



FERIT

Fakultet elektrotehnike, računarstva
i informacijskih tehnologija Osijek

Web programiranje

LV1: HTML i HTML obrasci

Contents

1	Uvod	2
2	Kategorije sadržaja HTML-a	2
2.1	Glavne kategorije sadržaja	2
2.1.1	Sadržaj metapodataka	2
2.1.2	Sadržaj toka dokumenta	3
2.1.3	Sadržaj odjeljenja	3
2.1.4	Naslovni sadržaj	3
2.1.5	Sadržaj fražiranja	3
2.1.6	Ugrađeni sadržaj	3
2.1.7	Interaktivni sadržaj	4
3	HTML semantika	4
3.1	HTML5 Elementi	4
3.1.1	<section> element	4
3.1.2	<article> element	4
3.1.3	<aside> element	5
3.1.4	<header> element	5
3.1.5	<footer> element	5
3.1.6	<nav> element	6
3.1.7	<figure> element	6
3.2	Prilagodba za čitače zaslona i ugnježdivanje elemenata naslova	6
4	HTML atributi	8
4.1	ARIA(Accessible Rich Internet Applications) atributi	9
5	HTML obrasci	10
5.1	Osnovni elementi i atributi HTML obrasca	10
5.2	Elementi kontrole unosa HTML obrasca	11
6	Predučitavanje sadržaja	14
7	Slike i dodatni sadržaj	16

1 Uvod

Hypertext Markup Language 5 (HTML 5) predstavlja najnoviji standard koji definira HTML. Prikazuje najnoviju verziju HTML jezika, s novim elementima, atributima i načinima ponašanja te skup tehnologija koji omogućava izradu raznovrsnijih i tehnički boljih web stranica i aplikacija. Taj se skup tehnologija ponekad se naziva HTML5 & friends i često se skraćuje na naziv HTML5 [3].

HTML5 tehnologije moguće je razvrstati u određene grupe prema njihovim funkcijama:

1. **Semantika:** Omogućava jasnije i preciznije opisivanja nekog sadržaja
2. **Povezivanje:** Omogućava komunikaciju s poslužiteljem na nove i inovativnije načine
3. **Izvanmrežno pohranjivanje:** Omogućava web stranicama lokalno pohranjivanje podataka na strani klijenta i učinkovitije korištenje izvanmrežne stranice
4. **Multimedija:** Izrada i reproduciranje video i audio materijala na webu
5. **Učinkovitost i integracija:** Naprednija optimizacija i bolje iskorištavanje računalnog hardware
6. **Pristup uređajima:** Omogućava upotrebu različitih uređaja za ulaz izlaz
7. **Stiliziranje:** Dopušta programerima kreiranje sofisticiranih tema i stilova

2 Kategorije sadržaja HTML-a

Svaki HTML element član je jedne ili više kategorije sadržaja - kategorije grupiraju elemente koje imaju zajedničke karakteristike. Ovo grupiranje nema čvrstu vezu (ne stvara se odnos između elemenata u grupi), ali pomaže definirati zajednička ponašanja i pravila koja se nalaze u kategoriju. Postoji mogućnost da pojedini element ne mora biti član niti jedne od navedenih kategorija.

Postoje tri vrste kategorizacije sadržaja:

1. **Glavne kategorije sadržaja** koje opisuju uobičajena pravila koja dijele mnogi drugi elementi.
2. **Kategorije sadržaja povezane s obrascima**, koja opisuju pravila zajednička za elemente koji su povezani s obrascima i akcijama vezanim za obrasce.
3. **Specifične kategorije sadržaja**, opisuje rijetke kategorije koje se sastoje od malog broja elemenata koje se koriste samo u specifičnom kontekstu.

2.1 Glavne kategorije sadržaja

Glavne kategorije sadržaja opisane su prema slici 1.

2.1.1 Sadržaj metapodataka

Elementi mogu mijenjati prezentaciju ili ponašanje html dokumenta, postavljaju veze na druge dokumente te mogu prenositi podatke izvan pojasa dokumenta. Elementi koji pripadaju ovoj kategoriji su:

<base>, <command>, <link>, <meta>, <noscript>, <script>, <style> i <title>.

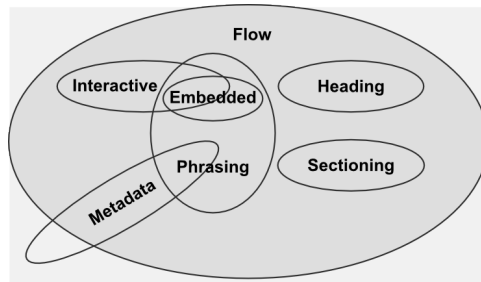


Figure 1: HTML glavne kategorije sadržaja

2.1.2 Sadržaj toka dokumenta

Elementi koji pripadaju kategoriji toka dokumenta obično sadrže tekst ili ugrađeni sadržaj (audio ili video). Elementi koji pripadaju toj kategoriji su:

`<a>`, `<abbr>`, `<address>`, `<article>`, `<aside>`, `<audio>`, ``, `<bdo>`, `<bdi>`, `<blockquote>`, `
`, `<button>`, `<canvas>`, `<cite>`, `<code>`, `<command>`, `<data>`, `<datalist>`, ``, `<details>`, `<dfn>`, `<div>`, `<dl>`, ``, `<embed>`, `<fieldset>`, `<figure>`, `<footer>`, `<form>`, `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, `<header>`, `<hgroup>`, `<hr>`, `<i>`, `<iframe>`, ``, `<input>`, `<ins>`, `<kbd>`, `<keygen>`, `<label>`, `<main>`, `<map>`, `<mark>`, `<math>`, `<menu>`, `<meter>`, `<nav>`, `<noscript>`, `<object>`, ``, `<output>`, `<p>`, `<picture>`, `<pre>`, `<progress>`, `<q>`, `<ruby>`, `<s>`, `<samp>`, `<script>`, `<section>`, `<select>`, `<small>`, ``, ``, `<sub>`, `<sup>`, `<svg>`, `<table>`, `<template>`, `<textarea>`, `<time>`, ``, `<var>`, `<video>`, `<wbr>` i tekst.

2.1.3 Sadržaj odjeljenja

Elementi koji pripadaju modelu sadržaja odjeljenja stvaraju semantičke cijeline u trenutnom dokumentu. Elementi koji pripadaju ovoj kategoriji su:

`<header>`, `<footer>`, `<article>`, `<aside>`, `<nav>` i `<section>`.

2.1.4 Naslovni sadržaj

Naslovni sadržaj definira naslov odjeljka, bilo označen eksplicitnim elementom (h1-h6) sadržaja odjeljka, ili implicitno definiran samim sadržajem naslova. Elementi koji pripadaju ovoj kategoriji su:

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` i `<hgroup>`.

2.1.5 Sadržaj fražiranja

Sadržaj fraze definira tekst i oznaku koju sadrži. Postupci fražiranja sadržaja čine odlomke. Elementi koji pripadaju ovoj kategoriji su:

`<abbr>`, `<audio>`, ``, `<bdo>`, `
`, `<button>`, `<canvas>`, `<cite>`, `<code>`, `<command>`, `<data>`, `<datalist>`, `<dfn>`, ``, `<embed>`, `<i>`, `<iframe>`, ``, `<input>`, `<kbd>`, `<keygen>`, `<label>`, `<mark>`, `<math>`, `<meter>`, `<noscript>`, `<object>`, `<output>`, `<picture>`, `<progress>`, `<q>`, `<ruby>`, `<samp>`, `<script>`, `<select>`, `<small>`, ``, ``, `<sub>`, `<sup>`, `<svg>`, `<textarea>`, `<time>`, `<var>`, `<video>`, `<wbr>` i tekst.

2.1.6 Ugrađeni sadržaj

Ugrađeni sadržaj uvodi resurse ili sadržaje iz druge okoline. Elementi koji pripadaju ovoj kategoriji su:

`<audio>`, `<canvas>`, `<embed>`, `<iframe>`, ``, `<math>`, `<object>`, `<picture>`, `<svg>`, `<video>`.

2.1.7 Interaktivni sadržaj

Interaktivni sadržaj uključuje elemente koji su posebno dizajnirani za interakciju s korisnicima. Elementi koji pripadaju ovoj kategoriji uključuju:

`<a>`, `<button>`, `<details>`, `<embed>`, `<iframe>`, `<keygen>`, `<label>`, `<select>`, and `<textarea>`.

3 HTML semantika

U ranijim verzijama HTML-a nazivi strukturnih elemenata nisu bili jasno definirani. Programeri su se oslanjali na dva strukturna elementa: `<div>` i ``. Zbog manjka semantičkih elemenata, tražilicama je bilo teško pravilno indeksirati ciljani sadržaj web stranica. Tražilica prilikom svog indeksiranja "crawl-a" HTML strukturu i traži određene informacije koje su joj bitne za indeksiranje [1].

HTML5 uveo je popis semantičkih elemenata kao i pravila prilikom strukturiranja elemenata. Poštujući definirana pravila, web preglednici, čitači zaslona i tražilice lakše tumače značenje određenih sadržaja, umjesto da ga samo prikazuju. Osim preglednika, programerima je lakše čitati strukturni kod koji se drži definiranih pravila. Na slici 2 prikazani su najčešće korišteni strukturni elementi prilikom oblikovanja web stranice.

3.1 HTML5 Elementi

3.1.1 `<section>` element

Prema W3c-u **section** element predstavlja okvir u kojemu se nalazi sadržaj sa definiranim naslovom (obično s H1). Web stranica može sadržavati jedan ili više sectiona [7].

```
<section>
  <h1>Section Naslov</h1>
  <p>Ovo je section paragraf.</p>
  
</section>
```

Kod 1: Primjer section elementa

3.1.2 `<article>` element

Article predstavlja element koji određuje neovisni, samostalni sadržaj od ostale cijeline. Moguće je ugnježdavati article elemente unutar section elemenata i obratno neovisno o poredku.

```
<section>
  <h1>Section Naslov</h1>
  <p>Ovo je section paragraf.</p>
  

  <article>
    <h1>Article naslov</h1>
    <p>Ovo je article paragraf</p>
  </article>
</section>
```

```
<article>
  <h1>Article naslov</h1>
  <p>Ovo je article paragraf</p>
</article>
```

Kod 2: Primjer article elementa

3.1.3 <aside> element

Element aside definira sadržaj koji može ali i ne mora pratiti sadržaj koji je definiran unutar u glavnom dijelu stranice. U većini slučajeva aside element ima H4 naslov u svojoj strukturi.

```
<aside>
<h4>Aside h4 naslov</h4>
<p>Aside h4 paragraf</p>
</aside>
```

Kod 3: Primjer aside elementa

3.1.4 <header> element

Header element predstavlja zaglavlje dokumenta. Potrebno ga je koristiti kao okvir za uvodni sadržaj web stranice ili aplikacije. Kao i u prošleme primjeru, header element se može koristiti više puta (Primjer kod 4).

```
<header>
  <h1>Naslov zaglavlja</h1>
  <h3>Podnaslov H3</h3>
  <p>Paragraf unutar header elementa</p>
</header>

<article>
  <header>
    <h1>Naslov zaglavlja unutar article</h1>
    <p>Paragraf unutar header elementa u article</p>
  </header>
</article>
```

Kod 4: Primjer header elementa

3.1.5 <footer> element

Footer predstavlja element u kojemu se sumira generalni sadržaj stranice i podatke o autorskim pravima, poveznice na uvjete korištenja, podatke za kontakt itd.

```
<footer>
  <p>Kontakt informacije:
    <a href="mailto:someone@example.com">
```

```

    kontakt@kontakt.com
  </a>.
</p>
</footer>

<article>
  <h1>Naslov zaglavlja unutar article</h1>
  <p>Ovo je paragraf</p>
  <footer>
    <p>Kontakt informacije: <a href="mailto:someone@example.com">
kontakt@kontakt.com</a>.</p>
  </footer>
</article>

```

Kod 5: Primjer footer elementa

3.1.6 <nav> element

Nav element prikazuje prostor koji je namjenjen glavne poveznice na web stranici.

```

<nav>
  <a href="/html/">HTML</a>
  <a href="/css/">CSS</a>
  <a href="/js/">JavaScript</a>
</nav>

```

Kod 6: Primjer nav elementa

3.1.7 <figure> element

Figure element grupira sliku i opis u jedinstveni okvir. Svrha elementa je da opiše vizualno značenje prikazane slike.

```

<figure>
  
  <figcaption>Fig1. - Ferit.</figcaption>
</figure>

```

Kod 7: Primjer figure elementa

Ostatak preostalih HTML5 semantičkih elemenata nalazi se na referenci [10]

3.2 Prilagodba za čitače zaslona i ugnježđivanje elemenata naslova

Elementi <h1> - <h6> predstavljaju šest razina naslova odjeljaka unutar HTML-a. <h1> predstavlja najveću razinu odjeljaka, dok se <h6> smatra najnižom razinom.

Generalno pravilo glasi da je potrebno imati samo jedan <h1> po stranici. Upotreba višestrukih <h1> nije greška, ali se ne smatra najboljom praksom. Razlog leži u tome da "crawler" pretraživača prvi <h1> orđeduje kao primaran i daje mu važnost. <h1> naslov predstavlja najvažniji naslov na stranici koja govori koja je svrha navedene stranice te ima jako veliki utjecaj na optimizaciju pretraživača i relevantnosti web stranice prilikom korisničke pretrage.

Prilikom razvoja web stranice potrebno je obratiti pažnju i prilagoditi pregled web stranice osobama s poteškoćama. Takve osobe najčešće koriste softver koji čita zaslon prozora i audio putem govori korisniku gdje se nalazi. Unutar web preglednika takav softver skače iz naslova u naslov kako bi što brže obradio i generirao korisniku sadržaj stranice. Zbog toga je vrlo važno da ne preskače jednu ili više razina naslova. Ovo može stvoriti zbrku, jer se osoba koja se kreće ovim putem može zapitati gdje je naslov koji nedostaje (kod 8).

```
Primjer lose prakse strukturiranja
<h1>Ovo je naslov</h1>
<h3>Ovo je h2 podnaslov</h3>
<h2>Ovo je h3 podnaslov</h2>
Primjer dobrog strukturiranja
<h1>Ovo je naslov</h1>
<h2>Ovo je h3 podnaslov</h2>
<h3>Ovo je h2 podnaslov</h3>
```

Kod 8: Primjer ugnježdživanje naslova

Kako bi se održala organizacija sadržaja stranice, naslovi se mogu ugnjezditi kao podnaslovi. Većina čitača zaslona može generirati popis svih naslova i podnaslova na stranici što osobi s poteškoćama može brzo odrediti hijerarhiju sadržaja web stranice.

- h1 Naslov
 - h2 Podnaslov
 - h2 Podnaslov 2
 - h2 Podnaslov 3
 - * h3 Podnaslov
 - * h3 Podnaslov 2
 - * h3 Podnaslov 3
 - h2 Podnaslov 4
 - * h3 Podnaslov
 - h4 Podnaslov
 - * h3 Podnaslov 2

Još jedna uobičajena tehnika navigacije za korisnike čitača zaslona je generiranje popisa dijelova odjeljaka i njegovo korištenje za određivanje izgleda stranice. Sadržaj odjeljka može se označiti HTML ARIA(Accessible Rich Internet Applications) atributima aria-labelledby i id atributom, s time da aria atribut sažeto opisuje svrhu navedenog odjeljka. Navedena tehnika korisna je u situacijama kada na istoj stranici postoji više elemenata odjeljaka (kod 9).


```

<header>
  <nav aria-labelledby="primarna-navigacija">
    <h2 id="primarna-navigacija">Primarna navigacija</h2>
    <!-- sadrzaj navigacije -->
  </nav>
</header>
<!-- sadrzaj -->
<footer>
  <nav aria-labelledby="footer-navigacija">
    <h2 id="footer-navigacija">Footer Navigacija</h2>
    <!-- sadrzaj navigacije -->
  </nav>
</footer>

```

Kod 9: Primjer korištenja aria-labelledby i id atributa za odvajanje odjeljaka navigacije

U ovom primjeru tehnologija čitanja na zaslonu objavila bi da postoje dva <nav> odjeljka, jedan koji se zove "Primarna navigacija" i jedan koji se zove "Footer navigacija". Ukoliko aria i id atributi nisu postavljeni osoba koja koristi softver za čitanje na ekranu morati će istražiti sadržaj svakog nav elementa kako bi odredila njihovu svrhu.

4 HTML atributi

HTML atributi predstavljaju poseban set oznaka za kontrolu ponašanja određenog HTML elementa. Drugim riječima, možemo ih definirati kao modifikatore HTML elemenata. Atributi mogu modificirati ili pružiti određene funkcionalnosti HTML elementu bez kojih on ne može normalno funkcionirati [4]. Postoje nekoliko tipova atributa:

- Obavezni atributi - bez njih HTML element ne može normalno funkcionirati
- Opcionalni atributi - koriste se za izmjenu osnovnih funkcionalnosti nekog elementa
- Standardni atributi - zajednički atributi koji se nalaze u više HTML elemenata
- Događajni atributi - uzrokuju pokretanje skripti u određenim uvjetima

Kod 10 prikazuje HTML oznaku za referenciranje poveznica unutar i izvan web stranice s pripadajućim obaveznim atributom koji definira URL poveznice, te opcionalni atribut koji predstavlja generičku informaciju čitačima zaslona da je navedeni element poveznica.

```

<a href="https://www.favicode.net/jobs.html" role="link">Eksterni link</a>

```

Kod 10: Primjer HTML elementa s navedenim atributima

Popis svih atributa moguće je pronaći na poveznici [9].

4.1 ARIA(Accessible Rich Internet Applications) atributi

HTML5 elementi imaju definiranu semantiku, ali postoje slučajevi kada izgled i struktura HTML-a neće biti dovoljna kako bi pokrili sve slučajeve korisničkog sučelja prilagođenog čitačima zaslona. U tu svrhu koriste se ARIA atributi. ARIA atributi rješavaju probleme s prilagodbom web stranice za navedene uređaje koji se ne mogu upravljati izvornim HTML-om. ARIA atributi omogućuju prevođenje elementa na specifičan način koji je prepoznat od strane čitača zaslona prilikom generiranja strukture sadržaja web stranice [2].

Kod 11 sadrži element iz liste koji predstavlja prilagođeni potvrdni okvir (checkbox). Klasa checkbox određuje elementu vizualnu karakteristiku koja prikazuje korisniku da se radi o potvrdnom okviru.

```
<li tabindex="0" class="checkbox checked">  
  Zelim se preplatiti na newsletter.  
</li>
```

Kod 11: Primjer elementa unutar liste koji predstavlja prilagođeni potvrdni okvir

Iako korisnik neće imati problema s navedenom funkcionalnosti elementa, čitač zaslona ne zna da navedeni element treba biti potvrdni okvir i samim time postoji mogućnost da korisnik s invaliditetom ne vidi navedenu funkcionalnost. Kako bi ispravili navedeni problem, pomoću ARIA atributa elementu možemo dati podatke koji nedostaju kako bi ih čitač zaslona pravilno tumačio (kod 12). Navedeni atributi role i aria-checked eksplicitno definiraju identitet elementa kao potvrdni okvir. Navedeni element bit će dodan u stablo pristupačnosti i bit će generiran unutar strukture sadržaja web stranice od strane čitača zaslona.

```
<li tabindex="0" class="checkbox" role="checkbox" checked aria-checked="true">  
  Zelim se preplatiti na newsletter.  
</li>
```

Kod 12: Primjer elementa unutar liste koji predstavlja prilagođeni potvrdni okvir

ARIA atributi ne mijenjaju ponašanje HTML elemenata ili njegove funkcionalnosti. ARIA atribut može izmijeniti postojeću semantiku ili dodati semantiku onim elementima gdje izvorna semantika ne postoji. ARIA pruža nekoliko mehanizama za dodavanje semantičkih oznaka i opisa elementima. Jedan od ARIA mehanizama predstavlja dodavanje pristupačnog teksta pomoću definiranog opisnog atributa. Dva najčešće upotrebljena opisna ARIA atributa su:

- aria-label
- aria-labelledby

aria-label atribut označava vrijednost koja će koristiti kao semantički izraz čitaču zaslona. Navedeni atribut nadvladava bilo koji izvorni mehanizam elementa za definiranje njegove semantike. aria-label koristi se za opisivanje vizualne svrhe određenog HTML elementa. Primjer - gumb koji koristi grafički prikaz umjesto teksta koji ga opisuje, te ga je potrebno opisati svima koji ne mogu pristupiti vizualnoj indikaciji navedenog elementa (kod 13).

```
<button aria-label="Close" onclick="myDialog.close()" class="button button-menu">X</button>
```

Kod 13: Primjer korištenja aria-label atributa

aria-labelledby uspostavlja odnose između objekata i njihovih oznaka, a njegova vrijednost treba biti jedan ili više ID-ova elemenata, koji se odnose na elemente koji sadrže tekst potreban za označavanje. U navedenom primjeru, skočni izbornik povezan je s njegovom oznakom pomoću atributa **aria-labelledby** (kod 14).

```
<div role="menubar">
  <div role="menuitem" aria-haspopup="true" id="fileMenu">File</div>
  <div role="menu" aria-labelledby="fileMenu">
    <div role="menuitem">Open</div>
    <div role="menuitem">Save</div>
    <div role="menuitem">Save as ...</div>
    ...
  </div>
  ...
</div>
```

Kod 14: Primjer korištenja aria-label atributa

Popis svih ARIA atributa moguće je pronaći na poveznici [8].

5 HTML obrasci

HTML obrasci su fleksibilni elementi a ujedno predstavljaju jednu od najsloženijih struktura u HTML-u. Moguće je kreirati bilo koju vrstu obrasca koristeći samo native HTML elemente i attribute. Korištenjem ispravne strukture HTML-a prilikom kreiranja obrasca osiguravamo da je obrazac upotrebljiv i dostupan.

5.1 Osnovni elementi i atributi HTML obrasca

Element `<form>` predstavlja glavni element unutar kreiranja obrasca. Formalno definira oblik i attribute koji određuju ponašanje obrasca. Unutar `<form>` elementa potrebno je definirati sadržaj obrasca.

Element `<fieldset>` prikladan je način za definiranje grupa unutar obrasca koji imaju istu svrhu, iz semantičkog razloga. Moguće je dodati element `<legend>` odmah ispod početne oznake `<fieldset>` kako bi formalno opisali sadržaj koji se nalazi unutar pojedine `<fieldset>` grupe.

Element `<label>` formalni je način definiranja opisa pojedinih elemenata HTML obrasca. Ako se pravilno implementiraju, čitači zaslona izgovarat će oznaku elementa obrasca s bilo kojim definiranim uputama vezano za pojedini element.

Primjer osnovne strukture HTML obrasca nalazi se u kodu 15.

```
<form>
  <fieldset>
```

```

<legend>Velicina zadanih slova</legend>
<p>
  <input type="radio" name="size" id="size_1" value="small">
  <label for="size_1">Mala</label>
</p>
<p>
  <input type="radio" name="size" id="size_2" value="medium">
  <label for="size_2">Srednja</label>
</p>
<p>
  <input type="radio" name="size" id="size_3" value="large">
  <label for="size_3">Large</label>
</p>
</fieldset>
</form>

```

Kod 15: Osnovna forma HTML obrasca

Najčešće korišteni atributi prilikom kreiranja formi, definiraju se unutar <form> elementa (kod 16):

- **action** - skripta ili set naredbi na server strani koja će izvršiti nad proslijeđenim podacima
- **method** - Metoda koja se koristi za prijenos podataka. Najčešće se koriste metode GET i POST.
- **target** - Definira ciljni prozor ili okvir na kojem će se prikazati rezultat skripte. Uzima vrijednosti poput blank, self, parent itd
- **enctype** - Definira kako preglednik kodira podatke prije nego što ih pošalje poslužitelju

```

<form action="/action.php" method="POST" target="_blank" enctype="multipart/
  form-data">
</form>

```

Kod 16: Osnovni atributi HTML obrasca

5.2 Elementi kontrole unosa HTML obrasaca

Tekstna polja <input> najosnovniji su elementi za kontrolu oblika obrasaca. Korisniku omogućuje unošenje bilo koje vrste podataka.

```

<input type="text" id="comment" name="comment" value="Osnovna vrijednost">.

```

Kod 17: Polje za unos teksta

Postoji nekoliko vrsta tekstualnog polja, a to je moguće ostvariti mijenjanjem type atributa u sljedeće vrijednosti:

- **password** - vrijednost lozinke ne dodaje nikakva posebna ograničenja za uneseni tekst, ali sakriva vrijednost upisanu u polje (npr. S točkicama ili zvjezdicama), tako da je drugi ne mogu lako pročitati.

- **hidden** - koristi se za stvaranje kontrole obrasca koja je nevidljiva za korisnika, ali se i dalje šalje na poslužitelj zajedno s ostatkom podataka obrasca nakon što se predaju

Osim navedenih vrijednosti koji se najčešće koriste, vrijednosti search, tel i url dodani su naknadno u HTML5.

Potvrdni okvir stvara se pomoću elementa `<input>` s atributom tipa (type) postavljenim na vrijednost potvrdnog okvira (kod 18).

```
<input type="checkbox" id="mrkva" name="mrkva" value="mrkva" checked>.
```

Kod 18: Deklariranje potvrdnog okvira

Uključivanje atributa "checked" potvrdni okvir automatski se potvrđuje prilikom pokretanja web stranice. Klikom na potvrdni okvir ili pridruženu naljepnicu potvrdni okvir se uključuje ili isključuje.

Radio gumb stvara se pomoću elementa `<input>` s atributom tipa (type) postavljenim na vrijednost radio gumba (kod 19).

```
<input type="radio" id="mrkva" name="mrkva" checked>.  
  
<input type="radio" id="id1" name="grupa" value="vrijednost1" >.  
<input type="radio" id="id2" name="grupa" value="vrijednost2" >.
```

Kod 19: Deklariranje radio gumba

Za razliku od potvrdnog okvira, nekoliko radio gumbova može biti povezano. Ako postoje više istih vrijednosti za atribut "name", radio gumbovi će se automatski grupirati pod određenu grupu. Moguće je provjeravati samo jedan gumb u određenoj grupi - ako se jedan gumb provjeri, svi ostali gumbi se automatski provjeravaju. Prilikom slanja obrasca, šalje se samo vrijednost provjerenog radio gumba. Ako ni jedan radio gumb nije potvrđen, smatra se da se niz radio gumbova nalazi u nepoznatom stanju te obrazac ne šalje vrijednosti.

Gumbi (button) predstavljaju stvarne semantičke gumbe, za razliku od radio gumbova ili potvrdnih okvira (kod 20). Postoje nekoliko vrsta gumbova koje se definiraju unutar "type" atributa:

- **submit** - gumb s navedenim atributom šalje podatke obrasca poslužitelju. Ako se ne definira vrsta gumba navedenim atributom, rezultat takvog gumba imat će "submit" vrijednost
- **reset** - resetira sve vrijednosti elemenata forme
- **button** - vrijednost atributa koja nema nativno ponašanje za razliku od "submit" ili "reset", ali se može definirati određena funkcionalnost pomoću javascript koda.

```
<button type="submit">  
  Ovo je <strong>submit gumb</strong>  
</button>  
<input type="submit" value="Ovo je submit gumb">  
  
<button type="reset">
```

```

    Ovo je <strong>reset gumb</strong>
</button>
<input type="reset" value="Ovo je reset gumb">

<button type="button">
    Ovo je <strong>anonimni gumb</strong>
</button>
<input type="button" value="Ovo je anonimni gumb">

```

Kod 20: Deklariranje gumbova - 2 načina

Kako bi zaokružili cijelinu, primjer obrasca s navedenim gornjim primjerima nalazi se u kodu 21.

```

<form action="obradi.php" method="POST" enctype="application/x-www-form-
urlencoded">
  <fieldset>
    <legend>Velicina podatka</legend>
    <p>
      <input type="radio" name="size" id="size_1" value="small">
      <label for="size_1">Mala</label>
    </p>
    <p>
      <input type="radio" name="size" id="size_2" value="medium">
      <label for="size_2">Srednja</label>
    </p>
    <p>
      <input type="radio" name="size" id="size_3" value="large">
      <label for="size_3">Large</label>
    </p>
  </fieldset>
  <fieldset>
    <legend>Kontakt podatci</legend>
    <p>
      <label for="email">Email</label>
      <input type="email" name="email" id="email" placeholder="Unesite email">
    </p>
  </fieldset>
  <p>
    <label for="ime">Prihvacam uvjete koristenja.</label>
    <input type="checkbox" name="uvjetikoristenja" id="uvjetikoristenja">
  </p>
  <button type="submit">Posalji</button>
</form>

```

Kod 21: Primjer obrasca s navedenim elementima kontrole unosa

Preostale attribute i elemente HTML obrasca moguće je pronaći na poveznicama [5] [6].

6 Preučitavanje sadržaja

Vrijednost deklarirana unutar rel atributa u <link> elementu omogućuje deklariranje zahtjeva za dohvaćanje pojedinih lokalnih ili eksternih poveznica unutar HTML <head> elementa, navodeći resurse koje stranica zahtjeva u što kraćem roku, prije učitavanja i obrađivanje stranice od strane preglednika. Preučitavanjem pojedinih elemenata poboljšavamo performanse web stranice jer elementi koji bi najčešće blokirali izvođenje stranice prilikom učitavanja ne predstavljaju više problem.

```
<head>
  <meta charset="utf-8">
  <title>Primjer preducitavanja sadrzaja unutar stranice</title>

  <link rel="preload" href="style.css" as="style">
  <link rel="preload" href="main.js" as="script">

  <link rel="stylesheet" href="style.css">
</head>

<body>
  <h1>Mjau Mjau</h1>

  <script src="main.js" defer></script>
</body>
```

Kod 22: Primjer preučitavanja pojedinih elemenata unutar stranice

U primjeru koda 22 preučitavamo CSS i JavaScript datoteke kako bi bile što prije dostupne za kasnije prikazivanje stranice. Navedeni primjer je trivijalan, jer preglednik vjerojatno otkriva elemente <link rel="stylesheet"> i <script> u istom dijelu HTML-a kao i preučitavanja, ali prednosti ovakvog pristupa su u preučitavanju zahtjevnih resursa iz CSS-a, poput fontova ili slika, dohvaćanje resursa koje JavaScript može zatražiti, poput JSON-a, uvezenih skripti ili web radnika i učitavanje velikih slika ili videa koji bi blokirali učitavanje stranice.

U HTML-u, različite vrste sadržaje dostupne su za preučitavanje. Moguće vrijednosti atributa su:

1. **audio**: audio datoteka, koja se obično koristi u <audio> elementu.
2. **document**: HTML dokument namijenjen za ugradnju pomoću <frame> ili <iframe>.
3. **embed**: resurs koji se ugrađuje unutar elementa <embed>.
4. **fetch**: Resurs kojem treba pristupiti pomoću fetch ili XHR zahtjeva
5. **font**: Datoteka fonta.
6. **image**: Datoteka slike.
7. **object**: Resurs koji se ugrađuje unutar elementa <object>.
8. **script**: JavaScript datoteka.
9. **style**: CSS tablica stilova.

10. **worker**: JavaScript web radnik ili zajednički radnik.
11. **video**: video datoteka, koja se obično koristi u <video> elementu.

Također, <link> elementi mogu prihvatiti atribut **type** koji sadrži MIME vrstu resursa na koji navedeni element pokazuje. To je posebno korisno kod predohvaćanja resursa - preglednik će upotrijebiti vrijednost atributa type da bi utvrdio podržava li korištenje navedenog resursa, te će ga ignorirati ukoliko ga ne podržava.

```
<head>
  <meta charset="utf-8">
  <title>Primjer korištenje type atributa</title>

  <link rel="preload" href="short.mp4" as="video" type="video/mp4">
  <link rel="preload" href="short.webm" as="video" type="video/webm">
</head>
<body>
<video controls>
  <source src="short.mp4" type="video/mp4">
  <source src="short.webm" type="video/webm">
  <p>Vas browser trenutno ne podržava niti jedan tip videa</p>
</video>
</body>
```

Kod 23: Korištenje atributa type unutar predučitavanja sadržaja

U primjeru koda 23 preglednici koji podržavaju MP4 datoteke unaprijed će učitati i koristiti MP4 sadržaj. Preglednici koji ne podržavaju MP4 i dalje mogu učitati WebM verziju datoteke, ali nemaju prednosti predučitavanja sadržaja. To pokazuje kako se prethodno učitavanje sadržaja može kombinirati s filozofijom progresivnog poboljšanja.

Također, postoje i druge značajke kod preddohvaćanja podataka, ali niti jedna nije prikladna za svrhu kao <link rel = "preload">:

1. **<link rel = "prefetch">** je već dugo podržan u preglednicima, ali namijenjen je za prethodno dohvaćanje resursa koji će se koristiti prilikom učitavanja sljedeće stranice
2. **<link rel = "prerender">** generira web stranicu u pozadini, ubrzavajući njeno učitavanje ako je korisnik potencijalno otvori. Navedeni koncept nije dobar jer stvara dodatno opterećenje na server i dodato troši korisnikov promet prilikom pretrage
3. **<link rel = "subresource">** bio je podržan u Chromeu u prošlosti za predučitavanje sadržaja, ali je imao problem: nije bilo načina za utvrđivanje prioriteta predučitavanja pojedinih stavki elemenata (pogledaj gornji primjer predučitavanja tipova video zapisa)

7 Slike i dodatni sadržaj



Figure 2: HTML5 semantička struktura

References

- [1] bitdegree. *The Origins of HTML5 Semantic Tags*. URL: <https://www.bitdegree.org/learn/html5-semantic-tags>. (pristup: 05.03.2020).
- [2] Google Developer. *Google developer docs*. URL: <https://developers.google.com/web/fundamentals/accessibility/semantics-aria>. (pristup: 07.03.2020).
- [3] MDN web docs. *HTML5 - Developer guides | MDN*. URL: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>. (pristup: 05.03.2020).
- [4] MDN web docs. *HTML5 - Developer guides | MDN*. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>. (pristup: 07.03.2020).
- [5] MDN web docs. *HTML5 input types - Developer guides | MDN*. URL: https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types. (pristup: 08.03.2020).
- [6] MDN web docs. *HTML5 other form control | MDN*. URL: https://developer.mozilla.org/en-US/docs/Learn/Forms/Other_form_controls. (pristup: 08.03.2020).
- [7] W3C. *HTML5 sections*. URL: <https://www.w3.org/TR/2011/WD-html5-20110525/sections.html#the-section-element>. (pristup: 05.03.2020).
- [8] W3C. *W3C aria working draft*. URL: <http://w3c.github.io/html-reference/aria/aria.html#aria-attrs-all>. (pristup: 08.03.2020).
- [9] w3schools. *HTML attribute*. URL: https://www.w3schools.com/tags/ref_attributes.asp. (pristup: 07.03.2020).
- [10] w3schools. *HTML5 tag reference*. URL: <https://www.w3schools.com/tags/default.asp>. (pristup: 06.03.2020).