Frontendᵝ → HTML → Basic Tags → Links

Theory: Links

🕐 33 minutes   0 / 5 problems solved

[Skip this topic] [**Start practicing**]

World Wide Web would not be called a web if it wasn't tightly woven by "threads" that connect one web page to another. **Hyperlinks** or just links play the role of threads here. We can follow links within a certain page to see its different sections, switch between several pages of a website, or even go from one website to a completely different one. All of this works thanks to the `<a>` tag that is used to create links.

You already know a little about the `<a>` tag from previous topics. In this topic, we will get into more detail.

§1. Syntax

Suppose we want to add a link to JetBrains Academy to our web page. We could just copy the site's URL and paste it as plain text, but it wouldn't look very presentable. It would be much better to use a special tag:

```
1    <a href="https://hyperskill.org/">This is a link to JetBrains Academy</a>
```

Let's take a closer look at this example. We wrapped some text using a starting tag `<a>` and a closing tag `</a>`. The letter `a` stands for "anchor". The text in the tags describes our link in the context and will be clickable, colored, and underlined by default in the browser. The URL goes in the `href` attribute in the starting tag.

Without any CSS styling, here is what links look like by default: unvisited links are blue, visited links are purple, active links (those that are clicked on) are red, and all of them are underlined. Of course, many of the links' properties can be changed with CSS. For example:

HTML:

```
1    <a href="https://hyperskill.org"> JetBrains Academy </a>
2    <br />
3    <a href="https://hyperskill.org" class="styled-
link"> JetBrains Academy, styled link </a>
```

CSS:

```
1    .styled-link {
2        font-size: 20px;
3        color: green;
4    }
```

The results will be the following:

JetBrains Academy

JetBrains Academy, styled link

The second link looks different, but it still functions as it's supposed to.

The `<a>` tag is an inline element. If we put several `<a>` tags one after another, they will be placed on the same line. When it's necessary to avoid it, you may use `<br />`.

§2. Absolute and relative addressing

URLs are used in `<a>` as addresses of linked pages. They have two forms: absolute and relative. Absolute form is a full URL, and relative form includes the path and parts of the URL that follow. Both can be used as a value of `href`, but they're used differently.

Here is an example of an absolute URL as `href`:

Current topic:

Links   ⋯

Topic depends on:

✕ HTML URL [Stage 1] ⋯

✕ Tags and attributes [Stage 1] ⋯

✕ Inline Elements [Stage 1] ⋯

Table of contents:

```
1    <a href="https://hyperskill.org">JetBrains Academy</a>
```

Usually, absolute addressing is appropriate when you want to direct the user to another site, in other words, to create an **external link**.

If a link takes the user from one page of the website to another page on the same site, in other words, if it's an **internal link**, it's better to use a relative URL:

```
1
<a href="tracks"> Link from https://hyperskill.org to https://hyperskill.org/track
s </a>
2
3
<a href="5"> Link from https://hyperskill.org/tracks to https://hyperskill.org/tra
cks/5 </a>
4
5
<a href="tracks/5"> Link from https://hyperskill.org to https://hyperskill.org/tra
cks/5 </a>
6
7
<a href="../projects"> Link from https://hyperskill.org/tracks to https://hyperski
ll.org/projects </a>
```

You may ask, why are relative URLs more preferable for internal links than absolute ones? Let's imagine someone developed a big project http://mycoolsite.com. It contains hundreds of pages, and every internal link in this project looks like this:

```
1    <a href="http://mycoolsite.com/[path]">...</a>
```

However, if they suddenly decided to change the site from http://mycoolsite.com to https://myreallycoolsite.com, they would have to rewrite every internal link on their site for it to work properly. As you can imagine, it's a huge amount of work that could have been avoided if a different type of addressing had been applied. If relative URLs had been used, such a problem would not have appeared because the changed part of the links (protocol and domain) doesn't participate in `href`. Keep it in mind when working on a real project!

§3. Page navigation

Another thing that we can do with `<a>` is navigation within a page. For navigation, we use links to the elements of the page that have the specified IDs. To create it, we need to fill the `href` attribute with the `#` and `id` of the target element:

```
1
<section id="wise">The wise man built his house on stone. Then a great flood came
there, and winds blew there, and fell down upon the house, and it did not fall: tr
uly, it was built on stone.</section>
2    <div style="height: 1000px;"></div>
3
<section id="foolish">Then the foolish man built his house on sand. Then it rained
, and a flood came there, and winds blew, and fell down upon the house, and the ho
use fell; and its fall was great.</section>
4    <div style="height: 1000px;"></div>
5    <nav>
6        <a href="#wise">Go to the wise man</a>
7        <br />
8        <a href="#foolish">Go to the foolish man</a>
9    </nav>
```

Try to insert this piece of code into some page and click on the links. If everything works well, the first link should scroll to the first section and the second link should take you to the second section.

Note that `div`'s are inserted here to make this particular example more descriptive and are not common practice. Usually, there is a lot of content between navigation points.

This same feature is also useful when you need to create a link to a specific part of another page. To do this, you need to fill `href` with the page link and add the `#` and `id` like this:

```
1 │
<a href="https://en.wikipedia.org/wiki/United_Kingdom#History">Go to the history o
f United Kingdom on Wikipedia</a>
```

§4. Attributes

We can create links that lead straight to the files that users will view in their browser and/or download. If you want the linked file to start downloading without opening in the browser, add the `download` attribute to the tag `<a>`:

```
1 │
<a href="https://somesite.com/somefile.pdf" download> Somefile.pdf will be downloa
ded! </a>
```

You can also specify the file name that the file will be saved as on the user's computer:

```
1 │
<a href="https://somesite.com/somefile.pdf" download="book"> Somefile.pdf will be
downloaded as book.pdf! </a>
```

The `target` attribute manages where the linked page will be opened. There are several options:

- `_self` is the default option: it opens the link in the current browsing context, which most often implies the current tab;
- `_blank` opens the page in a new tab.

There are other options to choose from: check the [documentation at developer.mozilla.org](#) for more.

Sometimes links have tooltips that appear when the cursor hovers over the link. If you want to clarify the purpose or the contents of the link this way, use the `title` attribute:

```
1 │
<a href="https://google.com" title="Go to google.com"> Let's google it! </a>
```

§5. Special links

Suppose we want to add a phone number and an email address to our page. We can write it as simple text, but a better choice would be to create active contacts with `<a>`. When the user clicks the link, they will be offered to call or to send an email using default applications:
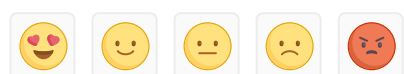
```
1     <a href="tel:+1234567890"> +1234567890 </a>
2
3     <a href="mailto:example@mail.com"> example@mail.com </a>
```

§6. Conclusion

In this topic, we considered different aspects of the `<a>` tag in more detail. Now you know how to create and style hyperlinks, choose between relative and absolute addressing, and use hyperlinks to create page navigation, contacts, and more. Let's move on to some practice tasks!

🗎 Report a typo

3 users liked this theory. **0** didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

[Start practicing]

This content was created 21 days ago and updated 12 days ago. Share your feedback below in comments to help us improve it!

Comments (0)   Hints (0)   Useful links (0)   Show discussion

This content was created 21 days ago and updated 12 days ago. Share your feedback below in comments to help us improve it!

Comments (0)   Hints (0)   Useful links (0)   Show discussion