

# Základy tvorby www stránek

---

## Co se v modulu dozvíte?

- jak funguje značkovací jazyk HTML
- jaká je základní struktura dokumentu
- jak propojit HTML stránku se styly
- jaká jsou pravidla pro strukturování dokumentu
- které jsou nejčastější tagy
- jak používat absolutní a relativní odkazy

## Z čeho se skládá webová stránka

Pro tvorbu www stránek budeme potřebovat na začátku dva základní typy souborů:

- **HTML soubory**, ve kterých je uložen věcný obsah stránek v logické struktuře (tj. je rozlišeno, co je hlavní nadpis, co je podnadpis, co je buňka tabulky nebo co je položka seznamu) a
- **CSS soubory** (kaskádové styly), které obsahují formátování dokumentu.

Od začátku tvorby stránek je žádoucí tyto dvě části (obsah a vzhled) nesměšovat.

Později přidáme ještě další typy souborů, například média (obrázky, videa, zvuk) nebo skripty (malé kousky kódu, které říkají, co se má při nějaké události stát, např. zobrazit doposud skrytý prvek), pomocí kterých dosáhneme skutečné interaktivity stránek. Pro dynamické stránky (tj. stránky, které se např. generují podle parametrů nebo zobrazují údaje z databáze) budeme později používat skriptovací jazyk na straně serveru.

## Jednoduchý příklad:

HTML dokument - soubor `prvni_soubor.html` »

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Moje první stránka</title>
6  </head>
7  <body>
8      <h1>Nadpis stránky</h1>
9      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a
10     diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam
11     viverra nec consectetur ante hendrerit. Donec et mollis dolor. Praesent et
12     diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue
13     enim, ut porta lorem lacinia consectetur. Donec ut libero sed arcu
14     vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur
15     adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a
16     semper sed, adipiscing id dolor. Pellentesque auctor nisi id magna
17     consequat sagittis. Curabitur dapibus enim sit amet elit pharetra
18     tincidunt feugiat nisl imperdiet. Ut convallis libero in urna ultrices
19     accumsan. Donec sed odio eros. Donec viverra mi quis quam pulvinar at
20     malesuada arcu rhoncus. Cum sociis natoque penatibus et magnis dis
21     parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies.
22     Mauris vitae nisi at sem facilisis semper ac in est.</p>
23     </body>
     </html>
```

Když si stránku otevřete, zobrazí se jen nadpis ("Nadpis stránky") a odstavec textu. Pro odstavec jsme použili generický text Lorem ipsum. Používá se proto, aby text vizuálně vypadal podobně jako běžný text.

Na stránce si všimněte několika vlastností:

- Text je ohraničen tzv. **tagy**, to jsou ty texty mezi znaky `<` a `>`. Tagy popisují, co je jimi ohraničeno. Tomuto způsobu strukturování dokumentu se obecně říká **značkovací jazyky**, HTML, jak z názvu vyplývá - HyperText Markup Language - je jedním ze značkovacích jazyků. Dalším příkladem je XML - eXtended Markup Language.
- Rozlišujeme otevírací a uzavírací tag, přičemž uzavírací se liší jen lomítkem, např. `<h1>` a `</h1>` - viz **řádek 8**. Název tagu pochází z anglického **heading** doplněného o úroveň nadpisu, v tomto případě 1 (nadpis první úrovně, tj. hlavní). Kdyby se text rozpadal do dvou kapitol, do stránky bychom vložili dvakrát nadpis druhé úrovně - `h2`.
- Na **řádce 1** je definován tzv. DOCTYPE, kterým říkáme, podle jakého standardu HTML se budeme řídit. Zápis `<!DOCTYPE html>` určuje jako standard HTML ve verzi 5.
- Na **řádce 2** začíná samotné HTML. Všimněte si, že celá stránka je na **řádce 23** uzavřena pomocí `</html>`
- Stejným způsobem je definován nadpis na **řádce 8**: text uzavřený mezi tagy `<h1>` a `</h1>` je nadpis. Totéž na dalších řádcích platí pro odstavec (anglicky **paragraph**): `<p>` a `</p>`.

Aniž bychom používali jakékoli formátování, text je naformátovaný pomocí stylů, které jsou vestavěné v prohlížeči. Když si stejnou stránku zobrazíte v různých prohlížečích, zjistíte, že se bude v detailech lišit. Každý prohlížeč sice respektuje základní pravidla, např. že nadpisy jsou větší a výraznější, ale liší se např. nastavením okrajů kolem textu.

## Terminologie

- **značkovací jazyk** - způsob, jak do prostého textu vnést pomocí značek strukturu. V případě HTML se značky jmenují **tagy**.
- **tag** - značka, která říká "tady začíná/končí konkrétní prvek", např. značka `<p>` říká "tady začíná odstavec". Rozlišujeme otevírací a uzavírací tagy - "tady to začíná", "tady to stejně končí". Liší se lomítkem, které se píše jen do uzavíracího tagu - např. `<h1>` (začátek nadpisu první úrovně) a `</h1>` (konec nadpisu první úrovně).
- **element** - celý prvek ohraničený otevíracím a uzavíracím tagem. V příkladu výše je to např. element `h1`: tohle celé `<h1>Nadpis stránky</h1>` je jeden element. Elementy se do sebe mohou vnořovat, jako je třeba `h1` a `p` vnořeno do elementu `body`.
- **atribut** - tag může být doplněn různými vlastnostmi, kterým se říká atributy. Častým atributem je třída (`class`), podrobnější rozlišení obecnějšího prvku. U odkazů (`a`) je atributem URL adresa, na kterou se přejde po kliknutí.

## Příklad

V příkladu vidíte ukázkou odkazu: tag `a` (z anglického "anchor", kotva) je doplněn atributem `href`, který říká, kam vede odkaz. Mezi tagy `<a>` a `</a>` je napsán text, na který bude možné kliknout a dostat se tak na stránky `gml.vse.cz`.

```
1 <a href="http://gml.vse.cz/">stránka GML</a>
```

## Doplnění vzhledu - první CSS

V tomto modulu se zatím nebudeme věnovat formátování a ukážeme si ještě další prvky HTML, kterými lze určit **strukturu dokumentu**. Než se do toho pustíme, ukážeme si jen velmi jednoduché propojení HTML a CSS na stejném příkladu, který už známe.

### Příklad: doplnění odkazu na CSS

V příkladu je zvýrazněn řádek s doplněním odkazu na soubor, kde se nacházejí **definice vzhledu jednotlivých prvků**, viz soubor `prvni_soubor_css.html` »

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Moje první stránka</title>
6   <link rel="stylesheet" href="styl.css" type="text/css" />
7 </head>
8 <body>
9   <h1>Nadpis stránky</h1>
10  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam
11 lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec
12 consectetur ante hendrerit. Donec et mollis dolor. Praesent et diam eget libero
13 egetas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem
14 lacinia consectetur. Donec ut libero sed arcu vehicula ultricies a non tortor.
15 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida
16 lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.
17 Pellentesque auctor nisi id magna consequat sagittis. Curabitur dapibus enim
18 sit amet elit pharetra tincidunt feugiat nisl imperdiet. Ut convallis libero in
19 urna ultrices accumsan. Donec sed odio eros. Donec viverra mi quis quam
20 pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et magnis dis
21 parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris
22 vitae nisi at sem facilisis semper ac in est.</p>
23 </body>
24 </html>
```

A když jsme definovali odkaz na soubor s určením stylů, samozřejmě musíme vytvořit i tento soubor (`styl.css`).

```
1 body {
2   background-color: #005f3a;
3   color: white;
4   width: 800px;
5   margin-left: auto;
6   margin-right: auto;
7   font-family: Georgia, 'Times New Roman', serif;
8 }
9 h1 {
10  color: #ffa;
```

```

10 }
11
12 p {
13     font-family: Helvetica, Arial, sans-serif;
14 }
15
16

```

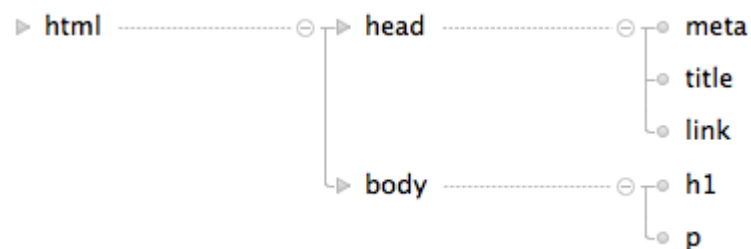
Co se oproti prvnímu případu změnilo: v HTML souboru jen jeden řádek, který říká, odkud se má brát definice vzhledu dokumentu, resp. jednotlivých jeho prvků. A máme nový soubor, ve kterém jsou tyto vlastnosti popsány. Jednotlivé vlastnosti není třeba teď podrobně vysvětlovat, většina z nich je zřejmých. Důkladně se jimi budeme zabývat v modulu věnovaném CSS.

Díky oddělení vzhledu (CSS) od struktury dokumentu (HTML) můžete snadno stanovit, jak budou vypadat např. **všechny nadpisy** nebo **všechny odstavce**. Později uvidíme, že lze velmi přesně zacílit konkrétní část HTML dokumentu a tu formátovat.

## Zpět k HTML

### Struktura dokumentu

Důležitým aspektem HTML dokumentu je fakt, že celý dokument je možné rozkreslit do podoby "stromu". Později tuto vlastnost výhodně využijeme jak ve formátování pomocí kaskádových stylů, tak u zacílení konkrétního prvku pomocí skriptování na straně klienta. Náš jednoduchý příklad bychom mohli překreslit takto:



Aby bylo možné takto HTML dokument překreslit, **nesmíme křížit tagy**.

### Příklad - chybně

```
1 <strong>tučný text <em>a tučná kurzíva</strong></em>
```

### Příklad - správně

```
1 <strong>tučný text <em>a tučná kurzíva</em></strong>
```

Pro lepší pochopení si zkusme vytvořit propracovanější HTML dokument a překreslit ho opět do stromové struktury. Jednotlivé tagy budou vysvětleny později.

```

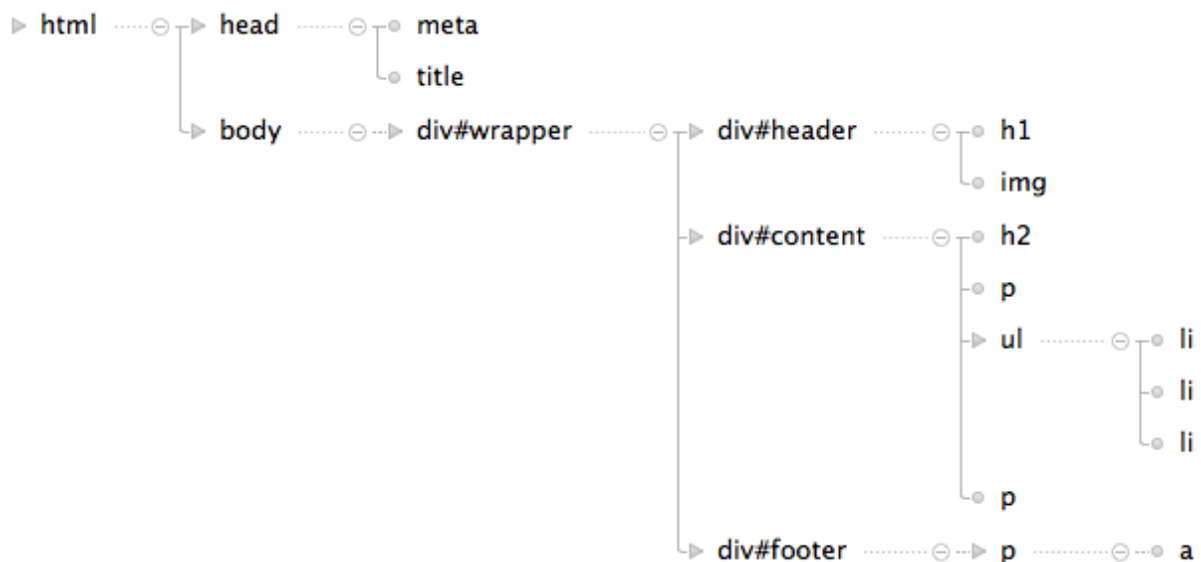
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Struktura HTML dokumentu - ukázka</title>
6 </head>
7 <body>
8     <div id="wrapper">
9         <div id="header">

```

```

11         <h1>Nadpis</h1>
12         
13         <p>Odstavec se <strong>zvýrazněným</strong> textem</p>
14     </div>
15     <div id="content">
16         <h2>Nadpis druhé úrovně</h2>
17         <p>Text odstavce</p>
18         <ul>
19             <li>první odrážka</li>
20             <li>druhá odrážka</li>
21             <li>třetí odrážka</li>
22         </ul>
23         <p>A ještě jeden odstavec.</p>
24     </div>
25     <div id="footer">
26         <p>Obsah patičky s <a href="jiny_soubor.html">odkazem</a></p>
27     </div>
28 </div>
29 </body>
</html>

```



## Používání mezer

Všimněte si ještě jedné důležité vlastnosti HTML: **používání mezer v kódu**. Protože HTML je značkovací jazyk, kde jednotlivé prvky určujeme pomocí značek, není důležité, kolik mezer a kam umístíme. Proto můžeme dokument strukturovat tak, aby byl přehledný - odsazením vnořených prvků nebo přidáním mezer tam, kde to je potřebné. Pamatujte si: **více mezer se interpretuje vždy jako jedna mezer**.

Přepíšeme-li náš poslední příklad do "minimalizované" podoby, dostaneme výsledek, který se v prohlžeči bude chovat stejně jako přehlednější příklad výše. Jen je daleko méně přehledný.

```

1 <!DOCTYPE html><html><head><meta charset="UTF-8"><title>Struktura HTML
2 dokumentu - ukázka</title></head><body><div id="wrapper"><div
3 id="header"><h1>Nadpis</h1><p>Odstavec se
4 <strong>zvýrazněným</strong> textem</p></div><div id="content"><h2>Nadpis druhé
5 úrovně</h2><p>Text odstavce</p><ul><li>první odrážka</li><li>druhá

```

```

6   odrážka</li><li>třetí odrážka</li></ul><p>A ještě jeden odstavec.</p></div><div
7   id="footer"><p>Obsah patičky s <a
8   href="jiny_soubor.html">odkazem</a></p></div></div></body></html>

```

## Tagy

Tagy jsou obvykle odvozeny z významu, který mají nést, např. `p` jako paragraph, odstavec. Nejprve si probereme ty, které píšeme do samotného textu a které se ve stránce zobrazují, tj. mezi tagy `<body>a</body>`, a později se budeme zabývat těmi méně viditelnými, jako jsou např. meta informace v hlavičce (`head`) stránky.

tag	anglicky	význam
<code>body</code>		tělo dokumentu - vše, co se má na stránce zobrazit, se píše mezi <code>&lt;body&gt;</code> a <code>&lt;/body&gt;</code>
<code>h1</code>	heading	nadpis první úrovně
<code>h2, h3, h4, h5, h6</code>		nadpisy dalších úrovní (pozor: číslování <code>h1, h2</code> atd. je podobné jako číslování stylů nadpisů ve Wordu)
<code>p</code>	paragraph	odstavec
<code>img</code>	image	obrázek
<code>div</code>	division	logicky oddělená část dokumentu (sekce, oddíl), např. patička
<code>ul</code>	unordered list	nečíslovaný seznam (pozor na seznamy: prvkem <code>ul</code> definujete seznam, ale položky seznamu mají svůj tag)
<code>ol</code>	ordered list	číslovaný seznam
<code>li</code>	list item	položka seznamu
<code>a</code>	anchor	odkaz, kotva
<code>span</code>		část textu označená pro další formátování pomocí CSS - pouhé použití tagu <code>span</code> se v dokumentu nijak neprojeví
<code>strong</code>		zvýraznění - obvykle se zobrazuje tučně
<code>em</code>	emphasis	jiné zvýraznění - obvykle se ukáže jako kurzíva
<code>br</code>	break	zalomení řádku (používá se např. při psaní adresy)
<code>table</code>		tabulka (opět pozor: stejně jako u seznamů nejprve definujeme tabulku a pak teprve její části)
<code>tr</code>	table row	řádek tabulky
<code>th</code>	table heading	buňka v záhlaví tabulky
<code>td</code>	table data	vlastní buňka

Přehled tagů samozřejmě není kompletní. Cílem bylo ukázat, že dokument je možné **významově** rozčlenit pomocí několika málo tagů. Zkuste si projít některé `www` stránky, které používáte, a podívejte se na ně očima HTML: kde byste použili které tagy? Uvidíte, že většinou vystačíte s tímto stručným výběrem.

Většina uvedených tagů dává textu, který ohraničují, nějaký význam, např. "toto je položka seznamu". Dva se ale vymykají: `div` a `span`.

- `div` odděluje logickou sekci dokumentu, jak je to ukázáno v příkladu, lze si ho představit jako (neviditelný) obdélník kolem několika prvků, například hlavičky dokumentu. Sám nedělá nic, resp. chová se jako **blokový** prvek, jako by byl od předchozího a následujícího obsahu oddělen odřádkováním. V HTML5 je tento prvek rozšířen tak, aby nesl význam a vznikly tak nové

tagy `article`, `section`, `figure`, `nav`, `header`, `footer`, `aside` (`div` lze v HTML5 stále použít). Přestože mají všechny tyto prvky jasné pojmenování, určují význam, nikoli vzhled.

- `span` opět odděluje kousek dokumentu, rozdíl oproti tagu `div` je ten, že jde o **řádkový** prvek, tj. funguje podobně jako např. `strong` - neudělá odřádkování před a za sebou. Využívá se pro vyznačení logické části textu, např. když chcete jinak formátovat kousek textu a nevystačíte s prvky `strong` a `em`

## Atributy

U některých tagů nevystačíme se samotným tagem, ale potřebujeme mu přidat různé vlastnosti. Nejlépe je to vidět u obrázku. Tag `img` říká "tady je obrázek". Potřebujeme doplnit, jaký obrázek.

```
1 
```

### Všimněte si způsobu zápisu:

- Atribut má vždy **název** a **hodnotu**. Zápis je `název="hodnota"`.
- Obsahuje-li tag více atributů, pak jsou oddělené **mezerou**.
- Názvy atributů jsou obvykle zkratkou anglického slova vystihujícího podstatu atributu:
- `src` z anglického "source" je zdroj obrázku, tj. kde se nachází obrázek, který chceme do dokumentu vložit
- `alt` z anglického "alternative" obsahuje popis obrázku, alternativní text, který se zobrazí, když např. v prohlížeči vypnete zobrazování obrázků nebo při využití programů, které čtou text ze stránek
- Protože u obrázku nemá smysl mluvit o "otevřacím" a "uzavíracím" tagu, je zakončen `/>`, čímž sám sebe uzavírá. Stejně funguje odsazení řádku (zapiše se `<br />`)

**Pozor:** některé atributy jsou povinné. Například pro obrázek je povinný atribut `src` a `alt`, viz [specifikace W3C](#). Pro každý tag je určeno, které atributy může obsahovat, tj. které atributy dávají pro daný tag smysl. Tři atributy se mohou vyskytovat naprosto u všech prvků, a to jsou `id`, `class` a `style`.

### Jiný příklad, tentokrát odkaz:

```
1 <a href="http://www.vse.cz/">stránky VŠE</a>
```

Tady máme jen jeden atribut: `href` (z anglického "hypertext reference"), který ukazuje, kam přejít po kliknutí. Druhou částí odkazu je samotný text, na který uživatel klikne; ten se nevkládá do atributu, ale je mezi tagy `<a>` a `</a>`.

## Absolutní a relativní URL

Vraťme se ještě k příkladu s odkazem. Po kliknutí na text [stránky VŠE](#) se otevře stránka specifikovaná v atributu `href`. Tyto odkazy, resp. hodnoty parametru `href` mohou být

- **absolutní**, jak je uvedeno v příkladu, nebo
- **relativní**, tj. jsou určeny relativní pozicí vůči dokumentu, ze kterého se odkazujeme.

**URL** je zkratka pro "Uniform Resource Locator", a je to jednoznačná identifikace objektu v rámci internetu.

### Absolutní odkazy

Absolutní odkazy poznáte snadno - stačí si všimnout, jestli mají všechny části. A povinné části jsou tři:

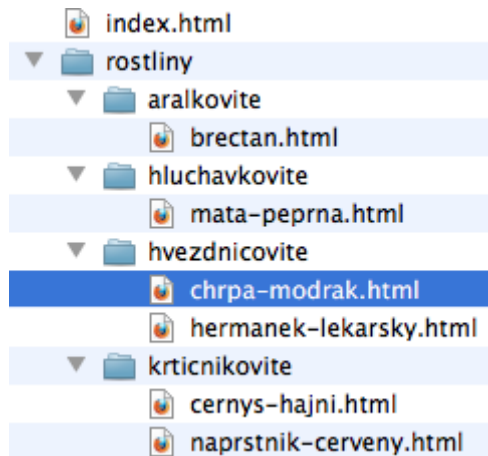
- protokol (např. `http://`)
- server (např. `www.vse.cz`) a
- soubor nebo adresář na serveru (např. `/`)

## Relativní odkazy

Relativní odkazy ukazují na dokument ve vztahu k dokumentu, ze kterého se odkazují. Kromě obvyklých názvů adresářů a souborů je třeba respektovat čtyři pravidla:

- adresáře se značí **běžným lomítkem (/)**, nikoli zpětným
- na aktuální adresář se odkážeme tečkou (**.**)
- na nadřazený adresář se odkážeme dvěma tečkami (**..**)
- nejvyšší adresář webu má označení **/**

**Příklad:** upravujeme soubor chrpa-modrak.html (na obrázku zvýrazněn) a dáváme do něj odkazy na různé další HTML soubory na našem webu.



1. Odkaz ze souboru `chrpa-modrak.html` na soubor `hermanek-lekarsky.html` využije pouze jména souborů, protože jsou oba soubory ve stejném adresáři:

```
1 Přečtete si také text o <a href="hermanek-lekarsky.html">heřmánku</a>
```

Totožný zápis by byl s využitím odkazu na aktuální adresář (**.**):

```
1 Přečtete si také text o <a href="./hermanek-lekarsky.html">heřmánku</a>
```

Mohli bychom se také odkázat přes kořenový adresář webu, což je ale v tomto případě zbytečně složité:

```
1 Přečtete si také text o <a href="/rostliny/hvezdnicovite/hermanek-lekarsky.html">heřmánku</a>
```

2. Odkaz ze souboru `chrpa-modrak.html` na soubor `brectan.html` je složitější - musíme o adresář výš, tj. **..** (to jsme v adresáři `rostliny`), poté do adresáře `aralkovite` a odkazujeme se na soubor `brectan.html`. Složeno dohromady: `../aralkovite/brectan.html`:

```
1 Další odkaz vede na <a href="../aralkovite/brectan.html">břečtan</a>
```

Stejně jako v předchozím příkladu bychom se mohli odkázat přes kořenový adresář (**/**) webu:

```
1 Další odkaz vede na <a href="/rostliny/aralkovite/brectan.html">břečtan</a>
```

3. Odkaz ze souboru `chrpa-modrak.html` na titulní stránku, tj. soubor `index.html`

- jsme v adresáři `hvezdnicovite`,
- odtud jdeme pomocí dvou teček (**..**) o adresář výš, tím se dostáváme do adresáře `rostliny`
- potřebujeme ještě o adresář výš, tj. další **..**



- to už jsme v hlavním adresáři webu, tj. už jen dopíšeme jméno souboru (`index.html`)

1 Zpět na `<a href="../../index.html">hlavní stránku</a>`

V tomto příkladu by bylo snazší odkázat se přes kořenový adresář webu:

1 Zpět na `<a href="/index.html">hlavní stránku</a>`