

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**TĪMEKĻA STANDARTU ATBILSTĪBAS  
PĀRBAUDE DAŽĀDĀM PĀRLŪKPROGRAMMĀM**

BAKALaura DARBS

Autors: **Līviņa Golosujeva**

Studenta apliecības Nr.: lg10066

Darba vadītājs: Dr. sc. comp. Jānis Zuters

RĪGA 2014

## ANOTĀCIJA

Tīmeklī lietotāji var apskatīt dažādas tīmekļa vietnes, kuras izstrādātas pēc to standartiem. Lai lietotāji piekļūtu šīm vietnēm, to datoros ir uzstādītas tīmekļa pārlūkprogrammas. Gan tīmekļa standarti, gan pārlūkprogrammas tiek regulāri atjaunotas, tāpat mainās arī lietotāju vēlmes un prasības. Svarīgi, lai tīmekļa vietņu izstrādātāji sekotu līdzī jaunumiem, aktualitātēm. Darba mērķi ir apkopot populārākos HTML un CSS jaunāko standartu elementus un atribūtus, to izmantošanas iespējas. Darbā tiek aprakstītas izstrādātāju visbiežāk pieļautās kļūdas. Tiek veikts pētījums, kurā apskatīts, kā dažādi koda gabali tiek attēloti populārākajās pārlūkprogrammās.

Atslēgvārdi: *tīmekļa vietne, pārlūkprogramma, tīmekļa standarti, HTML, CSS.*

## ABSTRACT

### Web standards compliance checking for different types of browsers

On the Web, users can view a variety of websites, which are designed according to web standards. Users have different browsers on their computer, where they can view websites. Both web standards and browser are regularly updated, also user needs and requirements are frequently changing. It is important that developers of websites follow latest web standart and other news, current events. The goal is to gather the most popular latest HTML and CSS standards elements and attributes that can be used. The paper describes most common mistakes developers make. In this work there is research, which looks at how the different pieces of code are displayed in the main browsers.

Keywords: web site, the browser, web standards, HTML, CSS.

# SATURS

Apzīmējumu saraksts .....	6
Ievads .....	7
1. HTML un CSS tīmekļa standarti.....	8
__1.1 Hiperteksta iezīmēšanas valoda .....	8
___1.1.1. Vēsture .....	8
___1.1.2. No HTML uz HTML5 .....	9
___1.1.3. HTML5 jaunie elementi .....	10
_____1.1.3.1. Semantiskie elementi .....	10
_____1.1.3.2. Multivides un grafiskie elementi.....	12
_____1.1.3.3. Formu elementi .....	13
_____1.1.3.4. Ievadlauku atribūti .....	14
__1.2. Kaskadētas stila lapas.....	16
___1.2.1. Vēsture .....	16
___1.2.2. CSS3 jaunie elementi .....	17
_____1.2.2.1. Rāmju atribūti.....	17
_____1.2.2.2. Fona atribūti .....	18
_____1.2.2.3. Teksta atribūti.....	20
_____1.2.2.4. Stila pāreju un animācijas atribūti.....	23
2. HTML un CSS standartu izmantošanas iespējas.....	24
__2.1. HTML un CSS standartu izmantošana elektroniskajās vēstulēs.....	24
___2.1.1. HTML5 izmantošana elektroniskajās vēstulēs.....	25
___2.1.2. CSS3 izmantošana elektroniskajās vēstulēs.....	26
__2.2. Izstrādātāju visbiežāk pieļautās HTML un CSS izmantošanas kļūdas .....	27
___2.2.1. Visbiežāk pieļautās kļūdas, izmantojot HTML standartus .....	27
___2.2.2. Visbiežāk pieļautās kļūdas, izmantojot CSS standartus.....	29
3. Pētījums par HTML un CSS kodu dažādās pārlūkprogrammās .....	32

__3.1. Tīmekļa vietnes galvene.....	33
__3.2. Tīmekļa vietnes kājene.....	35
__3.3. Tīmekļa vietnes multivides daļa.....	38
__3.4. Tīmekļa vietnes informatīvais saturs .....	41
__3.5. Tīmekļa vietnes informācijas bloku izkārtojums .....	45
__3.6. Pētījuma rezultāti .....	49
Secinājumi.....	50
Izmantotā literatūra .....	51

## APZĪMĒJUMU SARAKSTS

CSS	Kaskadētas stila lapas [8].
DOM	Digitālo resursu (digitalizēto un/vai digitāli radīto) pārvaldības sistēma [22].
HTML	Hiperteksta iezīmēšanas valoda [22].
HTTP	Tīkla Internet standartprotokols, kas nodrošina informācijas apmaiņu globālajā tīmeklī [22].
Javascript	Skriptu valoda, kas balstīta uz prototipu koncepta [22].
XHTML	Paplašināmā hiperteksta iezīmju valoda [22].
W3C	Globālā tīmekļa konsorcijs [22].

## IEVADS

Globālais tīmeklis ir populārākais tīkla Internet pakalpojuma veids. [1] Tīmeklī lietotāji var apskatīt dažādas tīmekļa vietnes, kuras izstrādātas pēc tīmekļa standartiem. Lai lietotāji piekļūtu tīmekļa lappusēm, to datoros ir uzstādītas tīmekļa pārlūkprogrammas. Izplatītākās pārlūkprogrammas ir Internet Explorer, Mozilla Firefox, Safari, Google Chrome, Opera [1].

Tā kā tīmekļa vietnes tiek izstrādātas izmantojot HTML standartus un to projektējums tiek realizēts izmantojot CSS standartus, ir svarīgi, ka vietņu izstrādātāji seko līdzi standartu jaunumiem un veido tīmekļa vietņu lapas atbilstoši tiem. Ņemot vērā internetā pieejamo informāciju un darba autora personīgo pieredzi, var teikt, ka arvien biežāk jaunie izstrādātāji neseko līdzi tīmekļa standartu jaunumiem un bieži vien nezina pat to pamatus. Tas traucē kvalitatīvi izstrādāt tīmekļa vietnes, un nezināšanas dēļ tiek pieļautas dažādas kļūdas.

Laikam ejot tīmekļa vietnes mainās un to saturs kļūst dinamiskāks, daudzfunkcionāls. Lai nodrošinātu, ka izstrādātās lapas ir ērti pieejamas un saprotamas pēc iespējas vairāk lietotājiem, jāņem vērā tīmekļa standartu īpatnības un to attēlojums dažādās pārlūkprogrammās, jo daļa no standartu piedāvātajiem elementiem vai atribūtiem netiek atbalstīti kādā no pārlūkprogrammām.

Darba mērķi ir apkopot populārākos HTML un CSS standartu elementus un atribūtus, kurus piedāvā to jaunākās versijas. Aprakstīt izstrādātāju visbiežāk pieļautās kļūdas, strādājot ar HTML un CSS, ņemot vērā internetā pieejamo informāciju un darba autora personīgo pieredzi. Izpētīt iespējas izmantot jaunākos tīmekļa standartus, izstrādājot elektronisko vēstuli. Izpētīt dažādu HTML un CSS koda fragmentu attēlojumu populārākajās pārlūkprogrammās. Tiek izvirzīta hipotēze, ka vissliktāk tīmekļa standarti tiek attēloti pārlūkprogrammā Internet Explorer.

Darbs sastāv no 3 nodaļām. Pirmajā nodaļā apkopota informācija par HTML, tā jaunākās versijas piedāvātajiem elementiem un izmantošanas iespējām; par CSS, tā jaunākās versijas piedāvātajiem atribūtiem un to izmantošanas iespējām. Otrajā nodaļā apkopotas izstrādātāju visbiežāk pieļautās kļūdas, strādājot ar HTML un CSS, un informācija par tīmekļa standartu izmantošanas iespējām elektroniskajās vēstulēs. Trešajā nodaļa ir pētījums, kurā aprakstīts, kā dažādi koda fragmenti izskatās populārākajās pārlūkprogrammās.

# 1. HTML UN CSS TĪMEKĻA STANDARTI

Globālais tīmeklis ir populārākais tīkla Internet pakalpojuma veids. Globālo tīmekli veido ar hipersaitēm savā starpā saistītas tīmekļa lappuses. Tīmekļa vietni veidojošie dati atrodas tīklam Internet pievienotos datoros, kuros ir uzstādīta speciāla programma - tīmekļa serveris. Tīmekļa serveris atbilstoši protokola HTTP prasībām nodrošina vietnē esošās informācijas nosūtīšanu lietotājiem. Protokolu HTTP izmanto hipersaišu veidošanai starp hiperteksta dokumentiem. Noklikšķinot peli uz kādas no hipersaitēm, ar šī protokola starpniecību tiek atvērts attiecīgais dokuments neatkarīgi no tā, kur šis dokuments tīklā Internet ir izvietots. Lai lietotāji piekļūtu tīmekļa lappusēm, to datoros ir uzstādītas tīmekļa pārlūkprogrammas. Izplatītākās pārlūkprogrammas ir Internet Explorer, Mozilla Firefox, Safari, Google Chrome, Opera. Konkrētu tīmekļa lappusi var atvērt, norādot pārlūkprogrammas adresi laukā tīmekļa adresi. Izvēlētajā lappuse tiek lejupielādēta lietotāja datorā un parādīta pārlūkprogrammas logā [1].

## 1.1 Hiperteksta iezīmēšanas valoda

Hiperteksta iezīmēšanas valoda jeb HTML (no angļu: HyperText Markup Language) ir iezīmēšanas valoda, kas ir izstrādāta tīmekļa lappušu un citas pārlūkprogrammā attēlojamās informācijas glabāšanai [2]. HTML ir iezīmēšanas forma, kas ir vērsta uz vienas lappuses teksta dokumentu attēlošanu speciālās programmas HTML lietotāja aģentos, kuru visizplatītākais paveids ir tīmekļa pārlūkprogramma [3].

### 1.1.1. Vēsture

1980. gadā fiziķis Tims Bērnerss-Lī (no angļu Tim Berners-Lee) piedāvāja ideju par sistēmu, ar kuras palīdzību darbinieki kompānijā, kurā viņš strādāja, varētu koplietot dokumentus. 1989. gadā Bērnerss-Lī iepazīstināja ar ideju par hiperteksta sistēmu, izstrādātu,

lai darbotos Internetā. Viņš precizēja HTML valodu un izveidoja pārlūkprogrammu un servera programmatūru 1990. gada beigās [4].

Pēc tam, kad 1994. gada sākumā HTML un HTML+ projekti novecoja, IETF izveidoja HTML Darba Grupu, kura 1995. gadā izveidoja “HTML 2.0” – pirmo HTML aprakstu, kurš tika izveidots, lai turpmāk kalpotu kā pamats visām nākotnē plānotajām izstrādēm [4].

1999. gada beigās tika izstrādāta “HTML 4.01”. Šo standartu attīstība turpinājās līdz 2001. gadam. 2004. gadā tika sākts darbs pie “HTML5” izstrādes, un tā testa versija plašākai publikai tika piedāvāta 2008. gada maijā [4].

### **1.1.2. No HTML uz HTML5**

Katra jaunā HTML versija tiek izstrādāta ar mērķi, lai nodrošinātu lielāku vienprātību starp izstrādātājiem un lai gūtu pārliecību, ka izstrādātie dokumenti pēc noteikta laika perioda nekļūs nelasāmi. HTML izstrādātājiem ir mērķis panākt, ka visas ierīces spēj iegūt datus no Interneta [5].

Pēc “HTML 4.01” izstrādātāji pievērsās plašākai hiperteksta iezīmju valodas izpētei un izveidei. Tā radās ideja izstrādāt jaunu valodu – paplašināmo hiperteksta iezīmju valodu XHTML. Tā ir iezīmju valoda, kas nodrošina tās pašas iespējas, ko HTML, taču nosaka daudz striktāku valodas sintaksi. Vajadzībā pēc striktākas HTML versijas radās līdz ar domu, ka tīmekļa saturu jāpiegādā plašākam ierīču spektram (piemēram, mobilajiem telefoniem), no kurām daudzām ir ierobežoti resursi, kurus nevar atļauties tērēt visdažādāko brīvās HTML valodas paveidu atbalstam [6].

Tā kā hiperteksta iezīmēšanas valodas standartu galvenais uzdevums ir nodrošināt, ka izstrādātie dokumenti ir vienotā stilā un vienādi viegli salasāmi, tika izveidots HTML5, kas satur ne tikai HTML standartus, bet arī XHTML un DOM. HTML5 ir mēģinājums definēt vienotu iezīmēšanas valodu, kura ļautu rakstīt vai nu HTML vai XHTML sintaksē [7].

### 1.1.3. HTML5 jaunie elementi

Kopš 1991. gada, kad HTML 4.01 versija kļuva par standartu, ļoti ir mainījies Internets un tas, kā to lieto. Mūsdienās vairāki HTML 4.01 elementi ir novecojuši, nekad nav izmantoti vai tiek izmantoti ne tā, kā tie ir paredzēti. Visi šie elementi ir izņemti vai pārrakstīti jaunajā HTML5 versijā. Tāpat, lai pielāgotos mūsdienu Interneta vajadzībām, HTML5 iekļauj dažādus jaunus elementus, lai labāk būtu iespējams attēlot multivides saturu, nodrošinātu ērtāku lapas izkārtojumu un formu apstrādi [10].

#### 1.1.3.1. Semantiskie elementi

Mūsdienās dažādas tīmekļa vietnes bieži lieto HTML kodu, izmantojot klases atribūtu (no angļu *class*), kas definē dažādas lapas daļas. Piemēram, galveni, kājēni, galveno lapas izvēlni.

2004. gadā HTML5 specifikācijas redaktors, Google pārstāvis Ians Hiksons (no angļu *Ian Hickson*) veica datu ieguves eksperimentu, izmantojot Google rādītājus vairāk nekā 1 biljonam tīmekļa lapu, lai noskaidrotu, no kā tīmeklis ir veidots, skatoties no tīmekļa dokumenta satura puses. Viņš publicēja vairākus analīžu rezultātus, kuros viens no tiem raksturoja visbiežāk izmantotās HTML elementu CSS klases [11]. HTML5 definē elementus, kuri ļauj precīzi norādīt uz konkrētām lapas daļām, lai nebūtu jāizmanto klases. Visbiežāk izmantotie šāda veida elementi aprakstīti zemāk redzamajā tabulā 1.1. "HTML5 semantiskie elementi".

Galvene un kājene ir elementu grupas, kuras atkārtojas katrā tīmekļa vietnes lapā. Reti ir gadījumi, kad tās kādā lapā ir atšķirīgas. Galvenais uzdevums, kuru jāveic galvenei un kājenei, ir nodrošināt lietotājam vissvarīgāko informāciju vienkopus lapas augšdaļā un lejas daļā. Tā kā galvene ir lapas augšdaļā, lietotājs to redz pirmo. Tajā tiek attēlota informācija par tīmekļa vietni un saites uz svarīgākajām lapām. Tīmekļa vietnes kājene nodrošina, ka pat tad, kad lietotājs nokļūst līdz lapas apakšdaļai, informācija, kura bija pieejama galvenē, ir tik pat pieejama un viegli atrodamā.

Vēl svarīgi elementi lapā ir pamata satura daļa un lapas galvenā izvēlne. Arī šiem informācijas blokiem HTML5 piedāvā speciālus elementus, kuri aprakstīti tabulā 1.1. “HTML5 semantiskie elementi”.

1.1. tabula “HTML5 semantiskie elementi”

Elementa kods	Apraksts	Ierobežojumi
<code>&lt;header&gt;&lt;/header&gt;</code>	Elements definē lapas vai tās daļas galveni. Var būt vairāki vienā lapā, bet nevar atrasties kājenes, adreses vai citā galvenes elementā.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 9.
<code>&lt;footer&gt;&lt;/footer&gt;</code>	Elements definē lapas vai tās daļas kājeni. Var būt vairāki vienā lapā.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 9.
<code>&lt;main&gt;&lt;/main&gt;</code>	Elements definē lapas pamata saturu. Tam būtu jābūt unikālam katrā lapā, tāpēc netiek iekļauti elementi, kuri lapās atkārtojas. Var būt tikai viens vienā lapā. Nedrīkst atrasties galvenes, kājenes, izvēlnes elementos.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās.
<code>&lt;nav&gt;&lt;/nav&gt;</code>	Elements definē lapas izvēlni. Nevajadzētu šajā elementā iekļaut visas lapas saites, tikai galvenās.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 9.

HTML5 semantisko elementu uzdevums ir atdalīt dažādas lapas loģiskās daļas, tā lietotājam padarot lapu ērtāk lasāmu un pārskatāmu.

### 1.1.3.2. Multivides un grafiskie elementi

Html 4.01 versijā vēl nebija definēti elementi, kuri ļautu tīmekļa lapā attēlot skaņas vai video elementus, tāpēc ļoti bieži tika izmantotas dažādas pievienojumprogrammas. Piemēram, *Flash* nodrošināja iespēju pievienot multivides saturu [12]. Jaunā HTML5 versija nodrošina šādu iespēju, ļaujot izstrādātājiem ērti un vienkārši pievienot dažādus multivides vai grafiskos objektus.

Tālāk redzamajā tabulā 1.2. “Multivides un grafiskie elementi” aprakstīti visbiežāk izmantotie šāda veida elementi.

1.2. tabula. “Multivides un grafiskie elementi”

Elementa kods	Apraksts	Ierobežojumi
<code>&lt;embed&gt;</code>	Elements definē konteineri ārējai lietotnei vai interaktīvam saturam (pievienojumprogrammas).	-
<code>&lt;canvas&gt;&lt;/canvas&gt;</code>	Elements definē konteineri, kurā tiek attēlots ar Javascript uzzīmēts grafiskais elements. Ja elements netiek atbalstīts – tiek attēlots teksts.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 9.
<code>&lt;audio&gt;&lt;/audio&gt;</code>	Elements definē skaņu, piemēram, mūziku vai citu tāda veida datu plūsmu. Šobrīd tiek atbalstīti trīs failu formāti – mp3, wav, ogg (atbilstoši pārlūkprogrammas iespējām).	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 9.
<code>&lt;video&gt;&lt;/video&gt;</code>	Elements definē video, piemēram, filmu vai citu tāda veida datu plūsmu. Šobrīd tiek atbalstīti trīs failu formāti – mp4, webm, ogg (atbilstoši pārlūkprogrammas iespējām).	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 9.

HTML5 multivides un grafisko elementu uzdevums ir nodrošināt iespēju tīmekļa vietnēs attēlot dažādus video, skaņas vai grafiskos objektus, pieprasot minimālus resursus no pārlūkprogrammas.

### 1.1.3.3. Formu elementi

Mūsdienās gandrīz katrā tīmekļa vietnē ir atrodama kāda forma. Lai izveidotu vienkāršu formu, pietiek ar HTML pamatelementiem.

HTML5 piedāvā trīs papildelementus, kuri nodrošina mūsdienīgu pieeju formu veidošanā. Tie aprakstīti tālāk redzamajā tabulā 1.3. "Formu elementi".

1.3. tabula "Formu elementi"

Elementa kods	Apraksts	Ierobežojumi
<code>&lt;output&gt;&lt;/output&gt;</code>	Elements definē lauku, kurā tiek attēlots aprēķinu rezultāts.	Netiek atbalstīts pārlūkprogrammā Internet Explorer.
<code>&lt;datalist&gt;&lt;/datalist&gt;</code>	Elements definē ievadlauka iespējamās vērtības. Šīs vērtības tiek iepriekš aprakstītas. Kad lietotājs sāks rakstīt datus ievadlaukā, tiks parādītas piedāvātās iespējamās vērtības. Lai savienotu ievadlauku ar šo sarakstu, jāizmanto ievadlauka atribūts list.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās, kuras ir vecākas par 10 un visās pārlūkprogrammas Safari versijās.
<code>&lt;keygen&gt;</code>	Elements definē lauku formā, kurš nodrošina atslēgu pāra izveidošanu. Šis atslēgu pāris ļauj panākt drošāku lietotāja pieslēgšanos u.c. Elements izveido drošo atslēgu un apstiprina publisko atslēgu.	Netiek atbalstīts pārlūkprogrammā Internet Explorer. Tiek atbalstīts pārlūkprogrammas Safari versijā 6.

Izmantojot šos elementus, formu veidošana kļūst vieglāka.

HTML5 formu elementu uzdevums ir nodrošināt maksimālu funkcionalitāti tīmekļa vietņu formām, izmantojot tikai HTML elementus. Ja kādā no pārlūkprogrammām šie elementi netiek atbalstīti, tie tiks attēloti teksta veidā.

### 1.1.3.4. Ievadlauku atribūti

Lai gan HTML5 piedāvā jaunus formu elementus, vislielāko popularitāti ieguvuši jaunie ievadlauku atribūti. Populārākie no tiem aprakstīti zemāk redzamajā tabulā 1.4. "Ievadlauku aizpildīšanas atribūti".

1.4. tabula "Ievadlauku aizpildīšanas atribūti"

Atribūta kods	Apraksts	Ierobežojumi
placeholder="<text>"	Atribūts definē informāciju teksta veidā, kura ļauj lietotājam saprast, kādi dati jāraksta ievadlaukā. Šis teksts parādās ievadlaukā, pirms lietotājs ir sācis tajā ievadīt datus.	Atribūts strādā šādiem ievadlauku tipiem: text, search, url, tel, email, password. Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās vecākās par 10.
autocomplete="on/off"	Atribūts definē vai ievadlauku (arī formas) nepieciešams aizpildīt ar pārlūkprogrammā saglabāto informāciju – datiem, kurus lietotājs ir ievadījis iepriekš.	Atribūts strādā šādiem ievadlauku tipiem: text, search, url, tel, email, password, datepickers, range, color. Dažās pārlūkprogrammās automātiskās ievadlauku aizpildes funkcionalitāte ir jāaktivizē.

Atribūti placeholder un autocomplete lietotājam palīdz saprast, kāda informācija būtu jāievada formas ievadlaukos.

HTML5 piedāvā vēl citus formu ievadlauku atribūtus, kuri nodrošina, lai lietotājam būtu ērti ievadīt informāciju, skatīt 1.5. tabulu.

1.5. tabula "Ievadlauku atribūti"

Atribūta kods	Apraksts	Ierobežojumi
<code>autofocus</code>	Atribūts definē, vai ievadlaukam ir jābūt fokusā, kad lapa tiek ielādēta.	-
<code>required</code>	Atribūts definē vai ievadlauks ir obligāti aizpildāms.	Atribūts strādā ar šādiem ievadlauku tipiem: text, search, url, tel, email, password, date pickers, number, checkbox, radio, file. Netiek atbalstīts pārlūkprogrammā Safari.
<code>list="&lt;list-id&gt;"</code>	Atribūts definē datalist elementa id, kurš nodrošina iepriekš definētas ievadlauka vērtības saraksta veidā.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās vecākās par 10 un visās Safari versijās.
<code>pattern="&lt;regular expression&gt;"</code>	Atribūts definē izteiksmi, kura tiek lietota, pārbaudot ievadlauka datus.	Atribūts strādā šādiem ievadlauku tipiem: text, search, url, tel, email, password. Netiek atbalstīts pārlūkprogrammā Safari.
<code>height="&lt;number&gt;"</code> <code>width="&lt;number&gt;"</code>	Atribūts definē formas augstumu un platumu. Ja tie netiek definēti, lapa mainīs izskatu pēc tam, kad tiks ielādēts ievadlauka saturs.	Atribūts strādā šādiem ievadlauku tipiem: image.

HTML5 jauno ievadlauku atribūtu uzdevums ir nodrošināt pēc iespējas lielāku funkcionalitāti, izmantojot tikai HTML elementus.

## 1.2. Kaskadētas stila lapas

CSS jeb kaskadētas stila lapas (no angļu Cascading Style Sheets) ir īpaša valoda, kas paredzēta, lai aprakstītu izskatu iezīmēšanas valodās veidotiem dokumentiem. Visplašāk tā tiek lietota Interneta mājas lapās, lai aprakstītu HTML elementu izskatu. Ar CSS palīdzību var izmainīt jebkura HTML elementa ārējo izskatu - krāsu, izmēru, novietojumu un dažādus citus atribūtus. CSS izmantošana gan atvieglo darbu izstrādātājiem un sniedz plašākas projektējuma iespējas, gan samazina HTML dokumentu apjomu, kas paātrina mājas lapas ielādi lietotājam [8].

Kaskadēto stila lapu galvenais uzdevums ir nodrošināt, lai HTML lapu saturs tiktu atdalīts no to noformējuma. Šāda satura atdalīšana var uzlabot tā pieejamību, lasāmību, nodrošina, ka vairākām lapām tiek lietots vienots stils, tādējādi samazinot lapas sarežģītību un strukturālā satura atkārtosanos [9].

### 1.2.1. Vēsture

Tā kā HTML ir ļoti ierobežotas stila iespējās, tika izveidotas kaskadētas stila lapas jeb CSS, kas ļauj pievienot stilu HTML lapām.

CSS piedāvā dažādas stila iespējas – sākot no teksta fonta maiņas, līdz rindkopu izlīdzināšanas iespējām. Viena no galvenajām CSS priekšrocībām ir iespēja vairākām lapām izveidot vienu kopīgu stilu. Izmantojot tikai HTML, stils katrai lapai jāraksta atsevišķi, bet CSS ļauj uz servera glabāt failu, kas izveidos nepieciešamo stilu visām lapām uzreiz [9].

Pirmā kaskadēto stila lapu versija tika piedāvāta 1996. gadā. Tās nosaukums bija CSS1 un tās izstrādātāji – W3C jeb Vispasaules tīmekļa konsorcijs. W3C ir visu pasaulē izmantoto tīmekļa standartu un projektējumu shēmu avots. Kopš pirmās CSS versijas iznākšanas, W3C ir piedāvājuši vēl divas jaunākas versijas – CSS2, kurš tika izveidots 1998. gadā, un CSS3, kurš pieejams no 1999. gada. No 1996. gada CSS ir tīmekļa projektējuma standarts [9].

CSS ceļš kā tīmekļa projektējuma standartam ne vienmēr ir bijis viegls. Sākumā pārlūkprogrammas tos neatbalstīja vai atbalstīja tikai daļēji. Pat mūsdienās projektējuma standarti netiek atbalstīti pārlūkprogrammās vienādi un tas, izstrādājot tīmekļa lapas, ļoti bieži rada dažādas kļūdas. CSS lēnais progress un atšķirības starp pārlūkprogrammām lika

izstrādātājiem domāt, ka labāk to neizmantojot tīmekļa lapu izstrādē, jo bija sarežģīti saprast, kas tiks attēlots un kā. Tomēr, neskatoties uz vairākām problēmām, CSS raksturo tīmekļa projektējumu un ir tā standarts [9].

## 1.2.2. CSS3 jaunie elementi

CSS3 ir pēdējais lietošanai piedāvātais CSS standarts. Jaunais standarts ir sadalīts moduļos. Tas satur “vecu CSS specifikāciju” (kura ir sadalīta mazākās daļās). Papildus ir pievienoti jauni moduļi [13].

Nosaukums “CSS3” nav tikai norāde uz jaunāko CSS funkcionalitāti, tas ir 3. līmenis CSS specifikācijas progresā. CSS3 satur gandrīz visu to pašu, kas definēts pēdējā CSS versijā – CSS2.1. Tas satur arī jaunas detaļas, kuras ļauj izstrādātājiem atrisināt dažādas problēmas, neizmantojot kodu, kurš neatbilst standartiem, sarežģītas funkcijas vai papildus attēlus [14].

Populārākie jaunie CSS3 elementi ir aprakstīti tālāk.

### 1.2.2.1. Rāmju atribūti

Izmantojot CSS3, ir iespējams izveidot īpašu stilu, kurš vēl spilgtāk ļauj norobežot dažādus tīmekļa vietnes elementus. Lai varētu pievienot apaļos stūrus, var izmantot rāmja rādiusa atribūtu, skatīt tabulu 1.6. “Rāmju rādiusa atribūts”.

1.6. tabula “Rāmju rādiusa atribūts”

Atribūta kods	Apraksts	Ierobežojumi
<code>border-radius:&lt;number&gt;px;</code>	Atribūts ļauj izveidot apaļus stūrus elementam.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 9.

Atribūts border-radius ir visbiežāk izmantotais no rāmju stila atribūtiem.

Gandrīz tik pat lielu popularitāti ieguvuši arī pārējie rāmju atribūti, kuri aprakstīt tabulā .

### 1.7. “Rāmju atribūti”.

1.7. tabula “Rāmju atribūti”

Atribūta kods	Apraksts	Ierobežojumi
box-shadow: <h-value> <v-value> <blur> <spread> <color>;	Atribūts ļauj izveidot elementa ēnu.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 9.
border-image: <source> <slice> <width> <outset> <repeat>;	Atribūts ļauj izmantot attēlu elementa rāmja izveidošanai.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 11.

Pirms CSS3 izstrādātāji bieži izmantoja attēlus, lai panāktu vēlamos rāmju stilus – apaļos stūrus, ēnu, rāmi kā attēlu. Tagad to iespējams panākt ar vienkāršu CSS atribūtu.

### 1.2.2.2. Fona atribūti

Mūsdienās dažādi tīmekļa lapu foni kļūst arvien sarežģītāki. Izstrādātāji arvien biežāk sāk izmantot dažādas papildus programmas, lai izveidotu nepieciešamo fonu. CSS3 ir izveidoti atribūti, kuri atvieglo fona izveidošanu un apstrādi tīmekļa vietnēs, papildinot jau iepriekšējās versijās izveidoto funkcionalitāti. Šie atribūti padara tīmekļa vietņu fona izstrādi ļoti vienkāršu, jo pietiek tikai ar dažām rindiņām CSS, lai panāktu vēlamo efektu, kurš piedāvāts lapas projektējuma dokumentos. Izstrādātājiem nav jādomā par attēlu vai papildus Javascript funkciju izmantošanu.

Tabulās 1.8. “Fona attēlu atribūti” un 1.9. “Fona gradientu atribūti” aprakstīti šie CSS3 standartu piedāvātie atribūti.

1.8.. tabula “Fona attēlu atribūti”

Atribūta kods	Apraksts	Ierobežojumi
background-size: <width> <height>;	Atribūts ļauj noteikt fona attēla izmēru. Izmēru var definēt pikseļos vai procentos. Procenti tiek aprēķināti ņemot vērā vecākelementa izmēru.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 9.
background-origin:<option>;	Atribūts definē fona novietojumu elementā. Tas var būt: border-box, padding-box, content-box.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 9.
background: <source>, <source>;	Atribūts ļauj izmantot vairākus fona attēlus. Katrs nākamais attēls jāatdala ar komatu.	-

Izmantojot CSS3 piedāvātos atribūtus un to uzlabojumus, izstrādātāji var veidot dažādus fonus un to kombinācijas jebkuram elementam tīmekļa vietnes lapā, ja to atļauj HTML elementa specifikācija. Ja iepriekš sarežģītu fonu izveidei vajadzēja izmantot speciālas Javascript funkcijas vai attēlu kombinācijas, kuras izveidotas kādā no attēlu rediģēšanas programmām, tagad tas ļoti bieži nav nepieciešams.

Bieži vien izstrādātājiem jāizmanto vairākas fona krāsas, attēlojot vienas krāsas toņu pārejas vai pāreju no dažādām krāsām – no vienas krāsas citā. Iepriekšējās CSS versijās tam tika izmantoti vienkārši attēli vai speciālas Javascript funkcijas. CSS3 nodrošina atribūtus, kuri piedāvā šādu funkcionalitāti. Tos izmantojot, izstrādātāji var radīt dažādus interesantus fona efektus – gan centrētus, gan lineārus – tikai ar vienu CSS atribūtu. Turklāt visi šie fona atribūti, kuri tiek piedāvāti jaunākajā CSS standartā, pēc W3C noteikumiem un aprakstiem, tiek atbalstīti visās populārāko tīmekļa pārlūkprogrammu pēdējās izdotajās versijās, kuras lietotāji jau izmanto.

Fona krāsu pāreju atribūti aprakstīt tabulā 1.9. “Fona gradientu atribūti”.

1.9.. tabula “*Fona gradientu atribūti*”

Atribūta kods	Apraksts	Ierobežojumi
background: linear-gradient( <direction>, <color>, <color>, .. );	Atribūts ļauj izveidot krāsu pāreju, izmantojot vairākas krāsas un definētu virzienu. Krāsu pāreja tiks attēlota lineāri.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 10.
background: radial-gradient( <center>, <shape-size>, <start-color>, .., <last color> );	Atribūts ļauj izveidot krāsu pāreju, izmantojot vairākas krāsas, pārejas sākumpunktu, izmēru. Krāsu pāreja tiks attēlota apļveidīgi.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 10.

Fona atribūtu uzdevums ir atvieglot izstrādātāju darbu, ļaujot tīmekļa vietnē izmantoto fona attēlu maksimāli izveidot, izmantojot tikai CSS atribūtus.

### 1.2.2.3. Teksta atribūti

Lapā var atrasties ne tikai dažādi bloka elementi, bet arī teksts. Tieši informācija, kura piedāvāta teksta veidā, ir vispiemērotākā svarīgas informācijas attēlošanai. Tā kā arvien biežāk tīmekļa vietnes tiek veidotas ļoti dinamiskas un ar dažādiem blokiem, kuri piesaista uzmanību ar spilgtām krāsām vai multivides saturu, lietotājam vienkāršs teksts kļūst neinteresants, un tas var radīt situāciju, ka svarīgā informācija netiek ievērot.

HTML piedāvā tekstu sadalīt paragrāfos un virsrakstos, bet ne vienmēr pietiek tikai ar vienkāršiem CSS elementiem, kā teksta izmērs, krāsa vai stils, lai panāktu vēlamo efektu un izceltu nepieciešamo informāciju.

CSS3 piedāvā dažādus atribūtus teksta noformēšanai. Tie aprakstīti tabulā 1.10. “Teksta atribūti”.

1.10. tabula “*Teksta atribūti*”

Atribūta kods	Apraksts	Ierobežojumi
<pre>text-shadow:&lt;h-shadow&gt; &lt;v-shadow&gt; &lt;blur&gt; &lt;color&gt;;</pre>	<p>Atribūts ļauj izveidot teksta ēnu. Iespējams pievienot vienu vai vairākas teksta ēnas.</p>	<p>Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 10.</p>
<pre>@font-face { font-family:&lt;value&gt;; src:&lt;value&gt;; }</pre>	<p>Atribūts definē teksta fontu. Fonts nav instalēts lietotāja datorā, tas atrodas kopā ar lapas failiem.</p>	<p>Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 9. ttf/otf fonta formāts darbosies pārlūkprogrammā Internet Explorer tikai tad, ja tas būs “installable”. svg fonta formātu neatbalsta pārlūkprogrammas Internet Explorer un Firefox. eot fonta formāts tiek atbalstīts tikai Internet Explorer pārlūkprogrammā, versijās sākot no 6.</p>

Pievienojot teksta ēnu vai izmantojot tam īpašu fontu, iespējams pievērst lietotāja uzmanību uz svarīgāko informāciju, pat ja tā attēlota teksta veidā.

Tā kā bieži teksts tiek ievietots kādā no tīmekļa vietnes loģiskajiem satūra blokiem, var rasties situācija, ka tas ir pārāk garš un neietilpst atvēlētajā vietā, kuru ierobežo konkrētā bloka izmēri. CSS3 piedāvā dažādus atribūtus, kuri ļauj definēt teksta izkārtojumu un veidu, kā tas tiks piemērots konkrētā bloka izmēram, skatīt tabulu 1.11.

1.11. tabula “Teksta izvietošanas atribūti”

Atribūta kods	Apraksts	Ierobežojumi
<code>word-wrap:&lt;option&gt;;</code>	Atribūts definē nosacījumus, kuri apraksta, kas notiks ar garu vārdu, kuram nav vietas esošajā rindā.	-
<code>text-overflow:&lt;option&gt;;</code>	Atribūts definē, kas notiks ar tekstu, kad tas iziet no elementa robežām.	-
<code>word-break:&lt;option&gt;;</code>	Atribūts definē nosacījumus, kuri apraksta teksta pārvešanu jaunā rindā	Netiek atbalstīts tekstiem, kuri rakstīti šādās valodās: ķīniešu, japāņu un korejiešu.
<code>column-count:&lt;number&gt;;</code> <code>column-gap:&lt;number&gt;px;</code> <code>column-rule:&lt;width&gt;</code> <code>&lt;style&gt; &lt;color&gt;;</code>	Atribūts definē, cik kolonnās jāsadala teksts, cik liela atstarpe jāatstāj starp kolonnām, kā tiks attēlotas atstarpes starp kolonnām.	Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 10. Pārlūkprogrammai Firefox atribūtam jāpievieno <code>-moz-</code> . Safari un Chrome pārlūkprogrammām atribūtam jāpievieno <code>-webkit-</code> .

Izmantojot CSS3 jaunus teksta atribūtus, izstrādātājiem ir iespēja noformēt tekstu atbilstoši vajadzībām un to maksimāli iespējams izdarīt tikai ar CSS.

### 1.2.2.4. Stila pāreju un animācijas atribūti

Lai piesaistītu tīmekļa vietnes apmeklētāju uzmanību, ne vienmēr pietiek tikai ar vienkārši noformētiem un norobežotiem elementiem. Mūsdienās arvien biežāk lietotāji vēlas redzēt lapās kustīgus elementus. Tam CSS3 piedāvā jaunus atribūtus. Populārākie no šiem elementiem aprakstīti tabulā 1.12. “Stila pāreju un animācijas atribūti”.

1.12.. tabula “Stila pāreju un animācijas atribūti”

Atribūta kods	Apraksts	Ierobežojumi
<pre>animation:&lt;name&gt; &lt;duration&gt; &lt;timing- function&gt; &lt;delay&gt; &lt;iteration-count&gt; &lt;direction&gt;;</pre>	<p>Atribūts definē elementa animāciju. Svarīgi noteikt animācijas ilgumu, jo pēc noklusējuma tas ir 0 un tā animācija nekad netiek parādīta.</p>	<p>Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 10. Pārlūkprogrammām Safari un Chrome atribūtam jāpievieno –webkit-.</p>
<pre>transition:&lt;property&gt; &lt;duration&gt; &lt;timing- function&gt; &lt;delay&gt;;</pre>	<p>Atribūts definē elementa pāreju no viena stila uz kādu citu. Svarīgi noteikt pārejas ilgumu, jo pēc noklusējuma tas ir 0 un tā pāreja nekad netiek veikta.</p>	<p>Netiek atbalstīts pārlūkprogrammas Internet Explorer versijās jaunākās par 10. Pārlūkprogrammā Safari atribūtam jāpievieno –webkit-.</p>
<pre>transform:&lt;transform- function&gt;;</pre>	<p>Atribūts definē, kā elements jāpārveido 2 vai 3 dimensijās.</p>	<p>Atbalstīts no Internet Explorer 9. Šai versijai atribūtam jāpievieno –ms- un tas ļaus pārveidot (tikai 2D). Pārlūkprogrammās Opera, Chrome un Safari atribūtam jāpievieno –webkit-.</p>

Stila pāreju un animāciju atribūtu uzdevums ir nodrošināt iespēju izveidot interaktīvu tīmekļa vietni izmantojot CSS atribūtus.

## 2. HTML UN CSS STANDARTU IZMANTOŠANAS IESPĒJAS

Izstrādājot tīmekļa vietnes, HTML un CSS standarti ir vieni no galvenajiem, kuri tiek izmantoti. Bieži izstrādātāji nenovērtē šo standartu apjomu un sarežģītību un paļaujas uz to, ka spēs piedāvāt elementus un atribūtus izmantot kvalitatīvi, necenšoties iedziļināties to specifikācijā.

Viena no sarežģītākajām tīmekļa vietnes daļām, kura jāizstrādā, ir elektronisko vēstuļu pamatformas. To veidošanā ir jāņem vērā vairāki noteikumi.

Necenšoties iedziļināties tīmekļa standartu noteikumos un specifikācijā, izstrādātāji bieži pieļauj dažādas kļūdas, kuras mēdz būt tik nopietnas, ka pat sabojā tīmekļa vietnes attēlojumu dažādās pārlūkprogrammās. Sekojot līdzi tīmekļa standartu jaunumiem un attīstībai, paveras plašas to izmantošanas iespējas.

### 2.1. HTML un CSS standartu izmantošana elektroniskajās vēstulēs

HTML elektroniskās vēstules tiek plaši izmantotas visos profesionālajos sarakstes veidos. Tās tiek izmantotas ne tikai elektronisko vēstuļu ziņojumos reklāmu veidā, bet arī dažādu lietotņu veidotiem ziņojumiem, piemēram, rēķiniem un apstiprinājuma elektroniskajām vēstulēm. Noformēti e-pasti pozitīvi ietekmē tirgotājus, dizainerus un patērētājus, bet izstrādātājiem to pareiza izveidošana var sagādāt galvassāpes, jo jāpārliciecinās, ka e-pasti izskatās labi uz visām elektronisko vēstuļu pārlūkprogrammā. Turklāt katrā no šīm programmām e-pasts izskatās atšķirīgi [17].

Tā kā HTML e-pasti tiek izmantoti arvien biežāk, ir izstrādāti daži pamatnoteikumi, kuri var atvieglot izstrādātāja darbu pie elektroniskās vēstules veidnes sagatavošanas.

- **Elektroniskās vēstules pamatstrukturai ieteicams izmantot tabulas** – izmantojot tabulas e-pastu veidošanā, ir ērtāk nodrošināt tā struktūru, ļaujot novietot vairākus elementus vienu otram blakus, neizmantojot float vai citus pozicionēšanas stilus, kuri varētu nestrādāt lielākajā daļā e-pastu pārlūkprogrammu.
- **Neizmanto pievienotus stila failus** – elektroniskās vēstules stilu vēlams iekļaut, izmantojot style elementu. E-pastu pārlūkprogrammas bieži vien ignorē

pievienotus stila failus, tāpēc visu nepieciešamo būtu vēlams rakstīt kopā ar HTML.

- **Svarīgāko elementu izcelšanai izmantot iekļautos stilus** (no angļu inline styles) – style elementa izmantošana ir ērta, ja nepieciešama globāla elementu noformēšana. Ja ir elementi, kuri īpaši jāizceļ, tam labāk izmantot iekļautos stilus pašā elementā, izmantojot atribūtu style.
- **Jāizvairās no attēliem un fona stiliem** – dažas e-pastu pārlūkprogrammas parāda attēlus, kuri iekļauti, izmantojot img elementu, bet ne attēlus, kuri iekļauti ar CSS vai kā fona attēli, vai abi kopā. Tāpat, veidojot elektroniskās vēstules, ieteicams izvairīties no tumšas fona krāsa. Ja kādu no gaišākiem foniem e-pasta pārlūkprogramma ignorēs, lietotājs redzēs tumšu tekstu uz tumša fona un nelasīs ziņu vai vēl vairāk – atzīmēs to kā mēstuli [17].

Ievērojot šos noteikumus ir lielāka iespēja, ka e-pasts izskatīsies labi uz vairākām e-pasta pārlūkprogrammām.

Protams, attīstoties HTML un CSS standartiem, ir mainījies veids, kā tiek rakstītas elektroniskās vēstules. Arvien biežāk izstrādātājiem jāveido e-pasti, kuros tiek izmantoti dažādi HTML5 un CSS3 standartu elementi.

### 2.1.1. HTML5 izmantošana elektroniskajās vēstulēs

Lai gan lielākā daļa elektronisko vēstuļu pārlūkprogrammu atbalsta tikai dažus HTML elementus, izstrādātājiem arvien biežāk rodas nepieciešamība izmantot HTML5 elementus to izstrādē. Nosūtot klientam informāciju par kādas populāras mūzikas grupas koncertu, liekas pašsaprotami, ka nepieciešams pievienot arī kādu video ar šo grupu, lai būtu skaidrs, par ko tieši ir e-pasts. Tieši multivides elementi - <canvas>, <audio>, <video> - ir tie, kurus var mēģināt izmantot HTML elektroniskās vēstules izstrādē.

Lai gan Outlook.com veida elektronisko vēstuļu pārlūkprogrammas visbiežāk multivides HTML5 elementus neatbalstīs, tos var izmantot WebKit veida elektronisko vēstuļu pārlūkprogrammās, kuras visbiežāk tiek izmantotas Apple kompānijas ražotajās ierīcēs. Lielai daļa e-pastu pārlūkprogrammu tiek rādīta alternatīva – attēls ar saiti vai vienkārši saite uz konkrēto multivides materiālu. Galvenais, kas būtu jāzina izstrādātājam, ir mērķauditorijas visbiežāk izmantotās elektronisko vēstuļu pārlūkprogrammas, un vienmēr jāpieāvā iespēja

skatīties multivīdi parastajā pārlūkprogrammā un, pirms elektroniskās vēstules sagataves apstiprināšanas, tā kārtīgi jāpārbauda [18].

Lai gan HTML5 izmantošana elektronisko vēstuļu izstrādē var izklausīties pozitīvi, ir jāņem vērā un pirms izstrādes jāapsver šādi nosacījumi [19]:

- Vienmēr ir jādomā par klientiem. E-pasts, kurš būs pārpildīts ar daudz dažādiem zvaniem un svilpēm (iespējams burtiski), var būt kaitinošs vai traucējošs klientiem. Dizaineri atgādina, ka vienmēr labāka ir vienkāršība un skaidrība.
- Video e-pastos var patērēt ļoti daudz resursus, kuri konkrētās situācijās var būt nepieejami vai pieejami ļoti minimāli, un, kā zināms, mūsdienās lēns e-pasts ir tas pats, kas miris e-pasts.
- Animācijas ir lieliskas, bet tāpat kā dažas PowerPoint prezentācijas, kuras ir pārpildītas ar dažādiem efektiem, tās pēc īsa brīža var kļūt ļoti kaitinošas.

Kā parasti, kaut kas jauns lietotājiem mēdz būt biedējošs un neuzticams sākumā. Tāpēc svarīgi, pirms HTML5 izmantošanas elektronisko vēstuļu izstrādē, izpētīt konkrēto klientu grupu, mērķauditoriju, veikt eksperimentus, izmantojot nelielus HTML5 elementu blokus, un izdarīt atbilstošus secinājumus, kuri ļaus noteikt, vai ir laiks izmantot jaunās tehnoloģijas.

### **2.1.2. CSS3 izmantošana elektroniskajās vēstulēs**

Lai gan CSS atbalsts elektronisko vēstuļu pārlūkprogrammās ir daļējs labākajā gadījumā, izstrādātāji tomēr var izmantot dažus no jaunā CSS standarta CSS3 stila atribūtiem tikmēr, kamēr ir iespējams nodrošināt pārlūkprogrammas, kuras šos atribūtus neatbalsta, ar pietiekami labām alternatīvām [20].

Ir trīs galvenie iemesli, kāpēc elementiem CSS3 stils elektroniskajās vēstulēs tiek rādīts daļēji vai netiek rādīts pilnībā [20]:

- CSS3 stila atribūtu neatbalsta konkrētā elektronisko vēstuļu pārlūkprogramma;
- CSS3 stila atribūts konkrētajā elektronisko vēstuļu pārlūkprogrammā tiek atbalstīts daļēji;
- CSS3 stila atribūtu neatbalsta pārlūkprogramma, kurā tika atvērta e-pastu pārlūkprogramma.

Lai gan CSS3 izmantošana elektronisko vēstuļu izstrādē var būt ļoti vilinoša, tomēr tiek ieteikts to lietot tikai ar dekoratīvu nozīmi elementiem, kuri arī bez šiem stiliem būs skaidri un

saprotami. Piemēram, var izmantot teksta ēnu vai veidot pogas, kuru attēlošana e-pastos pirms CSS3 izveidošanas tika panākta, izmantojot attēlus. Pat tad, ja kāds no CSS3 stiliem netiks atbalstīts vai tiks atbalstīts daļēji, elementi nezaudēs savu būtību un būs skaidri un saprotami klientiem.

## **2.2. Izstrādātāju visbiežāk pieļautās HTML un CSS izmantošanas kļūdas**

HTML un CSS standarti piedāvā daudz un dažādus elementus un atribūtus, kuri ļauj izveidot tīmekļa vietnes atbilstoši vēlamajam projektējumam. Tā kā standartos esošais piedāvājums ir tik apjomīgs, izstrādātāji bieži vien pieļauj kļūdas, jo vai nu nav pietiekami apguvuši HTML un CSS standartu pamatus vai arī neuzmanības dēļ kaut ko neuzraksta līdz galam.

Turpmākajās nodaļās tiek apskatītas izstrādātāju visbiežākās kļūdas gan veidojot tīmekļa vietņu pamatu, izmantojot HTML standartus, gan izveidoto pamatu noformējot atbilstoši nepieciešamajam projektējumam, izmantojot CSS standartus.

Kļūdas apkopotas, izmantojot darba autora personīgo pieredzi, strādājot ar jaunajiem izstrādātājiem, un tīmeklī pieejamo informāciju.

### **2.2.1. Visbiežāk pieļautās kļūdas, izmantojot HTML standartus**

Ikvienu tīmekļa vietni ir izstrādāta, izmantojot HTML standartus. Ejot laikam, standartu versijas ir mainījušās, un ir mainīties arī izmantojamo elementu un atribūtu piedāvājums.

Izstrādātājiem regulāri jāseko līdzi HTML jaunumiem, lai klientiem nodrošinātu, ka izveidotās lapas ir atbilstošas jaunākajiem standartiem. Tomēr bieži izstrādātāji tomēr neievēro šos jaunumus, tāpēc rodas dažādas kļūdas, kuras reizēm apdraud tīmekļa vietnes izskatu dažādās pārlūkprogrammās.

Šie ir izstrādātāju visbiežāk neievērotie HTML noteikumi:

- **Iekļautais** (no angļu inline) **elements nedrīkst saturēt bloka elementu** – pēc HTML noteikumiem, iekļautie elementi var saturēt tikai citus iekļautos elementus

vai nepieciešamo saturu. Bloka elements drīkst iekļaut iekļauto elementu, bet ne otrādāk.

- **Pārāk bieži tiek izmantots elements `<div></div>`** - tā kā HTML pēdējie standarti piedāvā daudz dažādus specifiskus elementus, kuri nodrošina iespēju sadalīt lapu nepieciešamajās daļās, `<div>` elementu iesaka izmantot tikai tad, ja neviens cits elements konkrētajai lapas daļai nav piemērotāks par to. [15]
- **Elementam nav definēti visi obligātie atribūti** – parasti elementus var izmantot bez atribūtiem, bet ir tādi, kuriem atribūti ir ļoti svarīgi. Piemēram, ievietojot tīmekļa vietnē attēlu, `<img/>` elementam bieži tiek aizmirsts atribūts `alt`, kurš norāda, ar kādu tekstu aizvietot attēlu, ja tas netiek atrasts. Atribūta tekstam noteikti jābūt loģiskam un viegli saprotamam.
- **Atkārtoti izmantoti vienādi elementa `id` atribūti vienā lapā** – lai gan šādi veidojot tīmekļa vietnes lapu, tās struktūra netiek salauzta, funkcionāli vairāki vienādi `id` vienā lapā var sabojāt sākumā iecerēto lapu. Elementu `id` atribūtiem jābūt unikāliem. Ja tie lapā kaut kur atkārtojas, tos noteikti vajadzētu mainīt uz atribūtu `class`.
- **Nepareizi aizvērts elements** – izmantojot pareizi uzstādītu koda redaktoru, šādi kļūdai nevajadzētu parādīties, bet bieži vien izstrādātāji neizmanto redaktora pieejamo uzstādījumu, kurš automātiski uzraksta elementa beigas tad, kad tiek uzrakstīts tā sākums. Tā rodas kļūdas, kurās elementi tiek uzrakstīti nepareizi. Piemēram, elementam `<img/>` atverošā un aizverošā daļa ir kopā, bet elementam `<div></div>` ir divas daļas, kuras definē bloka sākumu un beigas. Nepareizi uzrakstot elementu, sekas var būt dažādas – sākot ar to, ka pārlūkprogramma pati automātiski interpretēs elementu pareizi, līdz tam, ka netiks parādīta kāda no lapas daļām.

Šajā nodaļā aprakstītās izstrādātāju visbiežāk pieļautās HTML un HTML5 kļūdas, protams, nav visas – tikai maza daļa no tām. Tāpēc ir svarīgi, ka izstrādātāji, pirms tīmekļa vietņu veidošanas, iemācās HTML pamatus un regulāri seko līdzi valodas standartu jaunumiem.

## 2.2.2. Visbiežāk pieļautās kļūdas, izmantojot CSS standartus

Lai gan tīmekļa vietnes pamatu veido HTML, ar CSS tiek izveidots tas, ko lietotāji redz. CSS ļauj norobežot dažādus elementus, izcelt tos, kuriem ir svarīgāka nozīmē.

Tāpat kā HTML standartiem, CSS valodai ar laiku ir izveidojušies vairāki standarti. Izstrādātājiem ir jāseko līdzi jebkuriem standartu jaunumiem, pārlicinoties, ka jaunā tīmekļa vietne tiek izstrādāta atbilstoši noteikumiem.

Nepārzinot CSS standartu pamatus un nesekojojot līdzi jaunumiem, izstrādātāji mēdz pieļaut dažādas kļūdas, kuras bieži traucē tīmekļa vietņu lietotājiem lapas skatīties pilnvērtīgi.

Šīs ir izstrādātāju visbiežāk pieļautās CSS izmantošanas kļūdas:

- **Elementi tiek atlasīti ļoti specifiski** – rakstot CSS, jāpārlicinās, ka elementi tiek atlasīti precīzi, bet bieži vien izstrādātāji ar to pārcenšas. Lai gan ir svarīgi kodu rakstīt precīzi, pārāk specifiska norāde uz elementu vēlāk traucē stilu pārrakstīt, ja tas ir nepieciešams. Tāpēc labāks variants ir rakstīt elementa atlasīšanas kodu precīzi, bet īsi; ja nepieciešams, HTML kodā izveidot id.
- **Stili tiek pārrakstīti, izmantojot !important** – lai gan ir gadījumi, kad nepieciešams izmantot !important aiz kāda no stiliem, tas ir diezgan reti. Visbiežāk to izmanto kādiem ļoti specifiskiem globāliem elementiem vai pārrakstot jau esošu stilu, kuram nav iespējams precīzāk specificēt elementu. Izmantojot !important visu stilu pārrakstīšanai, ar laiku var nonākt līdz situācijai, kad vairs nav iespējams kaut ko pārrakstīt, tāpēc izstrādātājiem jāņem vērā, ka !important vajadzētu lietot tikai tad, ja citu variantu nav.
- **Pārāk daudz tiek izmantoti pozicionēšanas stili** – ir svarīgi iemācīties, kā pareizi pozicionēt elementus, bet ar to nevajadzētu aizrauties. [16] Izstrādātāji mēdz izmantot CSS stilus position:absolute vai position:relative pārāk bieži vai pat nevietā. Izmantojot relatīvu elementa novietojumu nepareizi, pārlūkprogrammā Internet Explorer ļoti bieži salūzt vēlamie stili, jo vecākelementa pozīcija tiek interpretēta nepareizi. Tāpat arī ar absolūtu elementa novietojumu var rasties dažādas problēmas. Piemēram, pozicionējot elementu absolūti un novietojot to konkrētā vietā lapā vai kādā elementā, atbilstoši izstrādātāja monitora izmēriem, mainot ekrāna izmērus, elements visticamāk atradīsies kādā citā vietā un vēlamais lapas projektējums būs salūzis. Tāpēc elementu pozicionēšanas stilus būtu jāizmanto pēc iespējas retāk un tikai tad, ja tas ir nepieciešams.

- **Elementa izmēra definēšana, kad tas nav nepieciešams** – sākot veidot tīmekļa vietnes ar CSS, izstrādātājiem bieži vien rodas vēlme uzstādīt elementam noteiktu izmēru, tā panākot, ka elements izskatās tā, kā noteikts lapas projektējumā, un atrodas tur, kur tas vajadzīgs. Šāda darbība neko sliktu lapai nenodarīs tad, ja tās saturs būs noteikts un to nav plānots mainīt, bet gadījumos, kad nav precīzi zināms elementa izmērs, tā uzstādīšana var radīt situāciju, kad elements satur datus, kuri neietilpst noteiktajā izmērā, līdz ar to salaužot lapas struktūru – tīmekļa vietnes lietotājs redzēs elementu un tā datus, kuri izplūdis ārpus definētajām elementa robežām.
- **Vienādu stilu atkārtošana vienā failā** – tā kā elementi ir dažādu un to nozīme ir atšķirīga, izstrādātājiem liekas loģiski, katru elementu noformēt atsevišķi, bet ne vienmēr tas ir nepieciešams. Tā kā CSS failus parasti cenšas izstrādāt pēc iespējas mazākus un precīzākus, labāk būtu apkopot stilus, kuri atkārtojas. Tāpēc ir ieteicams vispirms noformēt globālus elementus, izmantojot specifiskas klases, elementus vai vienkārši elementu specifiskācijas atdalot ar komatu. Pēc tam, rakstot CSS kodu tālāk, ir jāpievērš uzmanība tam, vai stili pārāk bieži neatkārtojas. Ja tā notiek, arī tos vajadzētu apvienot. Protams, vienmēr ir jāapskata visi iespējamie varianti un jāizvēlas tas, kurš būs īsāks un piemērotāks.
- **Nepareizi izmantoti stili margin un padding** – izstrādātāji mēdz jaukt margin un padding mērķi un izmanto tos nepareizi. Margin paredzēts, lai uzliktu atstarpi ārpus elementa, savukārt padding paredzēts atstarpes ielikšanai elementa iekšpusē. Ja elementam nav redzamu robežu, margin un padding lietošana neatbilstoši to mērķim neko daudz nemainīs, bet ja izstrādātājs vēlāk mēģinās elementa robežu noformēt redzamu, var nākties mainīt visus elementa stilus.
- **Nepareizi izmantots stils float** – CSS ļauj novietot elementus dažādās vietās un tam bieži tiek izmantots stils float, bet ne vienmēr izstrādātāji padomā par to, ka elementu saturs var būt mainīgs. Piemēram, uzliekot vairākiem elementiem float stilu un pēc tam vienā no tiem ievietojot attēlu, kurš ir lielāks par šo elementu, pārlūkprogrammā Internet Explorer elementi, kuri atradīsies blakus šim elementam, tiks nobīdīti. Vēl svarīgi pēc elementiem, kuriem izmantots float ievietot elementu, kuram izmantots clear stils, noņemot float stila ietekmi nākošajiem elementiem.

Šajā nodaļā, protams, nav aprakstītas visas iespējamās kļūdas, kuras izstrādātāji var pieļaut, strādājot ar CSS, bet tika aprakstītas visbiežāk sastopamās.

Lai nodrošinātu pēc iespējas kvalitatīvāku kodu, svarīgi, lai izstrādātāji iemācītos CSS standartu pamatus un regulāri sekotu līdzi to jaunumiem. Tāpat svarīgi domāt globāli un tehniski, ņemot vērā, ka var mainīties lapas saturs, ekrāna izmērs vai pārlūkprogramma.

### 3. PĒTĪJUMS PAR HTML UN CSS KODU DAŽĀDĀS PĀRLŪKPROGRAMMĀS

Šajā nodaļā aprakstīts autora veiktais pētījums un apkopoti tā rezultāti. Pētījums sastāv no 5 savā starpā nesaistītām daļām, kuras katra aprakstīta kādā no apakšnodaļām, un secinājumu daļas, kurā apkopoti iegūtie rezultāti un izdarīti secinājumi.

Pētījuma mērķis bija apskatīti dažādu jauno HTML elementu un CSS atribūtu, no standartiem HTML5 un CSS3, lietošanas veidus un to attēlojumu dažādās pārlūkprogrammās, izmantojot vairākus atšķirīgus un loģiskus kodus. Tāpat kodā tika attēlotas dažas no izstrādātāju visbiežāk pieļautajām kļūdām, lai uzskatāmi būtu redzams, kā tīmekļa vietne ar tām tiek attēlota dažādās pārlūkprogrammās.

Lai izvairītos no tīmekļa vietnēm, kuras ļoti atšķirīgi izskatās uz dažādām pārlūkprogrammām, un dažādām HTML un CSS kļūdām, skaidri ir jāzina, kāds ir elementu vai atribūtu mērķis un kā tie tiek attēloti dažādās pārlūkprogrammās.

Elementi un atribūti kodā tika rakstīti, atbilstoši HTML un CSS standartiem. Ņemot vērā, ka ne visas pārlūkprogrammas atbalsta visus elementus un atribūtus, tie tika rakstīti atbilstoši W3C standartu piedāvātajiem uzlabojumiem, kuri nodrošina elementa vai atribūta attēlošanu attiecīgajā pārlūkprogrammā tā, kā tas nepieciešams.

Lai pētījumā izmantotais kods un rezultāti būtu vieglāk saprotami un pārskatāmi, tie tika sadalīti vairākās loģiskās daļās – vispirms galvenes un kājenes daļa, pēc tam dažādi pamata informācijas attēlošanas veidi. Tika veidots tīmekļa vietnes paraugs, neizstrādājot tās funkcionalitāti.

Viens un tas pats kods tika pārbaudīts pārlūkprogrammu Chrome, Firefox, Internet Explorer, Safari jaunākajās lietotājiem pieejamās versijās, izmantojot operētājsistēmu Windows 7. Šīs pārlūkprogrammas, balstoties uz lapā W3schools esošajiem statistikas datiem, ir populārākās.

Izpētītā koda rezultāti tika apkopoti katras pētījuma daļas beigās, tabulā aprakstot elementus, kuri tika pētīti, un rezultātus. Ja elements tiek atbalstīts pārlūkprogrammā un izskatās tā, kā nepieciešams, tabulas laukā iezīmēts "+", ja netiek atbalstīts vai attēlots nepareizi – iezīmēts "-" un, ja nepieciešams, papildus informācija.

Papildus pētījuma mērķis ir noskaidrot, kura no pārlūkprogrammām vissliktāk atbalsta jauno HTML un CSS standartu populārākos elementus.

### 3.1. Tīmekļa vietnes galvene

Izstrādājot tīmekļa vietnes, ļoti liela nozīme jāpievērš galvenes izskatam un projektējumam, jo tā parasti ir pirmā lapas daļa, kuru lietotājs redz. Tajā noteikti jāattēlo tīmekļa vietnes tēma, nosaukums vai logotipa attēls. Svarīgi norādīt saites uz galvenajām tīmekļa vietnes lapām, izmantojot ikonas un/vai galveno izvēlni. Lai lietotāji ērti varētu atrast visu nepieciešamo, jāpievērš arī meklētājs.

Tīmekļa vietnes galvenes kods redzams attēlā 3.1. “Galvenes kods”.

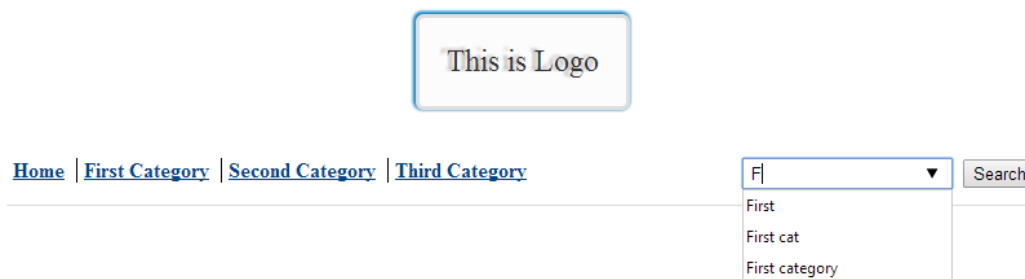
```
14 <body>
15 <section class="page">
16   <header>
17     <div class="logo">Lapas Logo</div>
18
19     <div class="search">
20       <input type="text" list="search-list" placeholder="Search.." />
21       <button type="button" class="search-button">Search</button>
22       <datalist id="search-list">
23         <option value="First">
24           <option value="First cat">
25           <option value="First category">
26           <option value="Second">
27           <option value="Third">
28         </datalist>
29     </div>
30
31     <nav>
32       <a href="home">Home</a>
33       <a href="first-category">First Category</a>
34       <a href="second-category">Second Category</a>
35       <a href="third-category">Third Category</a>
36     </nav>
37   </header>
```

3.1. att. “Galvenes kods”

Galvenes elementi tika izkārtoti, izmantojot CSS. Tā kā tie ir vienkārši un nav daudz, arī izmantotie stila atribūti ir vienkārši. Vienīgi logotipa izveidei izmantoti CSS3 rāmju un teksta noformēšanas atribūti, kuri to ļauj izcelt, radot iespaidu, ka izmantots attēls, skatīt attēlu 3.2. “Lapas galvene pārlūkprogrammā Chrome” Logotips visās pārlūkprogrammās tika attēlots vienādi.

Meklēšanas ievadlaukam piesaistīts elements datalist, kurš piedāvā meklēšanas ievadlaukā ievadāmo vērtību variantus. Tāpat ievadlaukam pievienots atribūts placeholder, kurš informē lietotāju.

Tā kā elements datalist vēl netiek atbalstīts pārlūkprogrammā Safari, tajās tas netika parādīts. Attēlā 3.2. “Lapas galvene pārlūkprogrammā Chrome” redzams, kā izskatās meklētāja ievadlauks ar atvērtu datalist elementu.



3.2. att. “Lapas galvene pārlūkprogrammā Chrome”

Lai gan datalist elements pārlūkprogrammā Safari netika parādīts, tas neizjauca tīmekļa vietnes galvenes projektējumu. Meklētāja ievadlauka funkcionalitāte nemainījās un lietotāji varēja ievadīt meklējamo frāzi, neizmantojot datalist atribūta piedāvātās vērtības.

Tīmekļa vietnes galvenē pētītie elementi, atribūti un to attēlojuma rezultāti aprakstīti tabulā 3.1. “Tīmekļa vietnes galvene dažādās pārlūkprogrammās, html”.

3.1. tabula “Tīmekļa vietnes galvene dažādās pārlūkprogrammās, html”

	<b>Chrome (34.0)</b>	<b>Firefox (29.0)</b>	<b>Safari (5.1.7)</b>	<b>Internet Explorer (10/11)</b>
<header>	+	+	+	+
Ievadlauka placeholder atribūts	+	+	+	+
<datalist>	+	+	-	+
<nav>	+	+	+	+

Tika apskatīti arī CSS3 atribūti, izmantojot tos galvenē redzamā logo noformēšanai, lai pēc iespējas vairāk tiktu izmantotas HTML un CSS piedāvātās iespējas.

CSS3 atribūtu attēlojums populārākajās pārlūkprogrammās aprakstīts tabulā 3.2. “Tīmekļa vietnes galvene dažādās pārlūkprogrammās, css”.

3.2. tabula “Tīmekļa vietnes galvene dažādās pārlūkprogrammās, css”

	<b>Chrome (34.0)</b>	<b>Firefox (29.0)</b>	<b>Safari (5.1.7)</b>	<b>Internet Explorer (10/11)</b>
border-radius	+	+	+	+
box-shadow	+	+	+	+
text-shadow	+	+	+	+

Gandrīz visi elementi un atribūti, kuri tika izmantoti galvenes izveidošanai, populārākajās pārlūkprogrammās attēloti pareizi, atbilstoši to specifikācijai. Vienīgais izņēmums ir elements `datalist`, kuru neatbalsta pārlūkprogrammā safari, bet tas definēts arī HTML5 standartos.

### 3.2. Tīmekļa vietnes kājene

Tīmekļa vietnes kājene ir vēl viena svarīga lapas daļa. Lai gan kājeni atšķirībā no galvenes lietotāji parasti neredz uzreiz, atverot tīmekļa vietni, tā atkārtojas katrā lapā un satur globālu informāciju, kura lietotājam varētu būt nepieciešama.

Tā pat kā galvenē, arī kājenē nav daudz elementu – tikai paši svarīgākie. Tā parasti sastāv no svarīgākajām saitēm uz kādu no tīmekļa vietnes lapām, noformētām neitrāli, lapas logo, bet mazāka nekā redzams galvenē, un autortiesībām. Lai lietotāji varētu pieteikties lapas jaunumiem, kājenē reizēm mēdz attēlot ievadlauku, kurā, ierakstot e-pasta adresi, tiem ir iespējams reģistrēties. Tāpat šajā lapas daļā mēdz novietot sīkus attēlus jeb ikonas, kurā pievienota saite uz kādu ārēju resursu, piemēram, tīmekļa vietnes īpašnieku tīmekļa dienasgrāmatu vai galeriju, kura tiek veidota un glabāta kādā citā vietnē.

Kājenes elementus parasti cenšas īpaši neakcentēt, tāpēc to stils ir ļoti vienkāršs un reti tiek izmantots CSS3, lai nodrošinātu kādu papildus efektu vai funkcionalitāti.

Tīmekļa vietnes kājenes kods redzams attēlā 3.3. “Kājenes kods”.

```
41 <footer>
42   <section class="links">
43     <ul class="info">
44       <li><a href="about">About us</a></li>
45       <li><a href="contacts">Contact us</a></li>
46     </ul>
47     <ul class="info">
48       <li><a href="articles">Our articles</a></li>
49       <li><a href="images">Our images</a></li>
50     </ul>
51   </section>
52
53   <section class="logo-in-footer">
54     <div class="logo small-logo">Page Logo smaller than large one</div>
55   </section>
56   <p class="copyrights">Our company copyrights, 2014</p>
57 </footer>
58 </section>
59 </body>
60 </html>
```

3.3. att. “Kājenes kods”

Tā kā kājenē esošie elementi ir vienkārši tāpat kā to stils, visām pārlūkprogrammām nav problēmu to attēlot tā, kā tas nepieciešams.

Attēlā 3.4. “Lapas kājene pārlūkprogrammā Chrome” redzams, kā kājenē izkārtoti un noformēti elementi.



3.4. att. “Lapas kājene pārlūkprogrammā Chrome”

Lai gan kājenē parasti ir vienkārši elementi un tāpat to stils, izstrādātāji bieži vien pieļauj dažādas kļūdas tieši šajā lapas daļā. Piemēram, uzstādot noteiktu izmēru kājenē esošajam logo, kurš ir mazāks par galvenē ievietoto, var rasties kļūda, ja lapas īpašnieks izlemj, ka kājenē esošajam logo jānomaina teksts. Attēlā 3.5. “Kājenes logo ar garu tekstu” redzams, kas notiek, ja elementam tiek uzstādīts noteikts izmērs un pēc tam pievienots saturs, kurš iepriekšējos izmēros neietilpst.

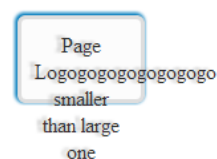
Tāpat redzams, ka vienā lapā atkārtojas divi gandrīz identiski elementi. Kā zināms, stilu atkārtošānās vienā failā tiek uzskatīta par kļūdu. Arī šajā gadījumā šāda kļūda varētu būt

iespējama, ja izstrādātājs nepadomā vai nezina, ka lapas galvenē ir tieši tāds pats elements, jo ļoti bieži galvenes un kājenes stilu kods failā neatrodas viens otram blakus.

- [About us](#)
- [Contact us](#)

- [Our articles](#)
- [Our images](#)

Our company copyrights, 2014



3.5. att. “Kājenes logo ar garu tekstu”

Lai samazinātu atkārtota koda iespējamību un nodrošinātu visā lapā vienotu stilu, ieteicams tīmekļa vietnes projektējumā pārskatīt jau sākumā un, pēc visu pieejamo failu izpētīšanas, vienam izstrādātājam ļaut aprakstīt un noformēt tos elementus, kuri lapās atkārtojas visbiežāk – galvene un kājene ir tikai daļa no šiem elementiem.

Tīmekļa vietnes kājenē pētītie elementi, atribūti un to attēlojuma rezultāti aprakstīti tabulā 3.3. “Tīmekļa vietnes kājene dažādās pārlūkprogrammās”.

3.3. tabula “Tīmekļa vietnes kājene dažādās pārlūkprogrammās”

	<b>Chrome (34.0)</b>	<b>Firefox (29.0)</b>	<b>Safari (5.1.7)</b>	<b>Internet Explorer (10/11)</b>
<footer>	+	+	+	+
float un clear	+	+	+	+
word-wrap: break-word;	+	+	+	+
word-break: break-all;	+	+	+	+
overflow: hidden;	+	+	+	+

Tā kā tīmekļa vietnes kājenē novietotie elementi ir tikai ar informatīvu nozīmi, to stils un attēlojums parasti ir ļoti vienkāršs. Arī šajā nodaļā pētītais kājenes kods bija diezgan vienkāršs un visi apskatītie elementi un atribūti populārākajās pārlūkprogrammās tika attēloti, atbilstoši to aprakstam standartos.

### 3.3. Tīmekļa vietnes multivides daļa

Tā kā lietotājiem parasti nav interesanti lasīt garus teksta gabalus, arvien biežāk tīmekļa vietnēs tiek izmantoti multivides elementi, lai nepieciešamo informāciju attēlotu dinamiski un interesanti lapas lietotājam. Visbiežāk tiek izmantoti skaņas vai video faili, un HTML5 šādu iespēju piedāvā, izmantojot audio vai video elementus.

Lai iegūtu elementa atskaņošanas materiālu, tiek izmantots src atribūts, kurā var norādīt gan ārēju saiti uz skaņas vai video resursu, gan uz servera esoša faila adresi. Tā kā skaņas vai video failā esošā informācija ne visiem ir interesanta, lietotājam noteikti jāpiedāvā iespēja multivides elementa atskaņošanu apturēt, tāpat svarīgas arī kontroles iespējas skaņai un video elementa izmēram. Visas šīs iespējas lietotājam var nodrošināt, izmantojot vienu audio vai video elementa atribūtu – controls. Tiek piedāvāti arī atribūti autoplay un loop, kuri attiecīgi ļauj skaņas vai video materiālu atskaņot automātiski, ielādējot lapu, vai nodrošināt tā atskaņošanu atkārtoti. Elementu atribūti nav obligāti, bet efektīvai to izmantošanai ieteicams izmantot vismaz vienu no diviem – controls vai autoplay. Ja netiek izmantots controls atribūts, skaņas elements nebūs redzams, lai nodrošinātu, ka tas tiek atskaņots, var pievienot atribūtu autoplay – elements tiks atskaņots, bet tas netiks parādīts lapā. Neizmantojot controls atribūtu video materiālam, tas tiks parādīts kā attēls, lai nodrošinātu, ka tas tiek atskaņots, jāpievieno atribūts autoplay.

Attēlā 3.6. “Multivides elementu kods” redzama multivides elementu izmantošanas iespējas un sintakse.

```
39 <main>
40   <section class="media">
41     <p>Listen to our latest sound..</p>
42     <audio controls autoplay loop>
43       <source src="example-sound.mp3" type="audio/mpeg">
44       Your browser does not support the audio element.
45     </audio>
46     <p>.. and here is the video example</p>
47     <video src="http://techslides.com/demos/sample-videos/small.mp4" controls autoplay loop>
48     Your browser does not support the <code>video</code> element.
49   </video>
50 </section>
51 </main>
```

3.6. att. “Multivides elementu kods”

Tā kā ne visas pārlūkprogrammas atbalsta video un audio elementus, noteikti jānorāda teksts, kurš paziņos, ka video vai skaņas elementu šajā pārlūkprogrammā nav iespējams parādīt. Tāpat var piedāvāt saiti, ja tā ir uz kādu ārēju vietni, uz vietu, kur skaņa vai video atrodas, lai lietotājam tiktu dota iespēja to mēģināt apskatīt vai noklausīties citur. Pārlūkprogrammas Safari

versija 5.1.7. neatbalsta audio un video elementus, tāpēc tiek parādīts elementā esošais teksts - skatīt attēlu 3.7. “Safari pārlūkprogramma neatbalsta multivides elementus”.

---

Listen to our latest sound..

Your browser does not support the audio element.

.. and here is the video example

Your browser does not support the video element.

---

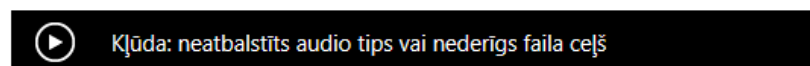
3.7. att. “Safari pārlūkprogramma neatbalsta multivides elementus”

Tāpat pārlūkprogramma var neatbalstīt formātu, kādā skaņa vai video pievienoti.

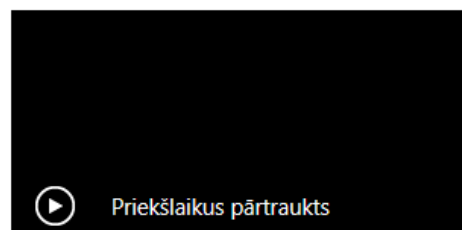
Dažām pārlūkprogrammām, piemēram, pārlūkprogrammas Internet Explorer versija 10 ir dažādi drošības uzstādījumi, tāpēc ir iespējams, ka tiks parādīts paziņojums, ka elementus nav iespējams atskaņot. Tas redzams attēlā 3.8. “Neatļauti multivides elementi”.

Pārlūkprogrammās Chrome un Firefox gan video, gan skaņas elementi tiek attēloti pareizi, tie tiek atskaņoti automātiski, atkārtoti un ir pieejamas elementu kontroles iespējas, jo izmantoti atribūti controls, autoplay un loop.

Listen to our latest sound..



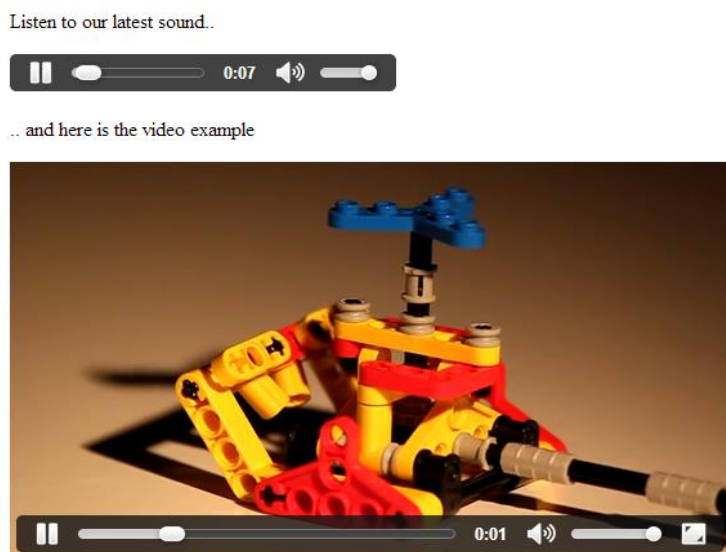
.. and here is the video example



3.8. att. “Neatļauti multivides elementi”

Pirms multivides elementu izmantošanas tīmekļa vietnes izstrādē, vajadzētu pārliicināties, ka tie tiek atbalstīti visās populārākajās pārlūkprogrammas, lai lietotājam nav pārsteigums, atverot lapu un ieraugot, ka video vai skaņa netiek atbalstīti.

Attēlā 3.9. “Multivides elementu attēlojums pārlūkprogrammā” var redzēt, kā tīmekļa vietnē parasti tiek attēloti skaņas un video materiāli, ja tiek atbalstīti gan HTML5 elementi, gan piedāvātie failu tipi.



3.9. att. “Multivides elementu attēlojums pārlūkprogrammā”

Ja lapā tiek izmantoti multivides elementi skaņas un video attēlošanai, jāņem vērā, ka to noformēšana, izmantojot CSS standartus, ir diezgan ierobežota. Elementus var pozicionēt, tiem var uzstādīt izmēru, pievienot rāmju stilu atribūtus. Lai gan CSS izmantošanas iespējas šiem elementiem ir ierobežotas, parasti bez tām ir iespējams iztikt, jo pārlūkprogrammu piedāvātais stils ir pietiekami vienkāršs un spēj iekļauties dažādos projektējumos. Protams, vienmēr ir iespēja izmantot Javascript funkcijas, lai piešķirtu elementiem vēlamo izskatu un atbrīvotos no šķietami liekām detaļām.

Tīmekļa vietnes multivides satura daļā pētītie elementi, atribūti un to attēlojuma rezultāti aprakstīti tabulā 3.4. “Tīmekļa vietnes multivides daļa dažādās pārlūkprogrammās”.

3.4. tabula “Tīmekļa vietnes multivides daļa dažādās pārlūkprogrammās”

	Chrome (34.0)	Firefox (29.0)	Safari (5.1.7)	Internet Explorer (10/11)
<audio> (mp3)	+	+	- netiek atbalstīts	+
<video> (mp4)	+	+	- netiek atbalstīts	- tiek atskaņots, bet bez attēla

Pētījuma rezultātā redzams, ka skaņas un video elementus pilnībā atbalsta tikai pārlūkprogrammas Chrome un Firefox.

Lai gan W3C dokumentācijās teikts, ka skaņas materiālus ar tipu mp3 un video materiālus ar tipu mp4 būtu jāatbalsta arī Safari pārlūkprogrammai, attēlā 3.7. “Safari pārlūkprogramma neatbalsta multivides elementus” skaidri redzams, ka tā tomēr nav. Iespējamais iemesls varētu būt tāds, ka oriģināli jaunākā pārlūkprogrammas Safari versija ir 7, bet tā vēl netiek atbalstīta operētājsistēmā Windows 7.

Arī pārlūkprogramma Internet Explorer neattēlo skaņas un video elementus uzreiz tā, kā tas ir paredzēts. Vispirms tie tiek aizliegti, balstoties uz pārlūkprogrammas uzstādījumiem, un tikai pēc tam, kad tiek nospiesta poga, kura apstiprina, ka elementus drīkst atskaņot, tie tiek parādīti. Diemžēl kvalitatīvi tika atskaņots tikai skaņas elements. Video elements tika atskaņots, bet video attēls netika parādīts. Tā vietā bija melns ekrāns tur, kur vajadzētu būt video attēlam.

Rezultātā var secināt, ka video un skaņas elementi no HTML5 standarta vēl nav piemēroti tīmekļa vietņu izstrādei, jo tie pilnībā tiek atbalstīti tikai divās pārlūkprogrammās – Chrome un Firefox. Ja ir ļoti nepieciešams šos elementus izmantot, būs jāraksta Javascript funkcijas un jānodrošina vēl citi papildus uzstādījumi, lai pārlicinoši elementu saturs tiktu parādīts arī pārlūkprogrammās Safari un Internet Explorer.

### **3.4. Tīmekļa vietnes informatīvais saturs**

Ejot laikam, tīmekļa vietņu projektējumi kļūst arvien dinamiskāki un orientēti vairāk uz lietotāja izklaidēšanu. Kaut gan dažādi attēli, video un skaņas materiāli aizņem lielu daļu no mūsdienīgām tīmekļa vietņu lapām, nedrīkst aizmirst arī par informācija, kura attēlota vienkārša teksta veidā.

Lai lietotājam nebūtu garlaicīgi, lasot piedāvāto tekstu, izstrādātāji mēdz to sadalīt vairākās loģiskās daļās, izveidojot pēc iespējas ērtāku un dinamiskāku tā izskatu. HTML5 standartā piedāvā vairākus elementus, kuri nodrošina dažādu funkcionalitāti, lai noformētu tekstu atbilstoši tā saturam un lietotāja vajadzībām.

Visbiežāk informācija teksta veidā tiek izmantota dažādās elektroniskajās dienasgrāmatās, ziņu un citās informatīvajās tīmekļa vietnēs. Lai dažādi raksti tiktu loģiski atdalīti viens no otra, ir iespējams izmantot article elementu. Tā iespējams tekstam pievienot stilu un visus rakstus lapā noformēt vienoti, izmantojot vienkāršu elementa atlasītāju, bez klasēm vai id. Vēl ļoti bieži tīmekļa vietnēs tiek izmantota funkcionalitāte, ka tiek rādīti tikai informācijas bloku virsraksti, uz kuriem noklikšķinot ar peles kursoru, tiek atvērta papildus informācija.

Lai lietotājam norādītu, cik lielu daļu no piedāvātā teksta jau ir izlasīts vai pārskatīts, var izmantot HTML5 elementu progress. To var pievienot atkārtoti pēc noteikta rakstu skaita, izmainot value atribūtu, vai rādīt vienmēr konkrētā lapas vietā un, atbilstoši pozīcijai, kādā lasītājs atrodas lapā, ar Javascript funkciju mainīt elementa value vērtību. Progress elementu var izmantot, lai attēlotu kādas darbības progresu. Tam tiek norādīti divi galvenie atribūti – value, kas norāda, cik daudz no progresā jau ir izpildīts, un max, kas norāda maksimālo progresā vērtību. Norādot progresu procentuāli, maksimālā vērtība parasti ir 100, bet tik pat labi tajā var norādīt lasāmās grāmatas lappušu skaitu vai raksta nodaļu skaitu.

Kodu, kā informācija teksta veidā tiek attēlota var apskatīt attēlā 3.10. “Tīmekļa vietnes informatīvā satura kods”.

```

39 <main>
40 <section class="information">
41 <progress value="25" max="100"></progress>
42 <article>
43 <h2>New article</h2>
44 <p>
45 Vix ei meliore menandri urbanitas.
46 Nec no facer malorum partiendo, bonorum ponderum ad mel,
47 nam aeterno evertitur efficiantur at.
48 Vix appareat theophrastus id, cu per oblique convenire,
49 ei cum iusto propriae. Te decore iisque argumentum sit, nam ad modo soleat nostro.
50 </p>
51 </article>
52 <details>
53 <summary>Information about something</summary>
54 <p>
55 Vix ei meliore menandri urbanitas.
56 Nec no facer malorum partiendo, bonorum ponderum ad mel,
57 nam aeterno evertitur efficiantur at.
58 Vix appareat theophrastus id, cu per oblique convenire,
59 ei cum iusto propriae. Te decore iisque argumentum sit, nam ad modo soleat nostro.
60 </p>
61 </details>
62 </section>
63 </main>

```

3.10. att. “Tīmekļa vietnes informatīvā satura kods”

Elements progress netiek rādīts pārlūkprogrammā safari, bet Chrome, Firefox un Internet Explorer tas tiek attēlots, atbilstoši pārlūkprogrammas iebūvētajiem stiliem. Pārlūkprogramma Chrome attēlo progresā lauku ar aptuvenām iedaļām, lai lietotājam būtu vieglāk orientēties, cik liela daļa no tā jau ir aizpildīta. To nodrošina pārlūkprogrammas iebūvētie stili, kuri uzstādīti konkrētiem elementiem.

Kā elements progress tiek attēlots pārlūkprogrammā Chrome var apskatīt attēlā 3.11. “Progresa elements pārlūkprogrammā Chrome”.



3.11. att. “Progresa elements pārlūkprogrammā Chrome”

Izmantojot HTML5 elementu details, kurš ļauj attēlot informāciju, lietojot funkcionalitāti, ka tiek rādīts tikai virsraksts un pārējais teksts atklāts tikai pēc noklikšķināšanas uz tā, jāņem vērā, ka tas vēl netiek atbalstīts lielākajā daļā pārlūkprogrammu. Tur, kur šis elements netiek atbalstīts, virsraksts un teksts tiks rādīti kā vienkārši virsraksta un teksta elementi. Šobrīd details elementu atbalsta tikai pārlūkprogramma Chrome un Safari versija 6.

Attēlā 3.12. “Atvērts detalizēta teksta elements pārlūkprogrammā Chrome” iespējams apskatīt, kā izskatās elements details, kad tas ir atvērts, noklikšķinot uz virsraksta ar peles kursoru, pārlūkprogrammā, kurā šis elements tiek atbalstīts un ir ar nepieciešamo funkcionalitāti.

▼ Information about something

Vix ei meliore menandri urbanitas. Nec no facer malorum partiendo, bonorum ponderum ad mel, nam aeterno evertitur efficiantur at. Vix appareat theophrastus id, cu per oblique convenire, ei cum iusto propriae. Te decore iisque argumentum sit, nam ad modo soleat nostro.

3.12. att. “Atvērts detalizēta teksta elements pārlūkprogrammā Chrome”

Izmantojot article elementu lapas rakstu attēlošanai, nav jāuztraucas par to, ka tas varētu tikt attēlots nepareizi, jo pat tad, kad tiks izmantota pārlūkprogramma, kura neatbalsta šo elementu, tā saturs tiks attēlots tā, kā paredzēts. Tikai tad jāņem vērā, ka nebūs pieejama HTML5 elementa funkcionalitāte un tā, ja nepieciešams, būs jānodrošina ar Javascript funkciju.

Attēlā 3.13. “Tīmekļa vietnes informatīvais saturs pārlūkprogrammā Firefox” var aplūkot, kā attiecīgi elementi progress, article un details tiek attēloti pārlūkprogrammas Firefox versijā 29.0.1.

Redzams, ka elements progress ir ar īpašu pārlūkprogrammas Firefox stilu, kurš tiek iebūvēts katrā pārlūkprogrammā atsevišķi.

Tāpat var pamanīt, ka netiek atbalstīts elements details, jo informācija tiek attēlota vienkārša teksta veidā bez nepieciešamās funkcionalitātes.



#### New article

Vix ei meliore menandri urbanitas. Nec no facer malorum partiendo, bonorum ponderum ad mel, nam aeterno evertitur efficiantur at. Vix appareat theophrastus id, cu per oblique convenire, ei cum iusto propriae. Te decore iisque argumentum sit, nam ad modo soleat nostro.

Information about something

Vix ei meliore menandri urbanitas. Nec no facer malorum partiendo, bonorum ponderum ad mel, nam aeterno evertitur efficiantur at. Vix appareat theophrastus id, cu per oblique convenire, ei cum iusto propriae. Te decore iisque argumentum sit, nam ad modo soleat nostro.

### 3.13. att. "Tīmekļa vietnes informatīvais saturs pārlūkprogrammā Firefox"

Lai gan pārlūkprogramma Firefox neatbalsta HTML5 elementu details, tas nesabojā lapas saturu – viss teksts tiek attēlots, atdalot virsrakstu no pārējās teksta daļas.

Ja tīmekļa vietnes teksta attēlošanai tiek izmantoti elementi, kurus vēl neatbalsta lielākā daļa pārlūkprogrammu, nepieciešamajai funkcionalitātei jāizmanto Javascript funkcijas, vai jāpārlicinās, ka lietotājiem netraucēs tas, ka teksts ir bez dinamiska satura un tas tiek attēlots atbilstoši parasta teksta attēlošanas standartiem, ņemot vērā pārlūkprogrammas iebūvētos stilus.

Tīmekļa vietnes informatīvā satura pētītie elementi, atribūti un to attēlojuma rezultāti aprakstīti tabulā 3.5. "Tīmekļa vietnes informatīvais saturs dažādās pārlūkprogrammās".

3.5. tabula "Tīmekļa vietnes informatīvais saturs dažādās pārlūkprogrammās"

	Chrome (34.0)	Firefox (29.0)	Safari (5.1.7)	Internet Explorer (10/11)
<progress>	+	+	-	+
<article>	+	+	+	+
<details>	+	-	-	-

Pētījuma rezultātā redzams, ka visas pārlūkprogrammas atbalsta tikai elementu article. HTML5 elements progress netika attēlots pārlūkprogrammā Safari, jo operētājsistēma Windows 7 izmanto pārlūkprogrammas Safari versiju 5.1.7., bet elements pilnībā tiek atbalstīts tikai no versijas 6, kura noteikti ir pieejama ierīcēs ar operētājsistēmu Mac OS.

Vissliktāk tiek atbalstīts elements details. Tas tiek parādīts visās pārlūkprogrammās, bet funkcionalitāte pieejama tikai pārlūkprogrammā Chrome. Tāpat kā elements progress, pārlūkprogrammā Safari details elements tiek atbalstīts tikai no versijas 6.

### 3.5. Tīmekļa vietnes informācijas bloku izkārtojums

Atverot tīmekļa vietni, lietotājam tiek parādīta tā lapas daļa, kura ietilpst ekrānā, sākot ar galvenes daļu, ja vien nav norādīts citādāk. Tīmekļa vietnes īpašniekiem, protams, ir vēlme, lai lietotājam šajā daļā tiktu parādīta visa nepieciešamā informācija, lai būtu skaidrs, kāda tīmekļa vietne ir atvērta un ko tā piedāvā. Lai to īstenotu, noder iespēja sakārtot lapas saturu loģiskos elementus, attiecīgi tos izvietojot.

HTML5 koda fragments ar vairākiem informācijas blokiem, kurš tika pētīts dažādās pārlūkprogrammās, apskatāms attēlā 3.14. “Tīmekļa vietnes informācijas bloku kods”.

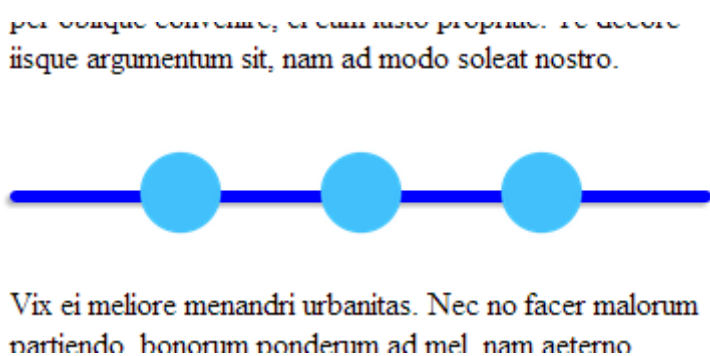
```
39 <main>
40   <section class="left">
41     <article>
42       <h2>New article</h2>
43       <p...>
50     </article>
51     <article>
52       <h2>New article</h2>
53       <p...>
60     </article>
61   </section>
62   <section class="middle">
63     <h1>Page Main Title</h1>
64     <p...>
71     <section class="middle-blocks">
72       <div class="block">
73         <figure>
74           
75         </figure>
76       </div>
77       <div class="block"...>
82       <div class="block"...>
87       <div class="middle-line"></div>
88     </section>
89     <p...>
96   </section>
97   <section class="right">
98     <figure>
99       
100    </figure>
101    <p>Right side images</p>
102    <figure...>
105  </section>
106 </main>
```

3.14. att. “Tīmekļa vietnes informācijas bloku kods”

Diezgan bieži tīmekļa vietnēs redzams izkārtojums, kurš ietver sevī trīs kolonnas – labās un kreisās puses un vidējo. Svarīgākā informācija parasti tiek attēlota vidējā kolonna, bet sānu kolonnas noder, lai piedāvātu lietotājam dažādas papildus iespējas.

Tapat lapā tiek attēloti dažādi attēli – fotogrāfijas, diagrammas, koda fragmenti u.c. HTML5 šāda satura attēlošanai piedāvā elementu `figure`, kurš raksturo ilustratīvu saturu iekļautu kādā lapas loģiskajā daļā, visbiežāk tekstā. Elements `figure` tiek izmantots kādas informācijas paskaidrošanai. Tā kā tas tiek iekļauts kādā citā elementā, kas ir lapas loģiskā daļa, visās pārlūkprogrammās tam ir iebūvētais stils, kas nodrošina, ka tas tiek rādīts ar nelielu atkāpi no galvenā satura. Attēlu elementiem `img`, kuri ir ļoti lieli, ieteicams uzstādīt noteiktu izmēru, nesabojājot attēla proporcijas, vai vislabāk – izveidot PHP vai Javascript funkciju, kura izmainīs attēla izmēru, pirms tas tiek ielādēts.

Lai padarītu tīmekļa vietnes elementu izkārtojumu interesantāku un ērtu lietotājam, ieteicams izmantot dažādus pozicionēšanas atribūtus, kurus piedāvā CSS. Kodā, kurš tika pētīts šīs nodaļas ietvaros, lapas vidējā daļā ir izvietoti 3 `figure` elementi, kuri satur attēlus. Zem šiem attēliem atrodas elementus, kurš attēlots kā līnija, skatīt attēlu 3.15. “Vidējās kolonnas ilustrācijas elementi”.



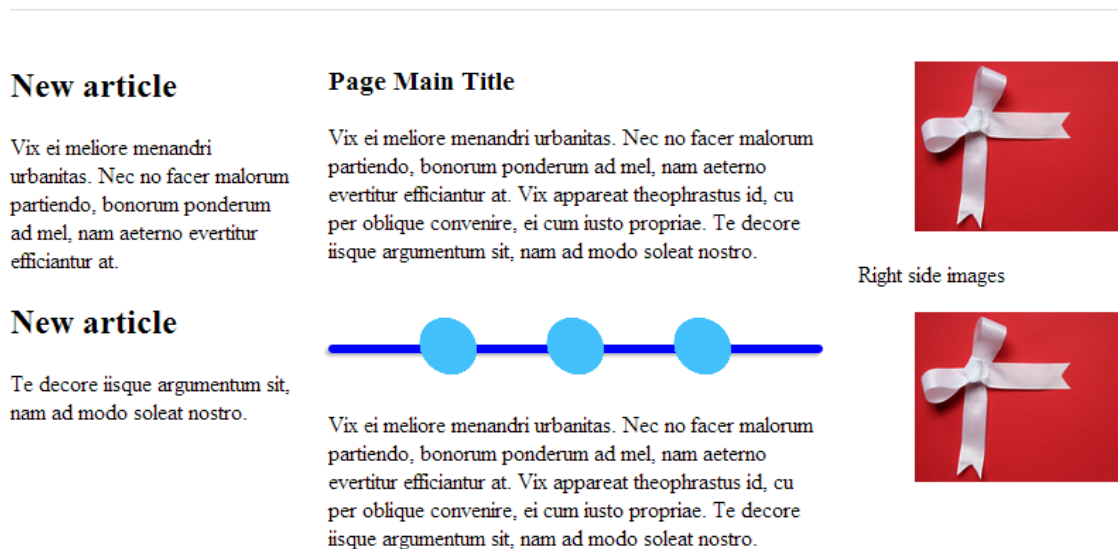
3.15. att. “Vidējās kolonnas ilustrācijas elementi”

Lai gan izskatās, ka līnija atrodas aiz elementiem, attēlā 3.14. “Tīmekļa vietnes informācijas bloku kods” redzams, ka elements ar klasi `middle-line` atrodas tieši pēc elementiem ar klasi `block`, kuri satur attēlus. Šāds efekts panākts, izmantojot CSS standartu piedāvātās pozicionēšanas iespējas. Līnija ir elementa rāmis ar apaļiem stūriem un ēnu, kura pozicionēta absolūti. Šādi aplū attēliem vajadzētu atrasties zem līnijas, bet atribūts `z-index` ļauj tos novietot augstāk, attiecībā pret līniju un elementu, kurš to satur. Tāpat jāpārlicinās, ka elements, kurš satur aplū attēlus un līniju, un aplū elementi ir pozicionēti relatīvi.

Vēl attēlu elementiem var izmantot dažādus animāciju vai stila pāreju atribūtus, lai padarītu to interesantākus. Piemēram, var izmantot transform atribūtu, lai mainītu attēlu formu. Izmantojot šo atribūtus, jāņem vērā, ka tas tiek atbalstīts tikai pārlūkprogrammās Firefox un Internet Explorer ar versiju 10. Lai nodrošinātu, ka CSS elementu pārveidošanas atribūts strādā arī citās pārlūkprogrammās, jāizmanto atribūta nosaukuma pielikumi – “-ms-”

pārlūkprogrammas Internet Explorer versijai 9 un “-webkit-” pārlūkprogrammām Chrome un Safari.

Atbilstoši pozicionējot dažādas tīmekļa vietnes lapas daļas, var iegūt jebkādu vēlamu izkārtojumu. Šajā nodaļā pētītais kods tika izkārtot trīs kolonnās – kreisajā kolonnā dažādi papildus raksti, labajā kolonnā attēli un informācija, vidus kolonnā galvenais raksts ar attēliem. Koda attēlojumu pārlūkprogrammā Chrome var apskatīt attēlā 3.16. “Tīmekļa vietnes informācijas bloki pārlūkprogrammā Chrome”.



3.16. att. “Tīmekļa vietnes informācijas bloki pārlūkprogrammā Chrome”

Izvietojot tīmekļa vietnes loģiskos blokus, vienmēr obligāti jāpārlicinās, ka tie izskatās vienādi labi visās nepieciešamajās pārlūkprogrammās. Ja kāds no blokiem tiek attēlots nepareizi vai izjauc kāda cita bloka struktūru, jāmēģina meklēt alternatīvi risinājumi vai kļūdu labojumi.

Šajā pētījuma daļā vairāk tika vērsta uzmanība uz to, kā tiek izkārtoti tīmekļa vietnes informatīvie bloki, jo viena no visbiežāk pieļautajām izstrādātāju kļūdām ir pozicionēšanas atribūtu nepareiza izmantošana, kas reizēm pārlūkprogrammām traucē attēlot elementus un informāciju tā, kā nepieciešams. Bloku izkārtojuma CSS kodu var apskatīt attēlā 3.17. “Tīmekļa vietnes informācijas bloku CSS kods”.

Tīmekļa vietnes informatīvo bloku izkārtojuma pētītie elementi, atribūti un to attēlojuma rezultāti aprakstīti tabulā 3.6. “Tīmekļa vietnes informatīvie bloki un to izkārtojums dažādās pārlūkprogrammās”.

```

42 main .left,
43 main .middle { float:left; }
44 main .right { float:right; }
45 main .left,
46 main .right { width:200px; }
47 main .middle { margin:0 25px; width:350px; }
48 main .middle .middle-blocks { float:left; width:100%; }
49 main .middle p { clear:both; }
50 main .middle-blocks .block {
51     float:left; width:50px; margin:0 20px; position:relative; z-index:10;
52     transform:skewX(7deg); -webkit-transform:skewX(7deg);
53 }
54 main .middle-blocks { position:relative; float:left; }
55 main .middle-blocks .middle-line {
56     position:absolute; width:100%; left:0; top:40px;
57     border-top:6px solid blue; border-radius:10px; box-shadow:-2px 2px 2px #bbb;
58 }

```

3.17. att. “Tīmekļa vietnes informācijas bloku CSS kods”

Bloku izveidošanai tika izmantots HTML5 elements section, jo vēl viena populāra izstrādātāju kļūda ir elementa div pārāk bieža un bez iemesla izmantošana.

3.6. tabula “Tīmekļa vietnes informatīvie bloki un to izkārtojums”

	Chrome (34.0)	Firefox (29.0)	Safari (5.1.7)	Internet Explorer (10/11)
<section>	+	+	+	+
<figure>	+	+	+	+
position: relative;	+	+	+	+
position: absolute;	+	+	+	+
transform: skewX(7deg);	- jāizmanto atribūta pielikums “-webkit-”	+	- jāizmanto atribūta pielikums “-webkit-”	+

Visas pārlūkprogrammas attēloja elementus un atribūtus tā, kā tas bija nepieciešams. Tika pārlūkprogrammās Chrome un Safari CSS atribūtam transform bija nepieciešams pielikums “-webkit-”, lai varētu kvalitatīvi attēlot elementa pārveidojumu.

Arī pozicionējot elementus, tīmekļa vietnes izskats netika sabojāts un visās pārlūkprogrammās bloki tika novietoti tur, kur tas nepieciešams. Protams, pozicionējot elementus, tika ņemti vērā noteikumi, ka pēc elementiem, kuri pozicionēti ar atribūtu float, jālieto elements ar atribūtu clear.

### 3.6. Pētījuma rezultāti

Populārākajās pārlūkprogrammās tika apskatīti dažādi kodi, kuri attēloja tīmekļa vietnēs visbiežāk izmantotos loģiskos informācijas blokus. Tika apskatīts kods galveni, kājēnei, lapas multivides elementiem, informatīvajai daļai un informatīvo bloku izkārtojumam.

Katrai no pētāmajām daļā tikai izveidots struktūra, izmantojot HTML kodu, un projektējums, stils, izmantojot CSS kodu. Gan HTML, gan CSS tika rakstīti, ņemot vērā jaunākos tīmekļa standartus un pēc iespējas vairāk tika izmantoti jaunie elementi, jo pētījuma galvenais mērķis bija noskaidrot, kā šie jaunie elementi tiek attēloti dažādās pārlūkprogrammās. Vēl pētījuma uzdevums bija apgāzt vai pierādīt teoriju, ka vissliktāk elementi un atribūti no jaunajiem tīmekļa standartiem tiek attēloti pārlūkprogrammā Internet Explorer.

Pētījuma rezultātā var secināt, ka svarīga ir ne tikai pārlūkprogramma, bet arī lietotāja operētājsistēma. Šis pētījums tika veikts uz datora, kurā ir operētājsistēma Windows 7. Tā kā šī operētājsistēma atbalsta pārlūkprogrammas Safari versiju 5.1.7., bet jaunākā oficiālā pārlūkprogrammas versija ir 7, nebija iespējams apskatīt elementus, kuri tiek atbalstīti no pārlūkprogrammas Safari versijas 6. Piemēram, elementus details un progress.

Visās pārlūkprogrammās tika atbalstīti HTML5 semantiskie elementi, piemēram, header, footer, main, nav. Tas nozīmē, ka to izmantošana tīmekļa vietņu izstrādē ir droša, jo tie tiks attēloti vienādi labi visās populārāko pārlūkprogrammu jaunākajās versijās.

Apkopojot rezultātus par pārlūkprogrammām, kuras vissliktāk atbalsta un attēlo jaunākos HTML un CSS pēdējo oficiālo standartu elementus, var secināt, ka teorija par to, pirmajā vietā no visām ir pārlūkprogramma Internet Explorer, ir nepareiza. Šī pētījuma rezultāti rāda, ka vissliktāk elementus un atribūtus atbalsta un attēlo pārlūkprogramma Safari. Protams, jāņem vērā, ka šī pārlūkprogramma ir izstrādāta, lai to pamatā izmantotu operētājsistēmā Mac OS, bet pētījums tika veikts, izmantojot operētājsistēmu Windows 7.

## SECINĀJUMI

Apkopojot atrasto informāciju par tīmekļa standartiem un izanalizējot pētījuma rezultātus, tiek secināts, ka darba mērķi ir izpildīti. Pēc pētījumu rezultātiem tiek secināts, ka izvirzītā hipotēze nav pareiza.

Noskaidrots, ka jaunākās tīmekļa standartu versijas ir izstrādātas tā, lai tīmekļa vietņu izstrādi padarītu pēc iespējas vienkāršāku, ļaujot tās pamatu maksimāli veidot, izmantojot tikai HTML un CSS. Standartu piedāvātie elementi un atribūti radīti, apkopojot informāciju par tīmekļa vietnēs visbiežāk izmantotajām metodēm.

Tika apkopotas izstrādātāju visbiežāk pieļautās kļūdas, ņemot vērā internetā pieejamo informāciju un darba autora personīgo pieredzi, kura iegūta, uzraugot un pārskatot jauno tīmekļa vietņu izstrādātāju darbu. Tā pat tika apkopota informācija par iespējām izmantot jaunākos tīmekļa standartus, lai izstrādātu elektroniskās vēstules.

Veiktajā pētījumā apskatīts, kā tiek attēloti dažādi koda fragmenti, un izpētītas iespējas, kā jaunākās HTML un CSS standartu versijas izmantot tīmekļa vietņu izstrādē, izmantojot populārākās no pārlūkprogrammām. Pētījuma rezultātā var secināt, ka lielai daļai no jaunākajiem HTML un CSS standartu elementiem un atribūtiem ir daļējs atbalsts populārākajās pārlūkprogrammās – tie tiek attēloti pareizi, atbilstoši to aprakstam tikai dažās no pārlūkprogrammām vai noteiktās to versijās. Bet standarti tiek regulāri atjaunoti, tāpat arī pārlūkprogrammas un arvien vairāk elementu tiek atbalstīti pārlūkprogrammās un tiek attēloti, atbilstoši to mērķim un aprakstam.

Kopumā var secināt, ka darbā aprakstītā tēma ir aktuāla un noderīga. Tā kā viena no darba sākumā izteiktajām problēmām bija jauno tīmekļa vietņu izstrādātāju nepietiekamās zināšanas par tīmekļa standartiem, darbs var kalpot kā uzziņas un mācību materiāls.

## IZMANTOTĀ LITERATŪRA

1. Tīmekļa lapušu publicēšana un veidošanas pamatprincipi  
<http://www.uzdevumi.lv/ExerciseRun/RunExercise?exerciseId=080294ef-06e3-4c8f-a193-bae65663bd34>
2. HTML, <http://lv.wikipedia.org/wiki/HTML>
3. HTML ievads, <http://www.w3.org/TR/html/introduction.html#introduction>
4. HTML vēsture, <http://www.w3.org/People/Raggett/book4/ch02.html>
5. A short history of HTML, <http://w3c-html.com/html-history.html>
6. XHTML, <http://lv.wikipedia.org/wiki/XHTML>
7. HTML5 ievads , <http://www.w3.org/TR/2011/WD-html5-diff-20110405/#introduction>
8. Ievads CSS, <http://coding.lv/read/01-ievads-css>
9. Cascading Style Sheets, [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
10. HTML4 and HTML5 tutorial, <http://www.w3schools.com/html/>
11. Ian Devlin “HTML5 multimedia develop and design”. Peachpit Press, 2012., 10.lpp
12. Ian Devlin “HTML5 multimedia develop and design”. Peachpit Press, 2012., 23.lpp
13. CSS tutorial, <http://www.w3schools.com/css/>
14. Introducing HTML5 and CSS3, <http://www.graphics.com/article-old/introducing-html5-and-css3>
15. HTML structural elements, [http://www.w3.org/wiki/HTML\\_structural\\_elements](http://www.w3.org/wiki/HTML_structural_elements)
16. The most common HTML and CSS mistakes to avoid,  
<http://webdesignledger.com/tips/the-most-common-html-and-css-mistakes-to-avoid>
17. How to create great HTML emails with CSS,  
<http://www.htmlgoodies.com/beyond/css/article.php/3679231>
18. Support for HTML5 elements in email,  
<http://www.campaignmonitor.com/blog/post/4040/support-for-html-5-elements-in-email>
19. HTML5 email – are we ready?, <http://www.clickz.com/clickz/column/2304045/html5-email-are-we-ready>
20. CSS3 animation in email, <http://www.pure360.com/blog/blog-entries/CSS3-Animation-in-email>
21. CSS3 support in email clients,  
<http://www.campaignmonitor.com/blog/post/3107/css3-support-in-email-clients/>
22. <http://www.termnet.lv/>

## DOKUMENTĀRĀ LAPA

Bakalaura darbs „Tīmekļa standartu atbilstības pārbaude dažādām pārlūkprogrammām”  
izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie  
informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors:

Līvija Golosujeva

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītājs: Dr. sc. comp. Jānis Zuters

Recenzents: Dr.dat., prof. Māris Vītiņš

Darbs iesniegts Datorikas fakultātē

Dekāna pilnvarotā persona:

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

Komisija: