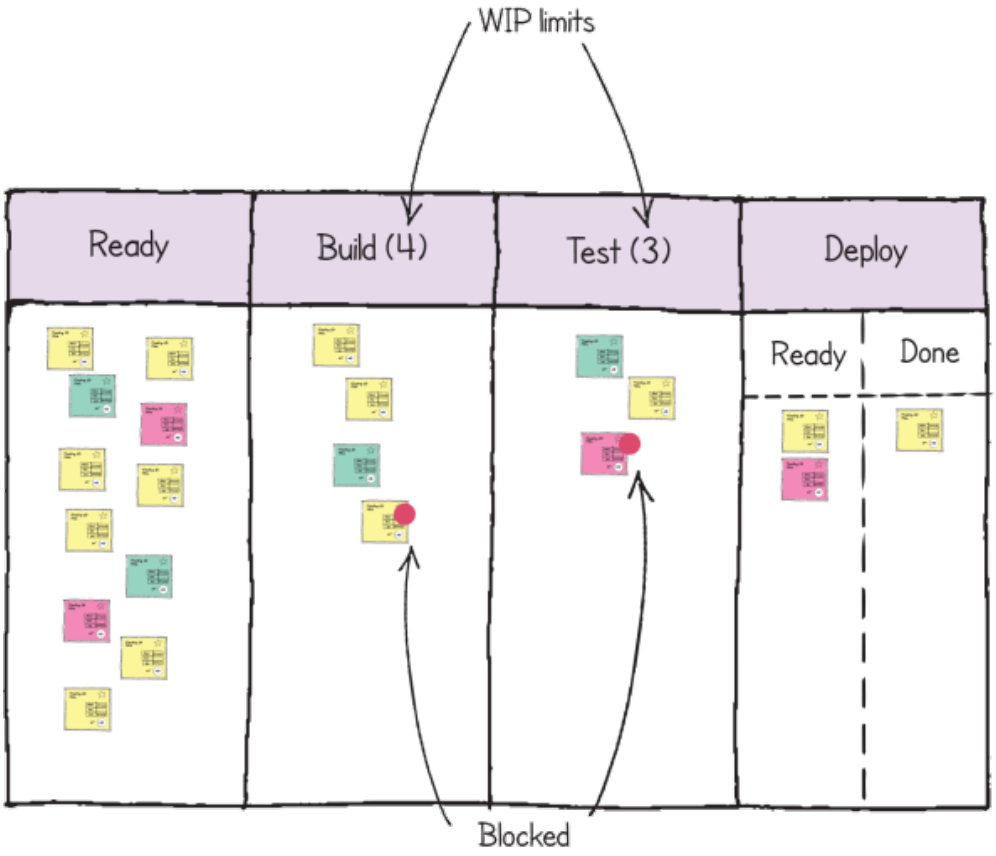
- **Sprint Review.** A four-hour timebox event held at the end of the sprint. The team demonstrates the product/changes to customers and gathers feedback on what to incorporate in the product backlog for delivery in subsequent sprints.
- **Sprint Retrospective.** A three-hour timebox event held after the sprint review (and before the next sprint planning). The team reviews their work, identifying opportunities for improvement work processes in subsequent sprints.

Kanban

Kanban is an Agile framework that uses visualization to understand processes and workflows better and actual work done in those processes.

If you decide to apply Kanban, follow these six core practices:

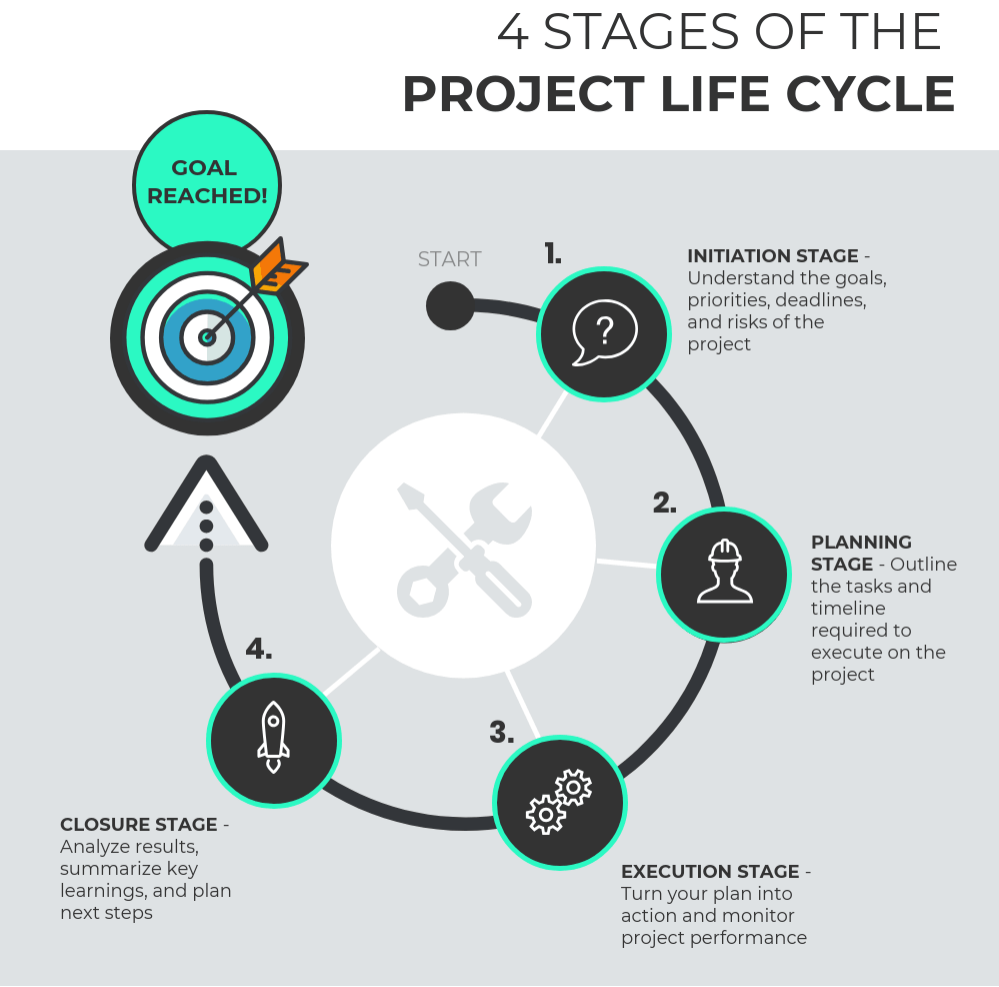
- Visualize the Flow of Work. Use cards or software to visualize the process activities on swim lanes.
- Limit Work in Progress (WIP). Encourage your team to complete work at hand first before taking up new work. The team pulls in new work only when they have capacity to handle it.
- Manage Flow. Observe the work as it flows through the swim lanes. Address any bottlenecks.
- Make Process Policies Explicit. Visually diagram the process rules and guidelines for managing the flow of work.
- Implement Feedback Loops. Throughout the work process, incorporate regular reviews with the team and customers to gather and incorporate feedback.
- Improve Collaboratively, Evolve Experimentally. As a team, look for and incorporate improvement initiatives, including through safe-to-fail experiments.



Project life cycle

The Project Life Cycle refers to a series of activities which are necessary to fulfill project goals or objectives.

Projects vary in size and complexity, but, no matter how large or small, all projects can be mapped to the following life cycle structure.



Identification Phase

This is the first stage of project life cycle. It is also known as initiation phase. The initiating processes determine the nature and scope of the project. If this stage is not performed well, it is unlikely that the project will be successful in meeting the business' needs. The key project controls needed here are an understanding of the business environment and making sure that all necessary controls are incorporated into the project. Any deficiencies should be reported and a recommendation should be made to fix them.

The initiating stage should include a plan that encompasses the following areas. These areas can be recorded in a series of documents called Project Initiation documents. Project Initiation documents are a series of planned documents used to create order for the duration of the project. These tend to include:

- project proposal (idea behind project, overall goal, duration)
- project scope (project direction and track)
- product breakdown structure (PBS) (a hierarchy of deliverables / outcomes and components thereof)

- work breakdown structure (WBS) (a hierarchy of the work to be done, down to daily tasks)
- responsibility assignment matrix (RACI) (roles and responsibilities aligned to deliverables / outcomes)
- tentative project schedule (milestones, important dates, deadlines)
- analysis of business needs and requirements against measurable goals
- review of the current operations
- financial analysis of the costs and benefits, including a budget
- stakeholder analysis, including users and support personnel for the project
- project charter including costs, tasks, deliverables, and schedules
- SWOT analysis: strengths, weaknesses, opportunities, and threats to the business.

Planning Phase

After the initiation stage, the project is planned to an appropriate level of detail. The main purpose is to plan time, cost, and resources adequately to estimate the work needed and to effectively manage risk during project execution.

Project planning generally consists of

- determining the project management methodology to follow
- developing the scope statement
- selecting the planning team
- identifying deliverables and creating the product and work breakdown structures
- identifying the activities needed to complete those deliverables and networking the activities in their logical sequence
- estimating the resource requirements for the activities
- estimating time and cost for activities
- developing the schedule
- developing the budget
- risk planning
- developing quality assurance measures
- gaining formal approval to begin work.

Implementation, Monitoring and Controlling Phase

During the third phase, the implementation phase, the project plan is put into motion and the work of the project is performed. It is important to maintain control and communicate as needed during implementation. Progress is continuously monitored and appropriate adjustments are made and recorded as variances from the original plan.

In any project, a project manager spends most of the time in this step. During project implementation, people are carrying out the tasks, and progress information is being reported through regular team meetings. The project manager uses this information to maintain control over the direction of the project by comparing the progress reports with the project plan to measure the performance of the project activities and take corrective action as needed. The first course of action should always be to bring the project back on course. If that can not happen, the team should record variations from the original plan and record and publish modifications to the plan. Throughout this step, project sponsors and other key stakeholders should be kept informed of the project's status according to the agreed-on frequency and format of communication. The plan should be updated and published on a regular basis.

Status reports should always emphasize the anticipated end point in terms of cost, schedule, and quality of deliverables. Each project deliverable produced should be reviewed for quality and measured against the acceptance criteria. Once all of the deliverables have been produced and the customer has accepted the final solution, the project is ready for closure.

Closure Phase

During the final closure, or completion phase, the emphasis is on releasing the final deliverables to the customer, handing over project documentation to the business, terminating supplier contracts, releasing project resources, and communicating the closure of the project to all stakeholders. The last remaining step is to conduct lessons-learned studies to examine what went well and what didn't. Through this type of analysis, the wisdom of experience is transferred back to the project organization, which will help future project teams.

Jira

Jira Software is a part of a family of products designed to help teams of all types manage work.

JIRA is a tool developed by Australian Company Atlassian. This software is used for bug tracking, issue tracking, and project management.

In order to report your progress, follow the plan, and respond to change, you will need robust Agile project management tools such as **user stories, epics, themes, and initiatives**.

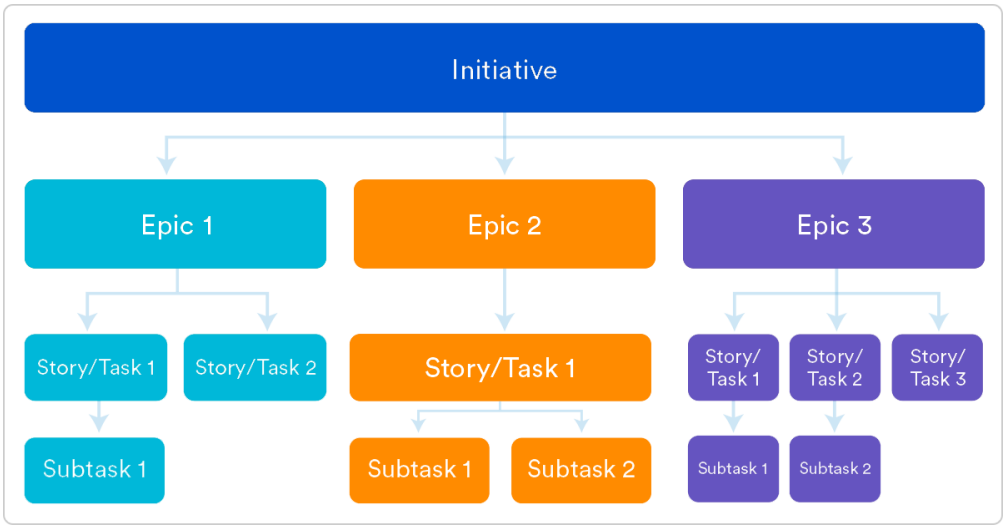
Themes are large focus areas that span the organization.

Initiatives are collections of epics that drive toward a common goal.

Epics are large bodies of work that can be broken down into a number of smaller tasks (called stories).

Stories, also called "user stories," are short requirements or requests written from the perspective of an end user.

Sub-Tasks are the sub-tasks of an issue.



Note:

VIDEO: Creating and Running an Agile Project in JIRA: Epics, Stories, Bugs, and Tasks

Chapter 8. ITCQF certification

International Technical Communication Qualifications Foundation is a certification program of technical communication which is based on international standards.

Training

ITCQF® Foundation Level training is a course which lasts for two days and can be held in class or online. It fully covers the content of the syllabus and supplements it with graphical summaries, practical examples and excercises.

ITCQF training

Exams

ITCQF® exams are organized in a consistent way worldwide and are based on a precisely defined number of questions, distributed in accordance to the syllabus topics.

ITCQF exams

Materials

There are 3 PDF documents that can be downloaded: ITCQF® Overview, The value of techwriting, and Syllabus (version 2.0)

ITCQF materials

Community

ITCQF® partnership model offers three basic types of cooperation – board, accredited training providers and certification bodies.

ITCQF community

Chapter 9. Glossary

Agile

A system of methods designed to allow the development team to match and track the business needs, especially in a context where business needs change frequently, important facts change, or where we are obliged to adapt to important uncontrolled factors.

Camtasia

A software suite, created and published by TechSmith, for creating video tutorials and presentations directly via screencast, or via a direct recording plug-in to Microsoft PowerPoint.

CAT

Computer-aided translation (CAT), also referred to as machine-assisted translation (MAT) or machine-aided human translation (MAHT), is the use of software to assist a human translator in the translation process. The translation is created by a human, and certain aspects of the process are facilitated by software; this is in contrast with machine translation (MT), in which the translation is created by a computer, optionally with some human intervention.

Confluence

A web-based corporate wiki (collaboration software) developed by Australian software company Atlassian.

Content strategy

A repeatable system that defines the entire editorial content development process for a website development project.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Customer journey map

A visual representation of the customer journey (also called the buyer journey or user journey). It tells the story of customers' experiences with a certain brand across all touchpoints.

Customer persona

A semi-fictional depiction of an ideal buyer based on market research and actual data about present consumers. It encompasses consumer demographics, behavior patterns, motivations, and objectives.

DDLC

Documentation Development Life Cycle (DDLC) is the process of developing the document that involves a systematic process that continues in cyclic order.

DITA

The Darwin Information Typing Architecture (DITA) specification defines a set of document types for authoring and organizing topic-oriented information, as well as a set of mechanisms for combining, extending, and constraining document types.

Git

Software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

GitHub

A provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features.

GitHub Pages

A static web hosting service offered by GitHub to GitHub users for hosting user blogs, project documentation, or even whole books created as a page.

HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

ITCQF

International Technical Communication Qualifications Foundation (ITCQF) is a comprehensive technical communication certification program developed in cooperation with recognized tech comm experts, and based on existing international standards.

JavaScript

One of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

Jekyll

a simple, blog-aware, static site generator for personal, project, or organization sites. Written in Ruby by Tom Preston-Werner, GitHub's co-founder, it is distributed under the open source MIT license.

Jira

A proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management.

Just-the-Docs

A modern, highly customizable, and responsive Jekyll theme for documentation with built-in search, easily hosted on GitHub Pages.

Kanban

A workflow management method for defining, managing and improving services that deliver knowledge work. It aims to visualize work, maximize efficiency, and improve continuously.

Localization

The process of adapting a product's translation to a specific country or region.

MadCap

A single-sourcing, topic-based authoring platform, which uses XML markup for maximum flexibility. MadCap Flare allows technical writers and documentation specialists to create, manage and publish content to a variety of languages and outputs.

Markdown

A lightweight markup language for creating formatted text using a plain-text editor, widely used in blogging, instant messaging, online forums, collaborative software, documentation pages, and readme files.

MemoQ

A proprietary computer-assisted translation software suite which provides translation memory, terminology, machine translation integration and reference information management in desktop, client/server and web application environments.

Oxygen

An all-purpose XML editor that supports a broad variety of XML-based technologies, including “transformation scenarios” that turn DITA input into HTML or PDF output.

PLC

Project life cycle (PLC) is a workflow of activities defined in the systematic ways to gain maximum benefits from a business project. PLC consists of four main phases through which the Project Manager and his team try to achieve the objectives that the project itself sets.

Project Management

The process of leading the work of a team to achieve goals and meet success criteria at a specified time. The primary challenge of project management is to achieve all of the project goals within the given constraints.

Raster image

A way to represent digital images. It can be created in a wide variety of formats, including the familiar .gif, .jpg, and .bmp. The image is represented in a series of bits of information that translate into pixels on the screen.

Scrum

A framework utilizing an agile mindset for developing, delivering, and sustaining complex products, with an initial emphasis on software development.

SME

A subject-matter expert (SME) is a person who is an authority in a particular area or topic.

Snagit

A screenshot program that captures video display and audio output, created and distributed by TechSmith

STE

Simplified Technical English (STE) is a controlled language developed to help second-language speakers of English to unambiguously understand technical manuals written in English.

Style guide

A set of standards for the writing, formatting and design of documents.

Technical communication

A practice-oriented field that emphasizes detailed, problem-solving tactics, such as analysis, research, design, and production to communicate intricate information effectively.

Technical writer

A professional information communicator whose task is to transfer information between two or more parties, through any medium that best facilitates the transfer and comprehension of the information.

Vector images

Graphical representations of mathematical objects such as lines, curves, polygons and its like. These graphics are generated by computer and they follow x and y axis as their reference definition.

Vistula

Akademia Finansów i Biznesu Vistula - a non-public academic higher education institution based in Warsaw.

Visual Studio Code

A freeware source-code editor made by Microsoft. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Waterfall

A breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialisation of tasks.

Web technologies

a general term referring to the many languages and multimedia packages that are used in conjunction with one another, to produce dynamic web sites such as this one.