{POWER.CODERS}

HTML best practice

Agenda

This class will be a combination of theory + practice

- Quiz
- History and terminology
- Project best practice
- Semantics and boilerplate
- Ressources and online material
- Exercises
- Appendix: More about domains

Quiz

What is HTML?

Quiz

What is HTML?

- **H**yper**T**ext **M**arkup **L**anguage

Quiz

What is HTML?

- **H**yper**T**ext **M**arkup **L**anguage
- HTML is composed of **tags** that together provide a **blueprint** for a webpage.

Quiz

What is HTML?

- **H**yper**T**ext **M**arkup **L**anguage
- HTML is composed of **tags** that together provide a **blueprint** for a webpage.
- **Hypertext** is a text enriched with hyperlinks.

Quiz

What is HTML?

- **H**yper**T**ext **M**arkup **L**anguage
- HTML is composed of **tags** that together provide a **blueprint** for a webpage.
- **Hypertext** is a text enriched with hyperlinks.
- **Markup language** uses tags to define the page layout and elements within the page. It is human-readable.

Quiz

What are a few different HTML tags?

Quiz

Which tag is used to create a link to another page?

1. `<l>`
2. `<link>`
3. `<a>`
4. `<p>`

Quiz

What is a container element compared to a stand alone element?

Quiz

What is a container element compared to a stand alone element?

Container Element

Quiz

What is a container element compared to a stand alone element?

Container Element

- contains other elements (nested elements) or content, a.g. a paragraph `<p>` contains text

Quiz

What is a container element compared to a stand alone element?

Container Element

- contains other elements (nested elements) or content, a.g. a paragraph `<p>` contains text

Stand Alone Element

Quiz

What is a container element compared to a stand alone element?

Container Element

- contains other elements (nested elements) or content, a.g. a paragraph `<p>` contains text

Stand Alone Element

- An element that cannot contain anything else, like `<br>` and `<img>`.

Quiz

What are the two tags that nest directly within the `<html>` tags?

Quiz

What are the two tags that nest directly within the `<html>` tags?

- `<head>`

Quiz

What are the two tags that nest directly within the `<html>` tags?

- `<head>`
- `<body>`

Quiz

What is a HTML comment?

Quiz

What is a HTML comment?

`<!-- Document Content -->` is only visible in the source code.

Quiz

What is a HTML comment?

`<!-- Document Content -->` is only visible in the source code.

Comments can be used to organize your code into sections so you (or someone else) can easily understand your code. It can also be used to 'comment out' large chunks of code to hide it from the browser.

Quiz

What is a relative path versus an absolute path?

Quiz

What is a relative path versus an absolute path?

If a file is part of the same web site, then a **relative URL** can be used. This can be only the name of the file.

Quiz

What is a relative path versus an absolute path?

If a file is part of the same web site, then a **relative URL** can be used. This can be only the name of the file.

If the file is located on another website, an **absolute URL** must be used. Absolute URLs contain the entire domain name and path.

Examples

```
<!-- Relative URLs -->
<a href="image-gallery.html">Image Gallery</a>
<a href="blog/first-blog-entry.html">My First Blog Entry</a>
<a href="../image-gallery.html">Back to Image Gallery</a>

<!-- Absolute URLs -->
<a href="http://www.my-colleague.com/blog.html">Blog of a Colleague</a>
```

Examples

```
<!-- Relative URLs -->
<a href="image-gallery.html">Image Gallery</a>
<a href="blog/first-blog-entry.html">My First Blog Entry</a>
<a href="../image-gallery.html">Back to Image Gallery</a>

<!-- Absolute URLs -->
<a href="http://www.my-colleague.com/blog.html">Blog of a Colleague</a>
```

- Inside the same folder we just use the filename, for example `portrait.jpg`.

Examples

```
<!-- Relative URLs -->
<a href="image-gallery.html">Image Gallery</a>
<a href="blog/first-blog-entry.html">My First Blog Entry</a>
<a href="../image-gallery.html">Back to Image Gallery</a>

<!-- Absolute URLs -->
<a href="http://www.my-colleague.com/blog.html">Blog of a Colleague</a>
```

- Inside the same folder we just use the filename, for example `portrait.jpg`.
- Two dots (`..`) refer to the parent directory.

Examples

```
<!-- Relative URLs -->
<a href="image-gallery.html">Image Gallery</a>
<a href="blog/first-blog-entry.html">My First Blog Entry</a>
<a href="../image-gallery.html">Back to Image Gallery</a>

<!-- Absolute URLs -->
<a href="http://www.my-colleague.com/blog.html">Blog of a Colleague</a>
```

- Inside the same folder we just use the filename, for example `portrait.jpg`.
- Two dots (`..`) refer to the parent directory.
- If we want to start in the root directory we add an `/` before the path of the file, for example `/portrait.jpg`.

Quiz

What does a complete link (anchor) element look like?

Quiz

What does a complete link (anchor) element look like?

```
<a href="http://google.com/" target="_blank">This goes to google</a>
```

Quiz

What does block-level and inline elements mean?

Quiz

What does block-level and inline elements mean?

Block-level elements

- start in a new line
- take up the full width of the page
- e.g. `<h1-6>`, `<p>`, `<br>`

Inline elements

- do not start in a new line
- only take up the necessary width
- e.g. `<a>`, `<em>`, `<strong>`

Quiz

What is an attribute? Explain and list some examples.

Quiz

What is an attribute? Explain and list some examples.

- An attribute provied additional information about the HTML element
- It is placed inside an opening tag, before the right ankle bracket
- Examples: class, id, style, src, href, ...

Quiz

What is a HTML entity?

Quiz

What is a HTML entity?

- **special** characters: like accent marks and German umlaut, e.g. `ü`
- **invisible** characters: like non-breaking spaces, e.g. .
- **reserved** characters: which would be interpreted as HTML code, e.g. `<`

Often used entities

- non-breakable space = ` `
- – = `&ndash;`
- — = `&mdash;`
- © = `&copy;`
- ¼ = `&frac14;`
- ½ = `&frac12;`
- ¾ = `&frac34;`
- « = `&laquo;`
- » = `&raquo;`
- ä = `&auml;`
- Ü = `&Uuml;`
- é = `&eacute;`
- è = `&egrave;`
- ← = `&leftarrow;`
- ↑ = `&uparrow;`
- → = `&rightarrow;`
- ↓ = `&downarrow;`

More entities

History and terminology

History of HTML

- Invented by Tim Berners-Lee

History of HTML

- Invented by Tim Berners-Lee
- First web page published August 6, 1991

History of HTML

- Invented by Tim Berners-Lee
- First web page published August 6, 1991
- Created "hypertext" to share scientific papers

History of HTML

- Invented by Tim Berners-Lee
- First web page published August 6, 1991
- Created "hypertext" to share scientific papers
- Intended as a standard way to structuring documents

History of HTML

- Invented by Tim Berners-Lee
- First web page published August 6, 1991
- Created "hypertext" to share scientific papers
- Intended as a standard way to structuring documents
- Standardized by W3C (World Wide Web Consortium)

History of HTML

- Invented by Tim Berners-Lee
- First web page published August 6, 1991
- Created "hypertext" to share scientific papers
- Intended as a standard way to structuring documents
- Standardized by W3C (World Wide Web Consortium)
- HTML 5 in 2014

Terms

- **Web design:** The process of planning, structuring and creating a website

Terms

- **Web design:** The process of planning, structuring and creating a website
- **Web development:** The process of programming dynamic web applications

Terms

- **Web design:** The process of planning, structuring and creating a website
- **Web development:** The process of programming dynamic web applications
- **Front end:** The outwardly visible elements of a website or application

Terms

- **Web design:** The process of planning, structuring and creating a website
- **Web development:** The process of programming dynamic web applications
- **Front end:** The outwardly visible elements of a website or application
- **Back end:** The inner workings and functionality of a website or application.

Project best practice

Defintion of best practice

Defintion of best practice

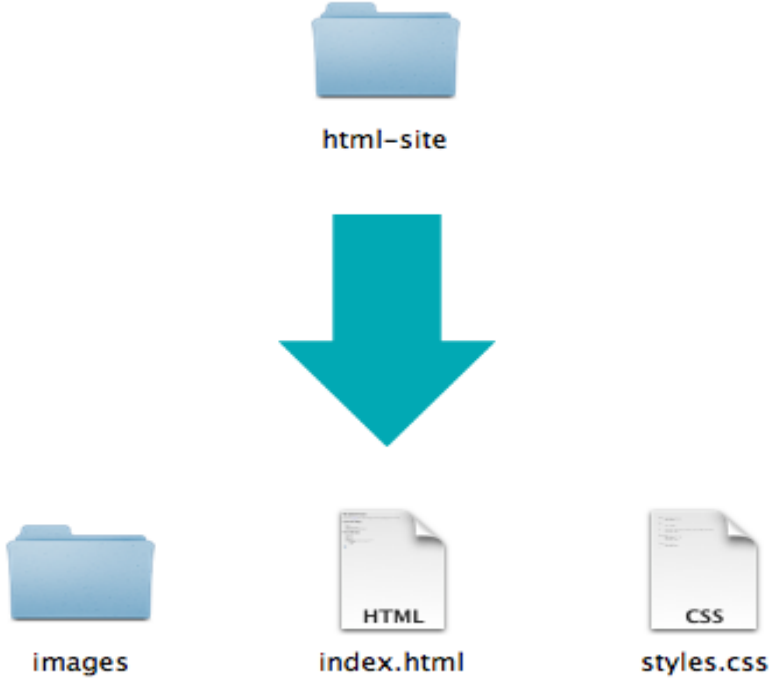
A **method** or **technique** that has been **generally accepted as superior** to any alternatives because it produces results that are superior to those achieved by other means or because it has become **a standard way of doing things**.

Folder Structure

All the files for your site should be stored within the same folder.

This includes:

- HTML Files
- CSS Files
- Images
- Script files
- Anything else that will appear on your site



Note: File names should not include spaces or special characters. File names ARE case sensitive.

Naming Folders + Files

- Name your file `index.html`
- In file and folder names, only use lowercase letters, numbers, hyphens/dashes.
- File names are usually case sensitive: `INDEX.html` vs. `index.html`
- Use the right extension: `.html` vs `.css` vs `.js`

Tips + Shortcuts

After each opening tag, the next element should be indented with a `tab` for better overview. Make sure you follow this habit.

Tips + Shortcuts

After each opening tag, the next element should be indented with a `tab` for better overview. Make sure you follow this habit.

Action	Windows	Mac
VSCode, save	`Ctrl` + `s`	`⌘` + `s`
VSCode, undo	`Ctrl` + `z`	`⌘` + `z`
Switch apps	`Alt` + `Tab`	`⌘` + `Tab`
Chrome, reload	`Ctrl` + `r`	`⌘` + `r`

A WELL DEVELOPED WEBSITE

A WELL DEVELOPED WEBSITE

- has **valid** code

A WELL DEVELOPED WEBSITE

- has **valid** code
- is **fast** and **performant**

A WELL DEVELOPED WEBSITE

- has **valid** code
- is **fast** and **performant**
- is **searchable** by humans and bots

A WELL DEVELOPED WEBSITE

- has **valid** code
- is **fast** and **performant**
- is **searchable** by humans and bots
- is **accessible** for disabled people

Anatomy of a Website

Your Content
+ HTML: Structure
+ CSS: Presentation
= Your Website

SEMANTICS

Semantic Markup

Definition semantics:

Semantic Markup

Definition semantics:

The meaning of a word, phrase or text.

Semantic Markup

Definition semantics:

The meaning of a word, phrase or text.

Definition semantic web:

Semantic Markup

Definition semantics:

The meaning of a word, phrase or text.

Definition semantic web:

Information is given a well-defined meaning, better enabling computers and people to work in cooperation.

An example

The quick brown fox jumps over the lazy dog.

An example

The quick brown fox jumps over the lazy dog.

- = English-language pangram (contains all letters of the English alphabet)

An example

The quick brown fox jumps over the lazy dog.

- = English-language pangram (contains all letters of the English alphabet)
- Can you see it in your mind?

Now imagine

The quick brown thing jumps over the lazy thing.

Now imagine

The quick brown thing jumps over the lazy thing.

What do you see?

Now imagine

The quick brown thing jumps over the lazy thing.

What do you see?

Every time you use a `div` or `span`, you tell the browser: **"There's a thing"**.

Now imagine

The quick brown thing jumps over the lazy thing.

What do you see?

Every time you use a `div` or `span`, you tell the browser: **"There's a thing"**.

And the browser asks, **"What kind of thing?"**

Now imagine

The quick brown thing jumps over the lazy thing.

What do you see?

Every time you use a `div` or `span`, you tell the browser: **"There's a thing"**.

And the browser asks, **"What kind of thing?"**

Non-graphical browsers and bots do not see how the website looks like. They need the tags to know the meaning of the content.

Advantages of semantics

Advantages of semantics

- easier to explain

Advantages of semantics

- easier to explain
- easier to understand

Advantages of semantics

- easier to explain
- easier to understand
- faster to process

Advantages of semantic web

Advantages of semantic web

- more searchable content

Advantages of semantic web

- more searchable content
- better search engine ranking

Advantages of semantic web

- more searchable content
- better search engine ranking
- less clutter of meaningless HTML

Advantages of semantic web

- more searchable content
- better search engine ranking
- less clutter of meaningless HTML
- less errors in source code and more performant

Advantages of semantic web

- more searchable content
- better search engine ranking
- less clutter of meaningless HTML
- less errors in source code and more performant
- better accessible for assistive technologies, like screen readers

Quick word to history

When you google **semantic web**, you fill find the term often in relation with **HTML 5**.

Quick word to history

When you google **semantic web**, you fill find the term often in relation with **HTML 5**.

A lot of new semantic tags were introduced in 2014 with HTML 5, but HTML was from the beginning a **language with semantic tags and meaning**.

Semantic HTML

Always use the tags best describing the content:

Semantic HTML

Always use the tags best describing the content:

- `<h1>` for headlines
- `<h2-6>` for sub headlines
- `<ul>` for unordered lists
- `<ol>` for ordered lists
- `<table>` for complex data

Semantic Block-level tags

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website
- `<main>` for the main and most important content of a page

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website
- `<main>` for the main and most important content of a page
- `<section>` for thematic grouping of content, e.g. news, team

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website
- `<main>` for the main and most important content of a page
- `<section>` for thematic grouping of content, e.g. news, team
- `<article>` is independent, self-contained content, e.g. news entry, blog entry

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website
- `<main>` for the main and most important content of a page
- `<section>` for thematic grouping of content, e.g. news, team
- `<article>` is independent, self-contained content, e.g. news entry, blog entry
- `<aside>` for less important content, e.g. in a sidebar or overlay

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website
- `<main>` for the main and most important content of a page
- `<section>` for thematic grouping of content, e.g. news, team
- `<article>` is independent, self-contained content, e.g. news entry, blog entry
- `<aside>` for less important content, e.g. in a sidebar or overlay
- `<footer>` for information at the end of document, e.g. copyright, address, author, legal info

Semantic Block-level tags

- `<header>` for introduction of the page, e.g. logo, page headline, content summary
- `<nav>` as container for navigational elements, links to access content within the website
- `<main>` for the main and most important content of a page
- `<section>` for thematic grouping of content, e.g. news, team
- `<article>` is independent, self-contained content, e.g. news entry, blog entry
- `<aside>` for less important content, e.g. in a sidebar or overlay
- `<footer>` for information at the end of document, e.g. copyright, address, author, legal info

Full list on MDN

Headlines

<h>-tags can be within different block-level elements, e.g. header, main, article, section, footer.
Use hierarchical order of headlines, e.g.:

Headlines

`<h>`-tags can be within different block-level elements, e.g. header, main, article, section, footer.
Use hierarchical order of headlines, e.g.:

- `<h1>` in the `<header>`
- `<h2>` in the `<section>`
- `<h3>` in the `<article>`

Headlines

`<h>`-tags can be within different block-level elements, e.g. header, main, article, section, footer.
Use hierarchical order of headlines, e.g.:

- `<h1>` in the `<header>`
- `<h2>` in the `<section>`
- `<h3>` in the `<article>`

Each article should contain at least 1 headline.

Each page should only have 1 `<h1>`-tag.

Section in articles

You can group articles in a section, e.g. news items within the section **news**.

Section in articles

You can group articles in a section, e.g. news items within the section **news**.

You can also have several sections within one article, if needed.

Section in articles

You can group articles in a section, e.g. news items within the section **news**.

You can also have several sections within one article, if needed.

Can you think of an example?

Exercise

Go to 2 websites of your choice and write the markup of the homepage.

- Which tags do you use? Why?
- Check your buddy's markup and discuss the choice of tags.

Exercise

Go to 2 websites of your choice and write the markup of the homepage.

- Which tags do you use? Why?
- Check your buddy's markup and discuss the choice of tags.

Done? Now, check the source code of the website.

Exercise

Go to 2 websites of your choice and write the markup of the homepage.

- Which tags do you use? Why?
- Check your buddy's markup and discuss the choice of tags.

Done? Now, check the source code of the website.

- What do you see?
- Why?

BOILERPLATE

What is a boilerplate?

The word **"boilerplate"** means standardized pieces of text for use as clauses in contracts or as part of a computer program.

An HTML boilerplate will contain the most common elements of a page as a sample that can be cloned and used as a starting point for a project.

HTML BOILERPLATE

Let's code our own boilerplate.
What should it contain?

One possibility

One possibility

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Boilerplate</title>
  </head>
  <body>
    <header>
      <h1></h1>
      <nav>
      </nav>
    </header>
    <main>
      <p></p>
    </main>
```

Examples of HTML boilerplates

- html5boilerplate.com
- htmlshell.com
- sitepoint.com
- initializr.com

Examples of HTML boilerplates

- html5boilerplate.com
- htmlshell.com
- sitepoint.com
- initializr.com

Check them out. Which one do you prefer? Why?

Ressources and online material

- Handout: HTML reference sheet
- When should alt tags be blank?
- HTML outliner
- W3C validator

Ressources and online material

- Why you should choose `article` over `section`
- HTML 5 Doctor
- Semantics in HTML 5
- An introduction to the semantic web
- The ADA checklist 2019

EXERCISES

Exercise 1

Create a website to show off your new skills. Core HTML tags you should be able to include on your page(s):

1. A *nav bar* with a few links that either navigate to other pages or act as anchor tags.
2. A *heading* to signal something important or declare a new section/paragraph.
3. A couple *paragraphs* describing the amazing things you want to share. Remember to use *emphasis* on key words!
4. A *list* of key things to know about something in your paragraph. Why not do an ordered list and unordered list?
5. An *image* of something relevant.
6. A short contact *form*
7. A *table* to display additional contact data
8. A *footer* at the bottom of the page declaring your copyright on such a wonderful write-up, and make sure you use the copyright *symbol!*
9. Your page should validate in this at this link: W3 Validator.
10. Keep your code readabable and maintanable for your future self and future contributors.

Remember: all your exercises go to Github. Use git for regular commits to your github repo.

Exercise 2

Draft the semantic outline of a newspaper page in HTML: tagesanzeiger.ch

- Which semantic tags would you use to describe the homepage?
- Create a HTML outline for a subpage
- Your page should validate in this at W3C Validator.
- Check your document outline and web semantics with HTML5 Outliner.
- Keep your code readabable and maintanable for your future self and future contributors.

Remember: all your exercises go to Github. Use git for regular commits to your github repo.

Exercise 3

Start with the HTML skeleton of your project website

- Be semantic
- Use comments to explain why you chose specific tags and keep it readable and maintanable
- Your page should validate in this at W3C Validator.
- Check your document outline and web semantics with HTML5 Outliner.

Appendix: More about domains

Domain

Anatomy of domain names:
subdomain.domain.topleveldomain

Domain

Anatomy of domain names:
subdomain.domain.topleveldomain

- powercoders.org
- www.gmail.com
- calendar.google.com

TOPLEVELDOMAIN (TLD)

The most common TLDs are:

TOPLEVELDOMAIN (TLD)

The most common TLDs are:

- `.com`: "Commercial entities"
- `.net`: "Networking companies"
- `.org`: "Non-profits"

TOPLEVELDOMAIN (TLD)

The most common TLDs are:

- `.com`: "Commercial entities"
- `.net`: "Networking companies"
- `.org`: "Non-profits"

As the use of these TLDs isn't restricted, they are used more flexibly than as originally intended.

Country-code TLDs

Each country has its own TLD (ccTLD), and they can decide who can register for domains with that TLD. Some also specify a set of "second-level-domain" for the TLDs.

Country-code TLDs

Each country has its own TLD (ccTLD), and they can decide who can register for domains with that TLD. Some also specify a set of "second-level-domain" for the TLDs.

- `.ch`
- `.fr`
- `.it`
- `.de`
- `.uk` / `.co.uk`
- ...

New gTLDs Program

Since 2013 new generic TLDs were introduced.
Over a thousand new gTLDs:

New gTLDs Program

Since 2013 new generic TLDs were introduced.
Over a thousand new gTLDs:

- thematic: `.hotel` `.bank`
- geographic: `.florida` `.berlin`
- product-related: `.toys` `.yoga`
- specific: `.tech` `.design`
- individual: `.codes` `.ninja`
- ...

What TLD should you get?

Most users expect to type in a `.com` or `.ch` domain, so you should always try to reserve that name (though it's often taken).

What TLD should you get?

Most users expect to type in a `.com` or `.ch` domain, so you should always try to reserve that name (though it's often taken).

If you're worried about competitors, you should purchase related TLDs (.info, .net, .biz).

What TLD should you get?

Most users expect to type in a `.com` or `.ch` domain, so you should always try to reserve that name (though it's often taken).

If you're worried about competitors, you should purchase related TLDs (.info, .net, .biz).

If you target a particular geographic market, it makes sense to get the ccTLD.

What TLD should you get?

Most users expect to type in a `.com` or `.ch` domain, so you should always try to reserve that name (though it's often taken).

If you're worried about competitors, you should purchase related TLDs (.info, .net, .biz).

If you target a particular geographic market, it makes sense to get the ccTLD.

For your own portfolio one of new gTLDs might be best, e.g. `.codes` or `.dev`.

Domain names

The domain name must follow these rules:

Domain names

The domain name must follow these rules:

- No spaces or underscores
- Dashes and numbers are OK

Domain names

The domain name must follow these rules:

- No spaces or underscores
- Dashes and numbers are OK

The domain name is not case-sensitive - google.com = GOOGLE.com.

How to choose a domain name?

A good domain name follows these guidelines:

How to choose a domain name?

A good domain name follows these guidelines:

- Represents your name or business name
- Short and memorable
- Easy to say aloud
- Easy to spell without mistakes
- Uses ASCII characters
- Doesn't infringe other's copyright

How to choose a domain name?

A good domain name follows these guidelines:

- Represents your name or business name
- Short and memorable
- Easy to say aloud
- Easy to spell without mistakes
- Uses ASCII characters
- Doesn't infringe other's copyright

But good domain names are often taken. You can try using hyphens (experts-exchange.com) or making a TLD form part of the name (del.icio.us) to make it more likely the name will be available.

Subdomain

Once you own a domain name, you can make subdomains for different aspects of your product/company, e.g.

Subdomain

Once you own a domain name, you can make subdomains for different aspects of your product/company, e.g.

- www.twitter.com
- search.twitter.com
- api.twitter.com

Subdomain

Once you own a domain name, you can make subdomains for different aspects of your product/company, e.g.

- www.twitter.com
- search.twitter.com
- api.twitter.com

The www-Subdomain is the most common. If you type in a domain, e.g. google.com, it automatically refers to the www-subdomain. Try it yourself.

Domain registrars

You must use a "domain registrar" to purchase a domain.
Just google "register domain" to find a registrar.

Domain registrars

You must use a "domain registrar" to purchase a domain. Just google "register domain" to find a registrar.

Worldwide the most popular domain registrar is GoDaddy.

Domain registrars

You must use a "domain registrar" to purchase a domain. Just google "register domain" to find a registrar.

Worldwide the most popular domain registrar is GoDaddy.

I register my domains with Metanet.