

### 13. Duomenų aprašymo kalbos *HTML* ir *XHTML*

*HTML* yra puikus pavyzdys, kaip laisvi atvirieji standartai gali pakeisti pasaulį. Šiame skyriuje apžvelgsime *HTML* istoriją, *HTML* kalbos elementus, *HTML* kalbos ir žiniatinklio tendencijas. Pamatysime, kaip kyla nesutarimai tarp laisvųjų standartų kūrėjų ir kaip tai gali veikti standarto vystymąsi ir naujų standartų atsiradimą.

#### 13.1. Dokumentų formato ir perdavimo standartai

**Žiniatinklis** – kitaip pasaulinis voratinklis – angl. *World Wide Web (WWW)* – tarpusavyje susietų *HTML* dokumentų ir juos papildančių kitokių failų, prieinamų internete *http* protokolu, visuma.

**HTML** – angl. *Hypertext Markup Language* – teksto su nuorodomis žymių kalba. Jos paskirtis – aprašyti dokumento struktūrą.

**HTTP** – angl. *Hyper Text Transfer Protocol* – hiperteksto persiuntimo protokolas – taisyklių ir susitarimų rinkinys, skirtas hipertekstinių dokumentų persiuntimo būdui apibrėžti, kai informacija siunčiama iš tarnybinės stoties į vartotojo kompiuterį. Kai vartotojas savo interneto naršyklėje nurodo adresą ar paspaudžia interneto puslapyje nuorodą, jo kompiuteris susijungia su tinklo tarnybine stotimi ir siunčia į ją pranešimą. Tada iš tarnybinės stoties siunčiami failai, reikalingi interneto svetainės puslapiui vaizduoti.

#### 13.2. Žiniatinklio atsiradimas

Žiniatinklis (*World Wide Web*) yra prieinamų ir tarpusavyje susietų *HTML* kalba aprašytų dokumentų visuma. Tam, kad taptų prieinami, jie turi būti pateikti tam skirtuose serveriuose. Be to, jie gali būti papildomi kitų formatų dokumentais ar grafinėmis iliustracijomis. Yra kelios genialios idėjos, siejamos su žiniatinklio gimimu.

Perduodant dokumentą iš vieno kompiuterio kitam nebuvo garantijos, kad gavėjas galės juo tinkamai pasinaudoti. Sudėtingesnio dokumento perdavimas nebuvo įmanomas tuo atveju, jei siun-

tėjas ir gavėjas nesinaudojo ta pačia programine įranga. Net ir naudojantis identiška programine įranga perduodant dokumentą galėjo kilti problemų dėl skirtingos vaizduoklių raiškos ar kitų skirtumų. Akivaizdžiai reikėjo rasti būdą, kaip perduoti dokumento esmę tarp skirtingų sistemų. Užtikrinti technologinį neutralumą, tai yra, kad dokumentu būtų galima pasinaudoti nepriklausomai nuo naudojamo kompiuterio ar operacinės sistemos. Būtent dokumento turinio, o ne jo išvaizdos perdavimas buvo pradinis tikslas.

Labai svarbus *HTML* komponentas yra nuorodos. Ne veltui jos paminėtos netgi kalbos pavadinime (*hypertext*). Buvo svarbu sukurti bendrą adresų sintaksę, kuri leistų nurodyti tikslią vietą konkrečiame dokumente.

Taip atrodo bendra nuorodos struktūra (adresas rašomas be tarpų):

protokolas : serveris / kelias / failas ? parametrai # vieta
--

Pavyzdžiui:

<a href="http://www.example.net/darbo/doku.php?id=12#5_sk">http://www.example.net/darbo/doku.php?id=12#5_sk</a>
---

Svarbu tai, kad bet kuris elementas gali būti praleistas. Nesant nurodyto protokolo naršyklė pasirenka įprastą protokolą *http*. Nesant nurodyto serverio naršyklė ieškos dokumento tame pačiame serveryje, kuriame buvo nuoroda. Taip gaunamos santykinės nuorodos. Kelias nenurodomas, jei norima atversti serverio šakniniame aplanke esantį dokumentą. Jei nenurodytas nei serveris, nei kelias, tada failo ieškoma tame pačiame aplanke, iš kurio gauta nuoroda. Nenurodžius failo serveris pateikia nustatytą standartinį aplanko failą. Parametrų gali ir nebūti, o nenurodžius konkrečios vietos dokumentas rodomas nuo pradžios.

Nuorodose į kitus resursus tame pačiame serveryje visada įmanomas absoliučių ir santykinųjų nuorodų pasirinkimas.

Pavyzdžiui, absoliutus adresas:

```
http://www.example.net/abc/def.html#antras
```

Tas pats resursas rodant iš kito aplanko tame serveryje:

```
/abc/def.html#antras
```

Tas pats resursas rodant iš kito failo tame pačiame aplanke:

```
def.html#antras
```

Vieta dokumente rodant iš to paties dokumento:

```
#antras
```

### 13.3. *HTML* elemento konstrukcija

*HTML* kalba sukurtas dokumentas – tai tekstas su specialiais žymenimis tarp ženklų „<“ ir „>“. Taip tekstas padalinamas į tokio formato elementus:

```
<komanda> turinys </komanda>
```

Turinys gali būti tekstas bei kiti elementai, komanda gali turėti parametrus:

```
<komanda param="reikšmė"> turinys </komanda>
```

### 13.4. Paprasčiausias *HTML* dokumentas

Visas *HTML* kodas pateikiamas elemente <html>, kuris susideda iš dviejų dalių. Pradžioje techninė informacija pateikiama elemente <head>, paskui dokumento turinys pateikiamas elemente <body>.

```
<html>
  <head>
    <title>Pirmasis HTML pavyzdys</title>
  </head>
  <body>
    <h1>Pirmasis HTML dokumento pavyzdys</h1>
    <p>Pirmojo dokumento pirmoji pastraipa.</p>
    <p>Pirmojo dokumento antroji pastraipa.</p>
  </body>
</html>
```

Be anksčiau paminėtų elementų, čia panaudoti dar trys:

`<title>` – taip antraštėje nurodomas dokumento pavadinimas, rodomas lango antraštėje ir naudojamas automatiškai kuriant nuorodą į dokumentą.

`<p>` – pastraipos elementas.

`<h1>` – skirtas kurti aukščiausio lygio pavadinimams dokumento tekste (*heading 1*). Smulkesnių skyrių pavadinimai žymimi `<h2>`, jų poskyriai – `<h3>` ir taip toliau.

Šis *HTML* kodas atitinka visų *HTML* versijų ir mokyklų reikalavimus. Prieš aptariant kitas komandas būtina paminėti versijas ir mūsų dėl *HTML* kodo vystymo krypčių.

### 13.5. Vaizdas ar turinys

Nuo pradžių *HTML* kalba sukurta kaip priemonė turiniui perteikti nepriklausomai nuo naudojamos įrangos. Autorius sužymi dokumento struktūrą, o naršyklė ją atvaizduoja taip, kad dokumento skaitytojui būtų kuo patogiau. Autorius neturėtų rūpintis išvaizda, nes jis nežino, kokios yra dokumentą atverčiančios naršyklės galimybės.

Tačiau *HTML* dokumentus ėmus kurti plačiau visuomenei paaiškėjo, kad daugelis autorių labiau rūpinasi vaizdu negu turiniu. Naršyklių kūrėjai siekė papildyti *HTML* kalbą kuo daugiau išvaizdą aprašančių elementų. Autoriai pamėgo klaidingai naudoti

turinio elementus dokumentui pagražinti. Pavyzdžiui, `<h1>` elemente pateikiamas visai ne pavadinimas, bet sakinio dalis, kurią norima pabrėžti, išskirti. *HTML* kūrėjai pritarė „švariam“ be dizaino priemaišų *HTML*, o naujai atsiradusi ir vis stiprėjanti nekvalifikuotų naudotojų grupė reikalavo galimybių gražinti dokumentą, kištis į jo atvaizdavimą.

Kalba papildyta ne tik išvaizdai skirtais elementais `<b>` (*bold* – pusjuodis šriftas), `<i>` (*italic* – kursyvas – pasvirasis šriftas), bet ir kitiems elementams suteikiami išvaizdai nurodyti skirti parametrai, pavyzdžiui, `<body background=“red“>`.

### 13.6. Ar toleruotinos *HTML* sintaksės klaidos

Nevienareikšmiška buvo ir bendruomenės pozicija dėl *HTML* taisyklių griežtumo. Kompiuteriams tampant vis galingesniems naršyklės lengvai ištaiso *HTML* netikslumus.

Pavyzdžiui, toks kodas, nors ir yra netikslus, buvo pripažintas teisingu:

```
<p>Miške kur eglės ošia,<p>Po pušim sena
```

Nors nepažymėta pirmosios pastraipos pabaiga, bet akivaizdu, kad jei prasidėjo antroji pastraipa, tai pirmoji pasibaigė.

Naršyklėms nebuvo sudėtinga priimti dokumentą nepriklausomai nuo to, ar didžiosios, ar mažosios raidės naudojamos *HTML* komandoms. Todėl komandas kiekvienas rašė kaip kam patinka. Paplito trys užrašymo sistemos: `<body>`, `<BODY>` ir `<Body>`. Sintaksės reikalavimai leido rašyti ir `<BoDy>` ar dar 12 šios komandos variantų.

Dėl sintaksės tikslumo nors ir nebuvo tokių didelių ginčų, kaip esmės ir formos kovoje, bet vieni autoriai siūlė praleisti nuspėjamus *HTML* komponentus, taip sutrumpinti dokumento apimtį, sutaupyti vietą serveryje bei krovimosi laiką, kiti siūlė laikytis griežtos *HTML* sintaksės.

**13.7. Klausimai ir uždaviniai**

1. Kokia yra *HTML* paskirtis?
2. Sukurti dokumentą tvarkinga *XHTML* kalba, susidedantį bent iš trijų skyrių. Automatiškai patikrinti *HTML* kokybę. Pavyzdžiui: „Biografija“, „Hobis“, „Gimtojo miesto rajonai“.
3. Rasti internete nedidelės apimties netvarkingą *HTML* dokumentą ir iš jo padaryti tvarkingą dokumentą *XHTML* kalba.