

Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

DELO S SPLETOM

Melita Kompolšek



Učbenik za 1. letnik srednjega strokovnega izobraževanja

TEHNIK RAČUNALNIŠTVA

Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

Melita Kompolšek

DELO S SPLETOM

Učbenik za 1. letnik srednjega strokovnega izobraževanja

TEHNIK RAČUNALNIŠTVA

Melita Kompolšek, Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

DELO S SPLETOM

Učbenik za 1. letnik srednjega strokovnega izobraževanja TEHNIK RAČUNALNIŠTVA

Strokovni recenzenti:

dr. Uroš Breskvar, Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

doc. dr. Samo Simončič, Fakulteta za strojništvo, Univerza v Ljubljani

Aleš Volčini, Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

Jezikovna recenzentka:

Marjeta Šušteršič Menart, Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

© Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana

Vse pravice pridržane. Noben del tega dela ne sme biti reproduciran ali prepisan v katerikoli obliki oziroma na katerikoli način – elektronsko, mehansko, s fotokopiranjem ali kako drugače – brez predhodnega dovoljenja lastnikov avtorskih pravic.

CIP - Kataložni zapis o publikaciji

Narodna in univerzitetna knjižnica, Ljubljana

004.22:621.3.037.3(075.3)(0.034.2)

KOMPOLŠEK, Melita

Delo s spletom [Elektronski vir] : učbenik za 1. letnik srednjega strokovnega izobraževanja Tehnik računalništva / Melita Kompolšek. - El. knjiga. - Ljubljana : Elektrotehniško-računalniška strokovna šola in gimnazija, 2016

Način dostopa

(URL): http://moodle.vegova.si/pluginfile.php/38529/mod_resource/content/1/Delo%20spletom.pdf

ISBN 978-961-93617-5-7 (pdf)

286861056

KAZALO

1	LOGIČNE OPERACIJE IN BOOLOVA ALGEBRA.....	13
1.1	Povzetek.....	15
1.2	Vprašanja in naloge za preverjanje znanja	15
2	KODIRANJE PODATKOV	16
2.1	Digitalni sistemi.....	16
2.2	Analogni sistemi	16
2.3	Številski sistemi.....	17
2.3.1	Desetiški sistem.....	17
2.3.2	Dvojiški oz. binarni sistem.....	17
2.3.2.1	Pretvorba iz dvojiškega v desetiški sistem.....	18
2.3.2.2	Pretvorba iz desetiškega v dvojiški sistem.....	18
2.3.3	Šestnajstiški sistem.....	18
2.3.3.1	Pretvorba iz šestnajstiškega v desetiški sistem	18
2.3.3.2	Pretvorba iz desetiškega v šestnajstiški sistem	18
2.3.3.3	Pretvorba iz dvojiškega v šestnajstiški sistem in obratno	19
2.4	Kodiranje	19
2.4.1	BCD-koda.....	19
2.4.1.1	Seštevanje	20
2.4.2	Kodiranje celih števil	22
2.4.2.1	Kodiranje “predznak, velikost”.....	22
2.4.2.2	Binarno kodiranje s premikom.....	22
2.4.2.3	Kodiranje z dvojiškim komplementom.....	23
2.4.3	Grayev kod	24
2.4.4	Kodiranje znakov	24
2.5	Povzetek.....	25

2.6	Vprašanja in naloge za preverjanje znanja	25
3	SPLET	26
3.1	Protokoli	26
3.1.1	Model ISO/OSI	26
3.1.2	Model TCP/IP	27
3.2	Namen in vsebina spletnih strani.....	27
3.3	Programi za dostop do spleta.....	28
3.3.1	Gumbi.....	28
3.3.2	Naslovna vrstica	29
3.3.3	Zaznamki.....	30
3.4	Spletni obrazci	30
3.5	HTML.....	31
3.5.1	Značke	31
3.5.2	Ogrodje.....	31
3.5.3	Naslovi	32
3.5.4	Odstavki in nove vrstice	32
3.5.5	Stili in oblike izpisa.....	33
3.5.6	Posebni znaki.....	33
3.5.7	Seznami	34
3.5.7.1	Neoštevilčen seznam.....	34
3.5.7.2	Oštevilčeni seznam	34
3.5.7.3	Gnezdenje seznamov	35
3.5.8	Slike.....	35
3.5.9	Tabele	36
3.6	Prekrivni slogi (CSS).....	37
3.6.1	CSS sintaksa in selektorji	40

3.6.1.1	Univerzalni selektor	40
3.6.1.2	Klasični selektor.....	40
3.6.1.3	ID in Class.....	40
3.6.1.4	Stanja selektorjev ter preostale tehnike.....	41
3.7	Povzetek.....	41
3.8	Vprašanja in naloge za preverjanje znanja	42
4	ELEKTRONSKA POŠTA	46
4.1	Sestava naslova elektronske pošte	46
4.2	Pravila pri uporabi elektronske pošte	46
4.3	Uporaba brezplačnega servisa za elektronsko pošto Gmail	47
4.3.1	Sestavljanje in pošiljanje novega sporočila.....	49
4.3.2	Odgovor na sporočilo	49
4.3.3	Posredovanje sporočila drugemu naslovniku	50
4.4	Povzetek.....	51
4.5	Vprašanja in naloge za preverjanje znanja	51
5	DIGITALNI PODPIS.....	52
5.1	Odpiranje podpisanega dokumenta	52
5.2	Varnostna opozorila.....	52
5.3	Povzetek.....	52
5.4	Vprašanja in naloge za preverjanje znanja	53
6	ZAKLJUČEK.....	54

PREDGOVOR

Gradivo Delo s spletom je namenjeno dijakom 1. letnika srednjega strokovnega izobraževanja, smer tehnik računalništva, oziroma vsem začetnikom, ki želijo usvojiti osnovno znanje za delo s spletom. Zgradba in vsebina gradiva omogočata bralcu hitro in enostavno dostopanje do želenih informacij, kar ga še dodatno motivira in s tem nehoti povečuje njegovo učinkovitost pri pridobivanju in iskanju novih informacij.

Vsebina je razdeljena na logično in smiselno zaključena poglavja, iz katerih je jasno razvidna, njihova tematika. Za preverjanje pridobljenega znanja so dodane naloge in vprašanja iz obravnavanega sklopa, ki smiselno sledijo zaključku vsakega vsebinskega dela. Gradivo obravnava vsebine, navedene v katalogu znanja, pri čemer predznanje ni potrebno, kar pa ne pomeni, da ni zaželeno. Vse, kar potrebujete, je želja po znanju, ostalo je v gradivu, ki je pred vami.

Avtorica

1 LOGIČNE OPERACIJE IN BOOLOVA ALGEBRA

Izjava je vsak stavek, ki je bodisi pravilen (resničen) bodisi nepravilen (lažen, neresničen).

Boolova algebra je osnovno matematično orodje, ki ga uporabljamo v analizi in sintezi digitalnih logičnih vezij. Pri Boolovi algebri računamo z logičnimi vrednostmi, in sicer z vrednostma 1 (true oz. resnica) in 0 (false oz. laž). Za osnovno računanje uporabljamo logične operacije:

- **Negacija ali logični ne** (angl. not) izjave x je izjava, ki jo zapišemo $\bar{x}$, x' in pomeni izjavo »ni res, da velja x « ali na kratko »ne x «. Negacija pravilne izjave je vedno nepravilna in obratno.

x	$\bar{x}$
0	1
1	0

Primer: Če je izjava x : *Danes je sobota*, je izjava x' : *Danes ni sobota*.

Primer: Če je $x = 1011\ 1101\ 1111\ 0000$, je $x' = 0100\ 0010\ 0000\ 1111$.

- **Konjunkcija ali logični in** (angl. and) je izjavna povezava, ki dvema izjavama x in y priredi novo izjavo $x \& y$ oz. $x \wedge y$ in pomeni » x in hkrati y « ali » x in y «. Konjunkcija izjav je pravilna natanko takrat, ko sta pravilni obe izjavi.

x	y	$x \& y$
0	0	0
0	1	0
1	0	0
1	1	1

Primer: Izjava *Medved je sesalec in tri je praštevilo* je konjunkcija izjav, ki je pravilna.

Primer: $0011 \wedge 0001 = 0001$

- **Disjunkcija ali logični ali** (angl. or) izjav x in y je izjava, ki jo zapišemo $x \vee y$ in pomeni » x ali y «, kar pomeni, da velja izjava x ali izjava y ali pa obe hkrati. Disjunkcija izjav je pravilna natanko takrat, ko je pravilna vsaj ena od izjav.

x	y	$x \vee y$
0	0	0

0	1	1
1	0	1
1	1	1

Primer: Izjava *Danes je petek ali pa dežuje* je disjunkcija izjav.

Primer: $0001 \vee 0010 = 0011$

- **Antivalenca ali ekskluzivni OR – XOR** izjav x in y je logična operacija $x \oplus y$, ki je pravilna le, kadar je ena od izjav pravilna in ena nepravilna.

x	y	xy
0	0	0
0	1	1
1	0	1
1	1	0

- **Ekvivalenca ali ekskluzivni NOR – XNOR** izjav x in y je logična operacija $x \odot y$, ki je pravilna le, kadar sta obe izjavi pravilni ali obe izjavi nepravilni (oz. je pravilnost prve enaka pravilnosti druge, torej ekvivalentni).

x	y	xy
0	0	1
0	1	0
1	0	0
1	1	1

V Boolovi algebri je vrstni red operacij natančno določen, in sicer tako, da se negacija vedno izvede prva, sledi ji konjunkcija in nato disjunkcija. Z uporabo oklepajev določimo vrstni red operacij v Boolovi algebri. V Boolovi algebri poznamo tudi naslednje trditve:

- $1 \wedge X = X$,
- $0 \wedge X = 0$,
- $1 \vee X = 1$,
- $0 \vee X = X$,
- $X' \vee X = 1$,

- $X' \wedge X = 0$,
- $0' = 1$.

1.1 Povzetek

V tem poglavju smo spoznali pojma logika in Boolova algebra. Predstavljene so logične operacije negacija, konjunkcija in disjunkcija. Na koncu pa so predstavljene še trditve v Boolovi algebri.

1.2 Vprašanja in naloge za preverjanje znanja

1. Zapiši rezultat true (resnica) ali false (laž):
 - a) Simon Jenko je napisal delo Sonetni venec.
 - b) 5 je manjše od 8.
 - c) 1 je sodo število.
 - d) 17 ni liho število.
 - e) Grozdje je zelenjava.
 - f) Ljubljana je glavno mesto Slovenije.
2. Zapiši trditve z Booleovo algebro:
 - a) x je večji od 13 in hkrati manjši od 9.
 - b) y je enak 12 ali večji od 22.
 - c) z je različen od 6 in manjši od 9 ali večji od 13.
 - d) x_1 je različen od x_2 ali x_1 je manjši od 12 in hkrati x_2 je večji od 4.
 - e) y_1 je enak 4 in hkrati je enak 7.
 - f) z_1 je enak 6 ali je enak 9.
3. Zapiši:
 - a) $x = 1001\ 0101\ 1101\ 1000$, $x' = ?$
 - b) $1001 \wedge 1101 = ?$
 - c) $1011\ 0010 \wedge 1110\ 1010 = ?$
 - d) $1101 \vee 1011 = ?$
 - e) $1010\ 1011 \vee 1110\ 0110 = ?$

2 KODIRANJE PODATKOV

S pojmom kodiranje imenujemo vsako pretvorbo podatkov iz ene oblike v drugo, pri čemer najpogosteje kodiramo zaporedje simbolov. Cilj kodiranja je shranjevanje, prenašanje, obdelava podatkov, razumljivost predstavitve in varovanje oziroma skrivanje vsebine (šifriranje). Ločimo dva načina kodiranja:

- **zvezno oziroma analogno kodiranje podatkov**, ko poteka sprememba podatka iz ene na drugo vrednost zvezno,
- **diskretno oziroma digitalno kodiranje podatkov**, ko poteka sprememba podatka iz ene v drugo vrednost korak za korakom – skokovito.

Preden si podrobneje pogledamo kodiranje, razložimo še pojme digitalni, analogni in številski sistemi.

2.1 Digitalni sistemi

Digitalni sistemi po svoji naravi procesirajo oziroma pretvarjajo števila. Če jih torej želimo uporabiti za procesiranje česarkoli drugega (npr. sliko, zvok, besedilo), moramo to najprej pretvoriti v številsko obliko. Procesiranje števil na abstraktnem nivoju se ravna po zakonih oziroma principih matematike in logike, pri konkretni praktični realizaciji takih sistemov pa je seveda potrebno upoštevati tudi fizikalne zakone. Digitalne sisteme je mogoče realizirati na mnogo različnih načinov, s pomočjo električne, elektronske, pnevmatske, hidravlične in drugih tehnologij. Predvsem (mikro) elektronske oblike realizacije digitalnih sistemov so zaradi ekonomičnosti izdelave, dosegle v zadnjih desetletjih izjemen razvoj. Tako se danes uveljavljajo na področjih, ki so bila včasih domena analognih sistemov ali pa odpirajo možnosti za povsem nove aplikacije.

2.2 Analogni sistemi

Analogni sistemi po svoji naravi zvezno procesirajo oziroma pretvarjajo fizikalne količine (npr. električni tok, napetost, tlak, masni in volumenski pretok, temperaturo itd.), njihovo delovanje pa uravnavajo fizikalni zakoni. Analogni sistemi pogosto pretvarjajo eno vrsto

fizikalnih količin (npr. termodinamske) v drugo (npr. električne) in pri tem izkoriščajo fizikalne zakonitosti.

Iz zgoraj opisanega sledi, da je prednost digitalnih sistemov v primerjavi z analognimi sistemi zlasti njihova zanesljivost.

2.3 Številski sistemi

Poznamo več številskih sistemov, pri čemer nam je verjetno najbolj poznan **desetiški sistem**, saj ga najpogosteje uporabljamo v vsakdanjem življenju. Poleg desetiškega sistema pa se v računalništvu pogosto pojavljata tudi **dvojiški** in **šestnajstiški številski sistem**.

2.3.1 Desetiški sistem

Desetiško število (npr. 165,93) je niz števk oziroma simbolov 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ki jih po dogovoru razumemo kot multiplikatorje zaporednih potenc števila 10. Torej število je enako vsoti multiplikatorjev in potenc števila 10

$$165,93_{10} = 1 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 + 9 \times 10^{-1} + 3 \times 10^{-2}.$$

Število 10 imenujemo baza desetiškega številskega sistema.

2.3.2 Dvojiški oz. binarni sistem

Baza dvojiškega oziroma binarnega številskega sistema je število 2, pri čemer poznamo v dvojiškem sistemu le simbola 0 in 1. Podobno kot v desetiškem sistemu, lahko tudi v dvojiškem sistemu vsako število zapišemo kot vsoto multiplikatorjev in potenc števila 2

$$11011_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

Dvojiški sistem je po naravi stvari najnaravnejši za opisovanje sistemov, katerih osnovni gradniki oziroma elementi se lahko v kateremkoli trenutku nahajajo v le enem od dveh stanj, npr. res-ni res, visoko-nizko, gor-dol, odprto-zaprto, vroče-hladno, da-ne itd. Kateremu od obeh stanj bo prirejena katera od dvojiških števil, je odvisno od izbire oziroma presoje načrtovalca. Potrebno je tudi omeniti, da dvojiško številko, torej števili 0 in 1, imenujemo bit, ki predstavlja okrajšavo za angleško besedo binary digit. Četverico bitov pa imenujemo v angleškem jeziku nibble, kar pomeni grižljaj, osmerico bitov imenujemo bajt, večjo n-

terico pa word oziroma beseda. Pri besedi se zapiše, iz koliko bitov je sestavljena (npr. 16-bitna beseda).

2.3.2.1 Pretvorba iz dvojiškega v desetiški sistem

Pretvorba iz dvojiškega v desetiški sistem poteka po naslednjem postopku

$$11011_2 = 1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 25_{10}.$$

2.3.2.2 Pretvorba iz desetiškega v dvojiški sistem

Pretvorba iz desetiškega v dvojiški sistem poteka s pomočjo ostankov pri deljenju s številom 2, pri čemer predstavlja prvi ostanek najnižjo, zadnji pa najvišjo številko.

$$25 / 2 = 12 \quad \text{in ostanek je } 1 \qquad 25 = 2 * 12 + 1$$

$$12 / 2 = 6 \quad \text{in ostanek je } 0 \qquad 12 = 2 * 6 + 0$$

$$6 / 2 = 3 \quad \text{in ostanek je } 0 \qquad 6 = 2 * 3 + 0$$

$$3 / 2 = 1 \quad \text{in ostanek je } 1 \qquad 3 = 2 * 1 + 1$$

$$1 / 2 = 0 \quad \text{in ostanek je } 1 \qquad 1 = 2 * 0 + 1$$

Število $25_{(10)}$ je torej $11001_{(2)}$.

2.3.3 Šestnajstiški sistem

Baza šestnajstiškega številskega sistema je število 16. **Šestnajstiški sistem** pa ima 16 simbolov 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Seveda pa lahko tudi v šestnajstiškem sistemu število zapišemo kot vsoto multiplikatorjev in potenc števila 16

$$3F9_{16} = 3 \times 16^2 + F \times 16^1 + 9 \times 16^0.$$

2.3.3.1 Pretvorba iz šestnajstiškega v desetiški sistem

Pretvorba iz šestnajstiškega v desetiški sistem poteka po naslednjem postopku

$$3F9_{16} = 3 \times 256 + F \times 16 + 9 \times 1 = 1017_{10}.$$

2.3.3.2 Pretvorba iz desetiškega v šestnajstiški sistem

Pretvorba iz desetiškega v šestnajstiški sistem poteka s pomočjo ostankov pri deljenju s številom 16, pri čemer predstavlja prvi ostanek najnižjo, zadnji pa najvišjo številko.

$$1017 / 16 = 63 \quad \text{in ostanek je } 9 \quad 1017 = 16 * 63 + 9$$

$$63 / 16 = 3 \quad \text{in ostanek je } 15 (=F_{(16)}) \quad 63 = 16 * 3 + 15$$

$$3 / 16 = 0 \quad \text{in ostanek je } 3 \quad 3 = 16 * 0 + 3$$

Število $1017_{(10)}$ je torej $3F9_{(16)}$.

Šestnajstiški številski sistem uporabljamo predvsem zaradi dveh lastnosti:

- zapis v šestnajstiškem sistemu je v primerjavi z dvojiškim precej krajši
- ker se da eno šestnajstiško števko zapisati z natanko štirimi dvojiškimi.

2.3.3.3 Pretvorba iz dvojiškega v šestnajstiški sistem in obratno

Za pretvorbo dvojiškega v šestnajstiški sistem in obratno moramo poznati dvojiške zapise števk šestnajstiškega sistema, in sicer vsaki šestnajstiški števk ustreza štirimestno dvojiško število.

$$0 \leftrightarrow 0000 \quad 1 \leftrightarrow 0001 \quad 2 \leftrightarrow 0010 \quad 3 \leftrightarrow 0011$$

$$4 \leftrightarrow 0100 \quad 5 \leftrightarrow 0101 \quad 6 \leftrightarrow 0110 \quad 7 \leftrightarrow 0111$$

$$8 \leftrightarrow 1000 \quad 9 \leftrightarrow 1001 \quad A \leftrightarrow 1010 \quad B \leftrightarrow 1011$$

$$C \leftrightarrow 1100 \quad D \leftrightarrow 1101 \quad E \leftrightarrow 1110 \quad F \leftrightarrow 1111$$

Sedaj lahko vsaki števk šestnajstiškega števila priredimo ustrezno štirimestno dvojiško število iz tabele:

$$3F9_{(16)} = 0011 \ 1111 \ 1001_{(2)}$$

$$101 \ 1000 \ 0100 \ 0001_{(2)} = 5841_{(16)}$$

2.4 Kodiranje

2.4.1 BCD-koda

Desetiško število lahko kodiramo tudi tako, da vsaki številki desetiškega števila priredimo ustrezno dvojiško četverico.

$$0 \leftrightarrow 0000 \quad 1 \leftrightarrow 0001 \quad 2 \leftrightarrow 0010 \quad 3 \leftrightarrow 0011$$

$$4 \leftrightarrow 0100 \quad 5 \leftrightarrow 0101 \quad 6 \leftrightarrow 0110 \quad 7 \leftrightarrow 0111$$

$$8 \leftrightarrow 1000 \quad 9 \leftrightarrow 1001$$

Za primer vzemimo desetiško število 529.

$$529_{(10)} = 0101\ 0010\ 1001_{(BCD)}$$

$$529_{(10)} = 0 \times 800 + 1 \times 400 + 0 \times 200 + 1 \times 100 + 0 \times 80 + 0 \times 40 + 1 \times 20 + 0 \times 10 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$$

BCD-oblika dvojiške predstavitev decimalnega števila ni enaka neposrednemu zapisu v dvojiški obliki:

$$529_{(10)} = 211_{(16)} = 0010\ 0001\ 0001_{(2)}$$

$$529_{(10)} = 0 \times 2048 + 0 \times 1024 + 1 \times 512 + 0 \times 256 + 0 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1$$

Kot je razvidno iz zgornjega primera, potrebuje BCD-koda za zapis iste informacije več bitov kot binarna koda, saj nikoli ne uporabi števil od 10 do 15, ki jih lahko prestavlja 4-bitna četverica. BCD-koda se precej uporablja, zlasti za vnašanje in prikazovanje numeričnih podatkov (na kalkulatorjih oz. žepnih računalih). Obstajajo električna vezja, ki pretvarjajo BCD-kodo v binarno oziroma obratno. Nekateri mikroprocesorji imajo poleg ukazov za binarno aritmetiko tudi ukaze za BCD-aritmetiko. Obstajajo pa tudi kode z drugačnimi utežmi, ki pa se danes redkeje uporabljajo.

2.4.1.1 Seštevanje

Seštevamo po pravilih za binarno seštevanje.

$$\begin{array}{rcl} 12_{(10)} & = & 0001\ 0010_{(BCD)} \\ + 34_{(10)} & = & + 0011\ 0100_{(BCD)} \\ \hline 46_{(10)} & = & 0100\ 0110_{(BCD)} \end{array}$$

Rezultat pa je potrebno popraviti s prištevanjem četverice $0110_{(BCD)} = 6_{(10)}$:

- če pride pri seštevanju do prenosa bita iz ene četverice na drugo ali

$$\begin{array}{rcl}
 19_{(10)} & = & 0001\ 1001_{(BCD)} \\
 + 48_{(10)} & = & + 0100\ 1000_{(BCD)} \\
 \hline
 & & 0110\ 0001_{(BCD)} \\
 & & + 0000\ 0110_{(BCD)} \\
 & & \hline
 67_{(10)} & = & 0110\ 0111_{(BCD)}
 \end{array}$$

- če je rezultat v četvorki večji od $1001_{(BCD)} = 9_{(10)}$.

$$\begin{array}{rcl}
 57_{(10)} & = & 0101\ 0111_{(BCD)} \\
 + 38_{(10)} & = & + 0011\ 1000_{(BCD)} \\
 \hline
 & & 1000\ 1111_{(BCD)} \\
 & & + 0000\ 0110_{(BCD)} \\
 & & \hline
 95_{(10)} & = & 1001\ 0101_{(BCD)}
 \end{array}$$

$$\begin{array}{rcl}
 88_{(10)} & = & 1000\ 1000_{(BCD)} \\
 + 99_{(10)} & = & + 1001\ 1001_{(BCD)} \\
 \hline
 & & 1\ 0010\ 0001_{(BCD)} \\
 & & + 0110\ 0110_{(BCD)} \\
 & & \hline
 187_{(10)} & = & 1\ 1000\ 0111_{(BCD)}
 \end{array}$$

$$\begin{array}{rcl}
 56_{(10)} & = & 0101\ 0110_{(BCD)} \\
 + 78_{(10)} & = & + 0111\ 1000_{(BCD)} \\
 \hline
 & & 1100\ 1110_{(BCD)} \\
 & & + 0110\ 0110_{(BCD)} \\
 & & \hline
 134_{(10)} & = & 1\ 0011\ 0100_{(BCD)}
 \end{array}$$

2.4.2 Kodiranje celih števil

Poznamo tri vrste kodiranja celih števil:

- **kodiranje “predznak, velikost”,**
- **binarno kodiranje s premikom** in
- **kodiranje z dvojiškim komplementom.**

2.4.2.1 Kodiranje “predznak, velikost”

Kodiranje “predznak, velikost” poteka tako, da en bit (recimo skrajno levi) namenimo za zapis informacije o predznaku števila, ki mu sledi

$$\begin{array}{llll}
 +7 \leftrightarrow 0111 & +6 \leftrightarrow 0110 & +5 \leftrightarrow 0101 & +4 \leftrightarrow 0100 \\
 +3 \leftrightarrow 0011 & +2 \leftrightarrow 0010 & +1 \leftrightarrow 0001 & +0 \leftrightarrow 0000 \\
 -0 \leftrightarrow 0000 & -1 \leftrightarrow 0001 & -2 \leftrightarrow 0010 & -3 \leftrightarrow 0011 \\
 -4 \leftrightarrow 0100 & -5 \leftrightarrow 0101 & -6 \leftrightarrow 0110 & -7 \leftrightarrow 0111
 \end{array}$$

Problem, ki se pojavi pri takšnem kodiranju, sta predvsem dve ničli (pozitivna in negativna) in dejstvo, da je seštevanje drugačno od odštevanja, zato se to kodiranje redkeje uporablja.

2.4.2.2 Binarno kodiranje s premikom

Binarno kodiranje s premikom poteka tako, da množico celih števil s prištevanjem konstante preslikamo v ustrezno množico nenegativnih celih števil, ki jih nato binarno kodiramo. Za lažje razumevanje takšnega kodiranja pogledjmo naslednji 4-bitni primer.

$$\{-8, -7, \dots, +7\} + 8 = \{0, 1, \dots, 15\} = \{0000, 0001, \dots, 1111\}$$

Značilnosti te vrste kodiranja so:

- levi bit predstavlja informacijo o predznaku,
- binarna števila predstavljajo aritmetično zaporedje (ugodno za binarno štetje),
- ničla se pojavlja le enkrat,
- pogosto se uporablja pri analogno-digitalnih in digitalno-analognih pretvorbah,
- je neugodno za aritmetične operacije.

2.4.2.3 Kodiranje z dvojiškim komplementom

Če je decimalno število nenegativno, je njegov **dvojiški komplement** enak njegovi binarni kodi. Če je decimalno število negativno, dobimo njegov dvojiški komplement tako, da najprej vzamemo njegovo velikost, jo pretvorimo v binarni ekvivalent, ki ga nato komplementiramo, to je spremenimo 1 v 0 ali obratno, da dobimo eniški komplement, in temu prištejemo še 1.

$$+7 \Rightarrow 0111$$

$$-7 \Rightarrow +7 \Rightarrow 0111 \Rightarrow 1000 \Rightarrow 1001$$

Značilnosti te vrste kodiranja so:

- levi (oz. najpomembnejši) bit predstavlja informacijo o predznaku,
- ena ničla, ki je enaka 0000, kar je ugodno z vidika realizacije,
- če jo primerjamo z binarno kodo s premikom, je njen bit s predznakom komplementiran,
- aritmetične operacije nad celimi števili so zelo enostavne, zato se množično uporablja v procesorjih.
- posebno vezje za odštevanje ni potrebno.

Seštevanje poteka po naslednjem principu:

$$+7 + (-4) = +3$$

$$0111 + 1100 = 0011$$

Odštevanje pretvorimo v seštevanje z dvojiškim komplementom in poteka po naslednjem principu:

$$+4 - (+5) = +4 + (-5) = -1$$

$$0100 - 0101 = 0100 + 1011 = 1111$$

Množenje poteka po naslednjem principu:

$$-3 * (+2) = -6$$

$$1101 * 0010 = 11010 = 1010$$

2.4.3 Grayev kod

Grayev kod se uporablja v merilnih napravah, kot so merilne letve, rotacijski enkoderji, itd. Grayevi kodi so lahko ciklični (enkoder) ali pa ne (letev).

2.4.4 Kodiranje znakov

Poleg števil lahko v binarno obliko kodiramo tudi znake. V računalništvu se za kodiranje znakov uporablja **standard ASCII** (angl. American Standard Code for Information Interchange), ki vsebuje 256 znakov. Za njihovo kodiranje je potrebno 8 bitov oziroma 1 zlog (bajt). Standard ASCII so razvili v šestdesetih letih v ZDA programerski strokovnjaki za najpogostejše uporabljene znake, s čimer so omogočili izmenjavo informacij med poljubnimi računalniki.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Slika 1: Tabela ASCII

Pomembno je opozoriti, da je standard ASCII uporaben samo v angleško govorečih državah in da v njem ni mogoče najti šumnikov. Te in ostale znake neangleške abecede je zato potrebno dodajati s pomočjo dodatnih kodnih standardov, ki jih imenujemo kodne tabele. Kodne tabele v zadnjem času izpodrivata standarda UCS in UTF.

2.5 Povzetek

V tem poglavju smo spoznali pojem kodiranje in se podrobneje seznanili z binarnim kodiranjem. Spoznali smo načine kodiranja znakov in standard ASCII.

2.6 Vprašanja in naloge za preverjanje znanja

2. Kaj je kodiranje?
3. Kateri vrsti kodiranja poznate?
4. Kaj je digitalni sistem?
5. Kaj je analogni sistem?
6. Kaj je številski sistem in katere številske sisteme poznate?
7. Opišite kako poteka pretvarjanje med različnimi številskimi sistemi?
8. Kaj pomeni, da podatke binarno kodiramo?
9. Kaj je BCD-koda in kako poteka takšno kodiranje?
10. Kako kodiramo cela števila?
11. Kako kodiramo znake?
12. Kaj je standard ASCII?

3 SPLET

Splet se morda zdi na prvi pogled zastrašujoč, saj vsebuje veliko število globalnih komunikacijskih omrežij z milijardami spletnih strani. Toda v tem poglavju bomo razložili osnove o spletu na poenostavljen in ne tehnični način. Internet, cyberspace in net so izrazi, ki se v splošnem navezujejo na izraz splet.



Slika 2: Splet

Vir: <http://www.petrostrategies.org> (14. 8. 2015)

Splet je omrežje računalnikov, ki so povezani med seboj in uporabljajo omrežni protokol TCP/IP. Namenjen je pridobivanju informacij, komunikaciji in prenosu podatkov. Je koristen vir informacij, ki sproti beleži vse novosti z vseh področij življenja.

3.1 Protokoli

Računalniki pri medsebojnem komuniciranju uporabljajo računalniški jezik, ki je strukturiran v obliki dogovorjenih pravil in postopkov, t. j. **protokolov**. Računalnika, ki uporabljata različne protokole, potrebuje posrednika, ki zna protokole ustrezno pretvoriti. Najbolj pogosta omrežna protokola sta **model ISO/OSI** (ang. International Standard Organization/Open System Interconnections) in **model TCP/IP** (angl. Transmission Control Protocol/Internet Protocol).

3.1.1 Model ISO/OSI

Komunikacija med sistemi poteka od aplikacijske plasti proti fizični plasti in obratno in jo imenujemo **vertikalna komunikacija**. Vsaka plast lahko razume le informacije, ki jih prejme od istoimenske plasti računalnika, s katerim komunicira. To imenujemo **horizontalna komunikacija**. Istoimenske plasti se sporazumevajo s pomočjo kontrolne

informacije, ki je dodana podatkom v obliki zglavja (angl. header) oziroma repa (angl. trailer). V oddajnem sistemu vsaka plast doda svoje zglavje podatkom, ki jih je prejela od višje plasti (enkapsulacija), v sprejemnem sistemu pa vsaka plast odvzame kontrolno informacijo. Model OSI najpogosteje omenja tri vrste podatkovnih enot, in sicer okvir, paket in sporočilo. **Sporočilo** je blok podatkov, ki si jih med seboj pošiljajo informacijske storitve. **Paketi** pa so sporočila, predelana v obliko, primerno za prenos skozi omrežje. Običajno je sporočilo sestavljeno iz večjega števila paketov. Paket povezovalne plasti je edini izmed paketov, ki poleg glave vsebuje tudi rep, zato paket na drugi plasti imenujemo tudi **okvir**.

3.1.2 Model TCP/IP

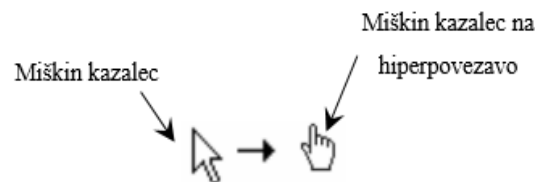
Model TCP/IP spada med t. i. de facto standarde. Razdeljen je na štiri plasti, kjer vsaka plast predstavlja svoj nabor funkcij in storitev. Višja protokolna plast predstavlja odjemalca, ki od nižje ležeče plasti zahteva neko storitev. Nižje ležeča plast tako služi kot strežnik višje ležeči plasti. Plasti modela TCP/IP so:

- **Aplikacijska plast**, ki je najvišje ležeča plast in zagotavlja povezavo uporabnika z omrežjem. Pomembnejše storitve aplikacijske plasti so danes predvsem internetne storitve, kot npr. Telnet, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), HTTP, SNMP, itd.
- **Transportna plast** je podobna transportni plasti modela OSI in zagotavlja zanesljiv prenos podatkov, definira zanesljiv protokol TCP, ki je povezano orientirana storitev, in hitrejši ter manj zanesljiv protokol UDP, ki je nepovezano orientiran.
- **Omrežna plast** skrbi za usmerjanje paketov od oddajnika do prejemnika. Definira protokol IP, danes najbolj razširjen protokol za usmerjanje.
- **Dostopovna plast** definira način prenosa podatkov med končnimi uporabniki in omrežjem, na katerega so ti priključeni.

3.2 Namen in vsebina spletnih strani

Na spletu obstaja veliko spletnih strani, ki so izdelane za določen namen, od posredovanja novic po svetu do poučevanja o različnih tematikah. Mogoče jih je primerjati s časopisi in knjigami, saj jih tudi sestavljajo posamezne strani. Spletne strani so praviloma med seboj neodvisne, včasih pa so med seboj tudi povezane s **hiperpovezavami** (imenujemo jih tudi

spletne povezave), ki nam omogočajo preskoke z ene spletne strani na drugo. Te povezave so ponavadi **podčrtane** in **obarvane modro**, lahko pa so zapisane tudi s katerokoli drugo barvo ali namesto besedila vsebujejo sliko. Ko se s kazalcem postavimo na hiperpovezavo, se puščica spremeni v **kazalno roko**.



Slika 3: Miškin kazalec na hiperpovezavi

Spletne strani so tisto, kar vidimo in preberemo na spletu, in so sestavljene predvsem iz besedila (besed), digitalnih medijev (slik, filmov in glasbe) in hiperpovezav.

3.3 Programi za dostop do spleta

Za brskanje po spletu uporabljamo programe imenovane **spletni brskalniki**. Najpogostejši so Internet Explorer, Firefox, Google Chrome, Safari in Opera.



Slika 4: Najpogostejše uporabljeni spletni brskalniki

Vsi služijo enakemu namenu, in sicer brskanju po spletu, in vsebujejo veliko podobnih gumbov.

3.3.1 Gumbi

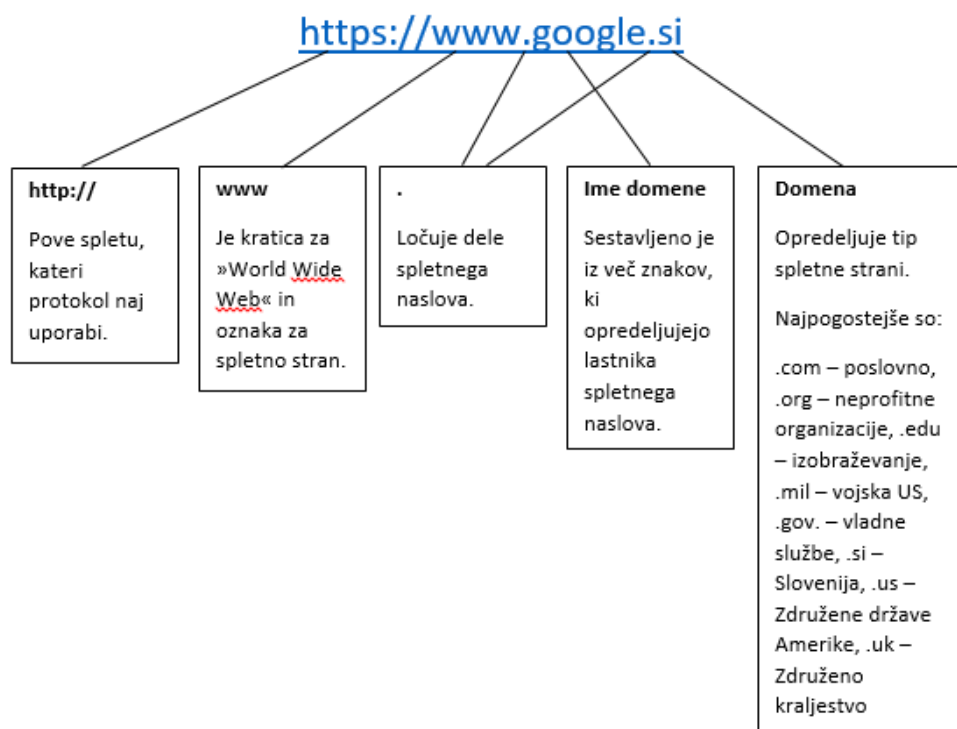
V vsakem spletnem brskalniku najdemo naslednje gumbe:

- **Gumb Pojdi nazaj** (angl. Back Button) omogoča, da se vrnemo na spletno stran, ki smo jo obiskali nazadnje; pogosto ga uporabljamo, če smo pomotoma kliknili na povezavo in se želimo vrniti na prejšnjo stran.

- Če kliknemo na gumb *Pojdi nazaj* in se želimo nato ponovno vrniti na stran, kjer smo bili preden smo kliknili na gumb *Pojdi nazaj*, kliknemo na **gumb Pojdi naprej** (angl. Forward Button).
- **Gumb Osveži stran** (angl. Refresh Button) je zelo uporaben na spletnih straneh, ki se pogosto posodablja, kot na primer novice, šport, vreme itd. S klikom na gumb *Osveži stran* se stran ponovno naloži.
- Ko odpremo svoj spletni brskalnik, je prva prikazana spletna stran naša domača stran. Svojo domačo stran lahko kadarkoli spremenimo in nastavimo na drugo spletno stran. Klik na **gumb Odpri domačo stran** (angl. Home Button) nas popelje na našo domačo stran.

3.3.2 Naslovna vrstica

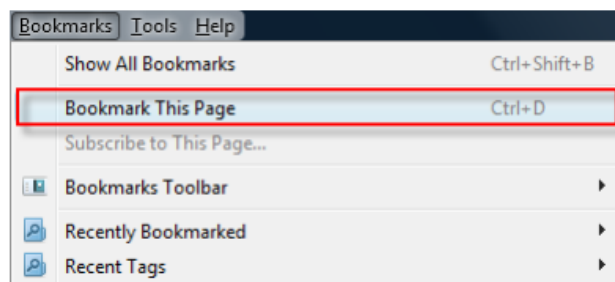
Naslovna vrstica prikazuje **URL-spletne strani**. URL je kratica za *Universal Resource Locator*, ki označuje naslov vsake posamezne spletne strani. Dve spletni strani ne moreta imeti istega URL-naslova. V naslovno vrstico lahko vnesemo poljuben URL, in sicer z levim klikom v naslovno vrstico. URL-ji se med seboj razlikujejo, kljub temu pa imajo skupne značilnosti.



Slika 5: URL

3.3.3 Zaznamki

Pri brskanju po spletu lahko naletimo na spletno stran za katero vemo, da jo bomo v prihodnosti pogosto potrebovali. V takšnem primeru nam spletni brskalniki ponujajo možnost shranitve določene spletne strani med **zaznamke**. Spletno stran shranimo med zaznamke tako, da kliknemo na *gumb Zaznamki* (angl. Bookmarks) in izberemo *Dodaj to stran med zaznamke* (angl. Bookmark This Page).



Slika 6: Zaznamki

3.4 Spletni obrazci

Spletni obrazci so odlično orodje za zbiranje informacij o določeni zadevi. Ponavadi jih uporabljamo za prijavo, anketiranje, vzpostavitev kontakta itd. Podatki, ki jih uporabnik vnese v polja spletnega obrazca, se prenesejo v bazo, ki je povezana s spletnim obrazcem, lahko pa se posredujejo tudi na elektronski naslov.

Se želite o svoji bodoči spletni strani pogovoriti po telefonu? Ni problema! Vpišite svoje podatke in pokličali vas bomo!

Ime in priimek:*

Podjetje:

Telefonska številka:*

Želen čas klica:

Kaj vas zanima? ?

* - obvezno polje

Zahtevam klic

Slika 7: Spletni obrazec

3.5 HTML

HTML (angl. Hyper Text Markup Language) je jezik za izdelavo spletnih strani in predstavlja osnovo spletnega dokumenta. **Spletni pregledovalniki** (angl. Browsers) so programi, ki berejo ukaze v jeziku HTML in prikazujejo ustrezno vsebino.

3.5.1 Značke

HTML-dokument lahko izdelamo v editorju (npr. Beležnici), kjer zapišemo HTML-elemente, sestavljene iz **značk** (angl. Tags), zapisanih v **lomljenih oklepajih** (npr. <TAG>). Poznamo **samostojne značke**, ki ne potrebujejo zaključka (npr. nova vrstica ali nov odstavek), in **začetne ter končne značke**, ki jih uporabljamo, kadar želimo določiti, do katerega mesta seže učinek značke (npr. krepko in poravnava). Končna značka ima pred imenom znak /.

3.5.2 Ogrodje

Opis strani oziroma ogrodje v jeziku HTML je naslednji:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> Naslov strani </title>
```

```
</head>
```

```
<body>
```

Vsebina strani

```
</body>
```

```
</html>
```

Med znački **<HEAD>** in **</HEAD>** napišemo informacije o dokumentu, ki pa se na strani ne izpišejo. V zgoraj predstavljenem primeru smo uporabili značko **<TITLE>**, ki podaja vsebino, ki se izpiše v zgornjem okviru okna. Za oblikovanje strani v slovenščini in za

pisanje šumnikov si pomagamo z uporabo elementa **META** (element s prazno vsebino, zato ne pišemo zaključne značke):

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">.
```

Torej, če uporabimo v glavi to značko, lahko uporabljamo šumnike č, š in ž kar s tipkovnice. Še bolj splošna rešitev za pisanje šumnikov pa je uporaba kode znakov v obliki &#n, kjer je n

n	Črka
268	Č
269	č
352	Š
353	š
381	Ž
382	ž

3.5.3 Naslovi

Na voljo imamo več nivojev naslovov, za katere uporabljamo elemente **H1** (glavni naslov), H2 (podnaslov), H3, H4, H5 in H6:

```
<h1> Naslov na 1. nivoju </h1>
```

```
<h2> Naslov na 2. nivoju </h2>
```

```
<h3> Naslov na 3. nivoju </h3>
```

```
<h4> Naslov na 4. nivoju </h4>
```

```
<h5> Naslov na 5. nivoju </h5>
```

```
<h6> Naslov na 6. nivoju </h6>
```

Pod in nad vsakim naslovom pusti brskalnik nekaj prostora in jih piše krepko, v velikostji pisave, ki je odvisna od nivoja naslova.

3.5.4 Odstavki in nove vrstice

Poznamo več vrst odstavkov, osnovni element za odstavke pa je **DIV**, ki poskrbi, da se besedilo prične na začetku nove vrstice, drugih oblikovalnih učinkov pa nima. Z njimi

ponavadi združujemo bloke HTML-značk, saj na ta način kasneje lažje oblikujemo celoten blok. Pogosto uporabljamo pri tvorjenju odstavkov tudi element **P**, ki pusti še nekaj prostora pred in nad odstavkom.

`<p>` To je odstavek. `</p>`

Za prelom vrstice na točno določenem mestu uporabimo element **BR**.

3.5.5 Stili in oblike izpisa

Na voljo imamo več različnih stilov in oblik izpisov:

- za izpis krepko uporabimo element **B**,
- za izpis ležeče uporabimo element **I**,
- za podčrtan izpis uporabimo element **U**,
- za izpis v obliki izpisa s pisalnim strojem uporabimo element **TT**,
- za povečan izpis uporabimo element **BIG** in
- za pomanjšan izpis uporabimo element **SMALL**.

Stile lahko tudi gnezdimo, kar pomeni, da lahko določeno besedilo oblikujemo z več zgoraj opisanimi elementi.

`<b>` To je krepko besedilo. `</b>`

`<i>` To je ležeče besedilo. `</i>`

`<u>` To je podčrtano besedilo. `</u>`

`<tt>` To je izpis besedila v obliki izpisa s pisalnim strojem. `</tt>`

`<big>` To je veliko besedilo. `</big>`

`<small>` To je majhno besedilo. `</small>`

3.5.6 Posebni znaki

Nekateri znaki v HTML imajo poseben pomen. Takšni znaki so `<`, `>`, `"` in `&`, ki jih v besedilu ne moremo uporabljati kot ostale znake, pač pa moramo uporabiti njihova imena, določena s standardom SGML:

Opis	Znak
<	<
>	>
"	"
&	&

3.5.7 Seznami

Sezname v HTML uporabljamo, kadar želimo po vrsti naštetih stvari. Poznamo neoštevilčene in oštevilčene sezname.

3.5.7.1 Neoštevilčen seznam

Za naštevane uporabimo element UL (angl. Unordered list), znotraj katerega naštevamo elemente LI (angl. List items), ki predstavljajo posamezne alineje seznama.

```
<ul>
```

```
<li> Prva alineja neoštevilčenega seznama </li>
```

```
<li> Druga alineja neoštevilčenega seznama </li>
```

```
<li> Tretja alineja neoštevilčenega seznama </li>
```

```
<li> Četrta alineja neoštevilčenega seznama </li>
```

```
</ul>
```

3.5.7.2 Oštevilčeni seznam

Za številčenje uporabljamo element OL (angl. Ordered list), znotraj katerega naštevamo z elementom LI, ki tako kot pri neoštevilčenem seznamu predstavlja posamezne alineje seznama.

```
<ol>
```

```
<li> Prva alineja oštevilčenega seznama </li>
```

```
<li> Druga alineja oštevilčenega seznama </li>
```

```
<li> Tretja alineja oštevilčenega seznama </li>
```

```
<li> Četrta alineja oštevilčenega seznama </li>
```


3.5.7.3 Gnezdenje seznamov

Zgoraj opisani vrsti seznamov lahko tudi gnezdimo, kar pomeni, da lahko naredimo seznam znotraj drugega seznama, pri čemer moramo paziti, da značke zaključujemo v obratnem vrstnem redu, kot smo jih odpirali.

 Prva točka oštevilčenega seznama

 Prva alineja označenega seznama prve točke oštevilčenega seznama

 Druga alineja označenega seznama prve točke oštevilčenega seznama

 Tretja alineja označenega seznama prve točke oštevilčenega seznama

 Četrta alineja označenega seznama prve točke oštevilčenega seznama

 Druga točka

 Tretja točka

3.5.8 Slike

Slike lahko vključimo v dokument s pomočjo elementa IMG, ki ima nekaj pomembnih lastnosti:

- **src:** S to lastnostjo določimo naslov slike oziroma ime datoteke. Če se slika nahaja v isti mapi kot spletna stran, je za vrednost lastnosti dovolj napisati samo ime datoteke. Slika mora biti zapisana na datoteki oblike GIF, JPG ali PNG.
- **width in height:** S tema lastnostma določimo velikost pravokotnega področja, namenjenega prikazu slike. Če velikosti ne predpišemo, bo slika prikazana v njeni originalni velikosti. Če podamo samo širino ali samo višino slike, bo slika tako široka oziroma visoka, kot smo določili, pri čemer pa se bo razmerje med njeno širino in višino

ohranilo. Če pa podamo tako širino kot tudi višino, bo prikazana slika točno takšnih dimenzij kot smo določili, pri čemer moramo biti pozorni na to, da se lahko slika popači.

- alt: S to lastnostjo na kratko opišemo kaj se nahaja na sliki. Ta opis slike se prikaže namesto slike, v primeru da pregledovalnik slike ne more prikazati.

```

```

```

```

```

```

```

```

3.5.9 Tabele

Tabele v HTML omogočajo, da v obliki vrstic in stolpcev razporedimo elemente, kot so na primer besedila, slike, sezname in celo druge tabele. Tabelo opišemo z elementom TABLE, njegova vsebina pa se začne z morebitnim opisom, ki se prikaže nad tabelo sredinsko poravnan čez širino celotne tabele, in sicer s pomočjo elementa CAPTION. Za tem lahko z elementi COL (angl. Column) in COLGROUP (angl. Column group) določimo lastnosti stolpcev ter na koncu določimo vsebimo posamezne celice. Za opis navadne celice uporabimo element TD (angl. Table data), za opis naslovnih celic pa običajno uporabljamo element TH (angl. Table heading). Celice združujemo v vrstice, ki jih opišemo z elementom TR (angl. Table row).

```
<table>
```

```
  <tr>
```

```
    <td> Prva vrstica, prvi stolpec </td>
```

```
    <td> Prva vrstica, drugi stolpec </td>
```

```
    <td> Prva vrstica, tretji stolpec </td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td> Druga vrstica, prvi stolpec </td>
```

```
    <td> Druga vrstica, drugi stolpec </td>
```

```
    <td> Druga vrstica, tretji stolpec </td>
```

</tr>

</table>

3.6 Prekrivni slogi (CSS)

Cascading Style Sheets ali CSS je značilnost HTML-ja, razvita s strani W3C. S CSS je mogoče ustvariti predloge za stile, ki določajo izgled različnih elementov teksta, odstavkov, naslovov, pisave, hiperpovezav, itd. na spletni strani.

CSS najpogosteje zasledimo v spletnih straneh napisanih v spletnem jeziku HTML ali XHTML, lahko pa se uporablja tudi v poljubnem XML dokumentu.

Z uporabo CSS-ja lahko ločujemo vsebinske podatke dokumenta (besedilo, slike, tabele, drugi elementi) od podatkov, ki določajo oblikovne lastnosti dokumenta in njegovih elementov. S pomočjo CSS lahko HTML elementom določimo razne vrste oblikovnih lastnosti (ozadje, barvo, pisavo, odmike, obrobo, pozicijo, velikost, stopnjo prosojnosti, itd.)

Vsebinsko ločujemo od oblike z namenom, da:

- zmanjšamo kompleksnost strani,
- zagotovimo fleksibilnost in nadzor nad lastnostmi,
- je moč uporabljati iste oblikovne lastnosti na več straneh, in sicer brez da bi bilo potrebno te lastnosti posebej določati vsaki strani,
- zagotovimo prikaz strani v različnih oblikah (prikaz na celem zaslonu ali prikaz v oknu).

Obstajajo trije načini, kako vpeljati CSS določila v HTML dokument:

- Prvi in najpogostejši način je dodajanje posebne CSS datoteke, ki jo nato povežemo z datoteko HTML. CSS dodamo tako, da naredimo novo datoteko z končnico .css. To datoteko moramo nasloviti tudi v HTML, in sicer to storimo ponavadi glavi z značko Link.

```
<link rel= "stylesheet" type="text/css" href=css/primer.css">
```

Pri čemer *rel* specificira razmerje med dano datoteko in povezanim dokumentom, *type* določi v kakšnem tiou je povezan dokument in *href* specificira pot do povezanega dokumenta.

Če torej želimo uporabiti zunanjo CSS datoteko, moramo v HTML dokumentu uporabiti ukaz s katerim uvozimo datoteko.

```
<link rel="stylesheet"
      href="http://www.primer.si/css/style.css"
      type="text/css" />
```

CSS datoteka style.css:

```
body{
    background-color: green;
}

h1{
    color: blue;
    text-align: justify;
    font-size:18px;
}

p{
    font-family:"Calibri";
    font-size:12px;
}

.nas-rdeci-tekst{
    color: red;
    font-size: 16px;
    margin: 12px;
}
```

V zgornjem primeru je moč opaziti, da je stil določen za tri osnovne HTML elemente in za en CSS razred, ki smo ga morali dodati in ga lahko priredimo HTML elementom tako, da ob njegovi oznaki dodamo `class="nas-modri-tekst"`

Poglejmo še primer HTML datoteke (npr. stran.html), ki bo predstavljala spletno stran, ki uporablja CSS datoteko style.css.

```
<html>
```

```

<head>

<link rel="stylesheet"

    href=http://www.primer.si/css/style.css

    type="text/css" />

</head>

<body>

<h1>Naslov strani</h1>

<p>Odstavek</p>

<p class="nas-rdeci-tekst">

</p>

</body>

</html>

```

- Drugi način je, da v HTML dokumentu, ponavadi v glavi, posameznemu elementu pripišemo CSS razred. To storimo podobno kot v prvem primeru, le da ne uporabimo dodatne datoteke.

```

<head>

<style type="text/css">

h1{

color:blue;

}

</head>

```

- Tretji način pa je vključitev style atributov v HTML elemente. V HTML dokumentu posameznemu elementu v ukazu style določimo, kakšne lastnosti naj ima.

```

<h1 style="color: blue; text-align: center">

```

Posamezen HTML dokument lahko uporablja eno ali več zunanjih stilskih datotek. V primeru, da so v njih za nek dokument na strani določena različna pravila, pa velja pravilo, ki je nazadnje določeno, ali pa pravilo, ob katerem piše !important.

3.6.1 CSS sintaksa in selektorji

Če želomo definirati selektor v CSS dajemo ukaz HTML datoteki naj doda dana pravila k temu selektorju. Pri tem poznamo več načinov, kako definirati selektorje ter kako s selektorji naslavljamo želene elemente v HTML.

3.6.1.1 Univerzalni selektor

Univerzalni selektor, ki ga zapišemo z *, izbere vse elemente najdene v HTML datoteki, pri čemer njegova pravila veljajo na vseh HTML elementih, zato ga ponavadi ne uporabljamo.

3.6.1.2 Klasični selektor

Klasični selektor izbere vse istoimenske elemente v HTML datoteki.

```
p{  
    color: red;  
}
```

3.6.1.3 ID in Class

ID je lastnost elementa. Z njim določimo ime, s katerim ga potem lahko oblikujemo v CSS z zaznamkom, ki se začne z znakom # pred njegovim imenom. V HTML datoteki sme biti z ID imenom poimenovan samo en element.

```
<p id="primer1">
```

Oblikovanje v CSS:

```
#primer1 {  
    color: red;  
}
```

Class je podoben identifikatorju, vendar za razliko od ID lahko z class-om ciljamo na skupine elementov, ki naj bi imeli enak izgled in funkcionalnost, tako nismo omejeni na točno določen element. Uporabljanje Class tehnike naslavljanja elementov je najpogostejša. V CSS ga naslavljamo tako, da dodamo piko pred imenom classa, v html pa podobno kot ID vendar da namesto ID pišemo Class.

```
<p class="primer2">
```

Oblikovanje v CSS:

```
.primer2 {  
    color: red;  
}
```

3.6.1.4 Stanja selektorjev ter preostale tehnike

Do elementov, ki se lahko v HTML datoteki nahajajo v več stanjih, dostopamo prek selektorjev. Poglejmo si tri stanja, ki so strukturirana za povezavne elemente (a) , vendar jih lahko uporabimo tudi na nekatere druge elemente.

Stanja nastopijo ob interakciji z miško, ko pritisnemo na določen element, ko ga prekrijemo z miško ter ko je bil ta element že obiskan oziroma kliknjen. Do teh stanj dostopamo v CSS z znakom za dvopičje ter z besedami *visited*, *hover*, *active*.

```
a:hover{  
    color: red;  
}  
  
a:active{  
    color: blue;  
}  
  
a:visited{  
    color: yellow;  
}
```

3.7 Povzetek

V tem poglavju bomo spoznali osnove spleta, torej kaj splet sploh je, kaj je namen in vsebina spletnih strani ter opisali protokole. Predstavili in opisali bomo tudi delo s programi za dostop do spleta ter spoznali spletne obrazce in predstavili oblikovanje spletnih strani v jeziku HTML ter prekrivnih slogov oziroma CSS. Seznanili se bomo s sintakso in z vsemi tremi načini uporabe CSS. Osvojili bomo povezavo CSS s HTML kodo in spoznali prednosti ločitve vsebine od oblike spletne strani.

3.8 Vprašanja in naloge za preverjanje znanja

1. Kaj je splet?
2. Kaj je hiperpovezava?
3. Katere spletne brskalnike poznate?
4. Kako je sestavljen URL?
5. Naštejte protokole in jih opišite.
6. Kaj so zaznamki?
7. Kaj je spletni obrazec?
8. Poiščite in prekopirajte na prazen list v Wordu ter shranite:
 - vsa prosta delovna mesta v občini Maribor,
 - telefonsko številko Zdravstvenega doma Maribor,
 - fotografijo Koče na Arehu,
 - fotografijo Maribora.
9. Na namizju svojega računalnika ustvarite mapo Živali. Na spletu poiščite vozni red vlaka iz Maribora v Celje. Ugotovite, prvi možni odhod vlak in ga prekopirajte v Word ter shranite v mapo Živali pod imenom Vozni red. Na spletu poiščite fotografijo koale in delfina (vsako posebej). Shranite slike v mapo Živali. V Slikarju delfinu dorišite velik nasmeh, kenguruju pa klobuk. Shranite pod novim imenom Kenguru s klobukom in Nasmejani delfin.

10. V jeziku HTML oblikujte naslednji seznam:

REGIJE IN VEČJA MESTA V SLOVENIJI

1. Pomurska regija
2. Podravska regija
 - Maribor
 - Ptuj
3. Koroška regija
4. Savinjska regija
 - Celje
 - Velenje
5. Zasavska regija
 - Trbovlje

6. Posavska regija
7. Jugovzhodna Slovenija
 - Novo mesto
8. Osrednjeslovenska regija
 - Ljubljana
 - Kamnik
9. Gorenjska regija
 - Kranj
10. Primorsko-notranjska regija
11. Goriška regija
12. Obalno-kraška regija
 - Koper/Capodistria

11. V jeziku HTML oblikujte naslednjo tabelo (v zadnji stolpec vstavite poljubne slike posameznih mest, ki jih najdete na spletu):

NAJVEČJA MESTA V SLOVENIJI PO ŠTEVILU PREBIVALCEV				
Št.	Ime mesta	Regija	Št. prebivalcev	Slike
1	Ljubljana	Osrednjeslovenska regija	258.837	
2	Maribor	Podravska regija	94.809	
3	Celje	Savinjska regija	37.490	
4	Kranj	Gorenjska regija	37.129	
5	Velenje	Savinjska regija	25.456	
6	Koper/Capodistria	Obalno-kraška regija	24.996	
7	Novo mesto	Jugovzhodna Slovenija	23.415	
8	Ptuj	Podravska regija	18.164	
9	Trbovlje	Zasavska regija	14.977	
10	Kamnik	Osrednjeslovenska regija	13.608	

12. Ustvarite dokument html s poljubnim besedilom in ostalimi elementi in ga oblikujte s pomočjo stilov, tako da bo končni izdelek vseboval spodnje oblikovanje.
- Spletna stran naj ima ozadje barve lavender.
 - Glavni naslov naj bo sredinsko poravnan.
 - Članek naj bo širine 450px in naj bo postavljen na sredino spletne strani (horizontalno). Obdaja naj ga črn neprekinjen rob debeline 1px. Ozadje naj bo belo, vsebina pa od robov odmaknjena za 5px.
 - En odstavek naj ima rdeče ozadje. Besedilo naj bo bele barve in velikosti 22pt. Poravnano naj bo na sredino. Notranji odmiki naj bodo 5px, spodnji zunanji odmik pa 15px.
 - Podpis naj bo sredinsko poravnan ter s črto zgoraj in spodaj.
 - Odstavki naj bodo obojestransko poravnani, prvo vrstico naj imajo zamaknjeno za 1 cm.
 - Bločni element, ki vsebuje sliko, naj bo postavljen ob desno stran tako, da ga ostalo besedilo obliva. Levi zunanji odmik naj bo 10px, obdaja pa naj ga neprekinjena črta barve lavender in debeline 1px.
 - Besedilo pod sliko naj bo sredinsko poravnano in naj ima 2px prostora zgoraj in spodaj.
 - Navedki in lokacije v besedilu naj bodo nagnjeni, osebe pa naj bodo izpisane krepko.
 - Bločni element z navedkom naj bo na levi strani (obdan z besedilom), na desni strani (zunaj) pa naj bo 10px prostora (med elementom in besedilom, ki ga obdaja). Njegova vsebina naj bo od robov odmaknjena 15px (zgoraj in spodaj) in 10px (levo in desno). Širina elementa naj bo 30% širine članka. Njegovo ozadje naj bo barve lavender. V ozadju elementa naj bo še slika quotes.png, postavljena na spodnjo desno stran brez tlakovanja.
13. Naredite spletno stran, ki bo imela izpisane vse vrste naslovov od <H1> do <H6>, prva dva poravnana desno, druga dva na sredino in tretja dva na levo stran spletne strani.
14. Naredite spletno stran, ki bo vsebovala osem odstavkov. Prvi naj bo napisan krepko, drugi nagnjeno, tretji podčrtan, četrti prečrtan, peti naj bo v pisavi pisalnega stroja, šesti naj bo napisan z drugo barvo, sedmi z drugačno pisavo in osmi z drugo velikostjo črk.

15. Naredite spletno stran z modrim ozadjem, rumene barvo pisave, kjer v odstavku uporabite ukaz za preskok v novo vrstico ter na koncu naredite črto dolžine pol ekrana in debeline 5.
16. Naredite spletno stran, kjer boste predstavili v obliki neoštevilčenega seznama vrste zelenjave.
17. Naredite enak seznam kot pri prejšnji nalogi, vendar sedaj seznam oštevilčite najprej z malimi črkami, potem z arabskimi števili in nato še z velikimi rimskimi števili.
18. Naredite spletno stran s tremi povezavami do spletnih iskalnikov in na koncu še povezavo do vase e-pošte.
19. Naredite spletno stran, ki naj vsebuje tri slike, poravnaj bo poravnana levo, druga na sredino in tretja desno. Določite jim tudi višino in širino.
20. Naredite tabelo z robom 2, napisom nad tabelo in štirimi vrsticami ter tremi stolpci.

4 ELEKTRONSKA POŠTA

Elektronska pošta je način sestavljanja, pošiljanja in sprejemanja sporočil preko računalniških omrežij in drugih komunikacijskih sistemov. Omogoča izmenjavo sporočil s posamezniki ali skupinami ljudi po vsem svetu. Preko elektronske pošte lahko pošiljamo tudi različne datoteke. Internetno standard za pošiljanje elektronske pošte je SMTP (angl. Simple Mail Transfer Protokol). Za dostopanje do poštnega predala pa se uporabljata protokola POP3 in IMAP.

4.1 Sestava naslova elektronske pošte

Naslov elektronske pošte je sestavljen iz:

- naslovnika,
- @,
- naziva poštnega strežnika.



Slika 8: Sestava naslova elektronske pošte

4.2 Pravila pri uporabi elektronske pošte

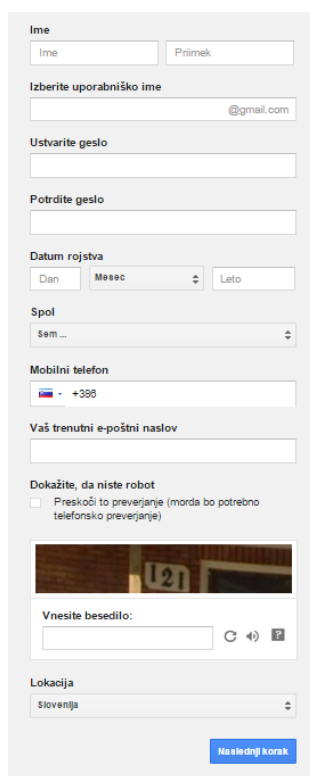
Osnovna pravila elektronske pošte so:

- elektronska sporočila naj bodo kratka in jedrnata, saj je namen elektronske pošte hitro sporočanje,
- elektronsko sporočilo naj vsebuje zadevo, ki naj odraža vsebino elektronskega sporočila,
- ne pošiljajmo verižnih pisem,
- v elektronskem sporočilu ne uporabljajmo velikih črk, saj to pomeni vpitje,
- v primeru pošiljanja reklamnih sporočil in oglasov je potrebna predhodna pridobitev soglasja prejemnikov,

- ob pošiljanju sporočila s priponko se je potrebno prepričati, da ima prejemnik potreben program za odprtje te datoteke,
- ne predstavljajmo se pod lažnim imenom in ne pišimo elektronskih sporočil v imenu nekoga drugega,
- v elektronskem sporočilu ne uporabljamo kratic in narečja,
- pazimo pri pošiljanju čustvenih ikon (smeškov).
- uporabljamo polja BC in BCC

4.3 Uporaba brezplačnega servisa za elektronsko pošto Gmail

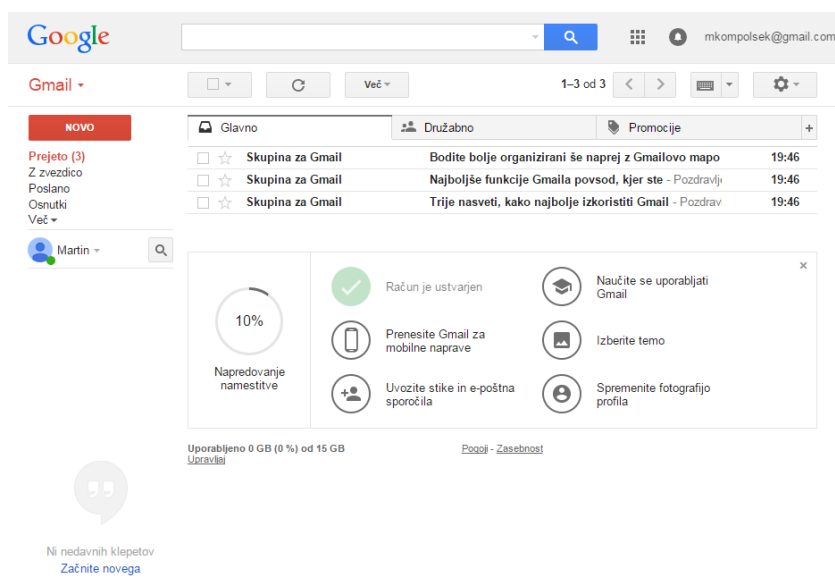
Gmail je storitev brezplačne elektronske pošte, ki jo nudi podjetje Google. Poštni predal ustvarimo tako, da najprej vtipkamo URL naslov www.gmail.com in kliknemo na možnost Ustvarite račun. Prikaže se nam spletni obrazec.

The image shows a web form for creating a new Gmail account. The form is titled 'Ustvarite račun' (Create account) and contains several input fields and a checkbox. The fields are: 'Ime' (Name) with sub-fields for 'Ime' and 'Preimek'; 'Izberite uporabniško ime' (Choose a username) with a text box and '@gmail.com' suffix; 'Ustvarite geslo' (Create password) and 'Potrdite geslo' (Confirm password) text boxes; 'Datum rojstva' (Date of birth) with dropdowns for 'Dan' (Day), 'Mesec' (Month), and 'Leto' (Year); 'Spol' (Gender) with a dropdown menu; 'Mobilni telefon' (Mobile phone) with a dropdown for country code and a text box for the number; 'Vaš trenutni e-poštni naslov' (Your current email address) text box; a checkbox for 'Dokažite, da niste robot' (Prove you're not a robot) with a link to skip verification; a CAPTCHA image showing a house with the number 121; a text box for 'Vnesite besedilo:' (Enter text); and a 'Lokacija' (Location) dropdown menu set to 'Slovenija'. A blue button labeled 'Naslednji korak' (Next step) is at the bottom right.

Slika 9: Spletni obrazec za ustvarjenje novega računa

Izpolnimo potrebne podatke in kliknemo na gumb Naslednji korak. V primeru, da smo vse podatke izpolnili pravilno in da ni drugih težav, prejmemo obvestilo o uspešnem ustvarjenju računa.

Po uspešno ustvarjenem računu, se s svojim uporabniškim imenom in geslom prijavimo v Googlov poštni nabiralnik in odpre se naš poštni predal.



Slika 10: Poštni predal

Prejeto elektronsko sporočilo odpremo s klikom na želeno sporočilo.

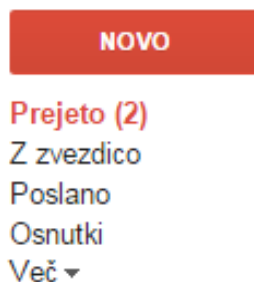


Slika 11: Odpiranje elektronsko sporočila

Po končani uporabi poštne predala se obvezno odjavimo.

4.3.1 Sestavljanje in pošiljanje novega sporočila

Novo elektronsko sporočilo sestavimo in pošiljamo tako, da kliknemo na gumb Novo.



Slika 12: Gumb Novo

Ob kliku se odpre pogovorno okno za pošiljanje novega sporočila. V polje Za vtipkamo elektronski naslov naslovnika, v polje Zadeva vtipkamo predmet sporočila ter natipkamo sporočilo in kliknemo na gumb Pošlji. Če želimo isto sporočilo poslati hkrati tudi kot kopijo še nekomu naslovniku, kliknemo na možnost Kp (v desnem zgornjem kotu pogovornega okna). Odpre se polje za vnos elektronskega naslova prejemnika kopije. V primeru, da želimo skriti naslove in imena prejemnikov pred drugimi, lahko pošljemo skrito kopijo, tako da izberemo možnost Skp. Če pošiljamo elektronsko sporočilo več naslovnikom, naslove med seboj ločimo s podpičjem in presledkom. Včasih želimo poleg elektronskega sporočila poslati še datoteko. V tem primeru elektronskemu sporočilu priložimo priponko, tako da kliknemo na gumb Priložite priponko in izberemo želeno datoteko. Postopek ponovimo za vsako priponko, ki jo želimo pripeti elektronskemu sporočilu.



Slika 13: Gumb Priložite priponko

4.3.2 Odgovor na sporočilo

Ko v poštni nabiralnik prejmemo elektronsko sporočilo, ga odpremo in preberemo. Na sporočilo lahko tudi odgovorimo, tako da kliknemo na gumb Odgovori. V sporočilu se avtomatsko izpiše datum, naslov pošiljatelja in vsebina pošiljateljevega sporočila.

Simon

Pozdravljeni,

potrjujem udeležbo na sestanki.

Lepo pozdrav,
delavec

Dne 12. avgust 2015 20:20 je Simončič, Samo <samo.simoncic@fs.uni-lj.si> napisal/-a:

Pozdravljeni,

sporočam vam, da bo sestanek zaposlenih v petek, 14. 8. 2015 ob 8. uri v prostorih sejne sobe podjetja.

Lep pozdrav,
direktor

Slika 14: Odgovor na sporočilo

4.3.3 Posredovanje sporočila drugemu naslovniku

V primeru, da želimo prejeto sporočilo posredovati nekemu drugemu prejemniku, pritisnemo na možnost Posreduj.

delavec@gmail.com

Fwd: Sestanek

Pozdravljeni,

posredujem vam spodnje sporočilo.

Lep pozdrav,
posredovalec sporočila

----- Posredovano sporočilo -----

Od: **Simon i. Sam** <sam_simon_i._@f._uni-._si>
Datum: 12. avgust 2015 20:20
Zadeva: Sestanek
Za: "melita.kompolsek@gmail.com" <melita.kompolsek@gmail.com>

Pozdravljeni,

sporočam vam, da bo sestanek zaposlenih v petek, 14. 8. 2015 ob 8. uri v prostorih sejne sobe podjetja.

Lep pozdrav,
direktor

Slika 15: Posredovanje sporočila drugemu naslovniku

V polje Za vtipkamo elektronski naslov naslovnika. V polju Zadeva pa se pojavi napis Fwd in ime sporočila. Lahko vpišemo še morebitno sporočilo za naslovnika in kliknemo na gumb pošlji.

4.4 Povzetek

To poglavje zajema snov sodobnega sporočanja, torej elektronske pošte. Spoznali bomo sestavo naslova elektronske pošte in pravila pri uporabi elektronske pošte. Na koncu poglavja pa se bomo podrobneje seznanili z enim od najpogostejše uporabljenih brezplačnih servisov za elektronsko pošto.

4.5 Vprašanja in naloge za preverjanje znanja

1. Kaj je spletna pošta?
2. Kako je sestavljen naslov elektronske pošte?
3. Naštete pravila pri uporabi elektronske pošte.
4. Ustvarite nov Gmail račun in pošljite testno sporočilo znancu.
5. Kakšna je razlika med pošiljanjem kopije in skrite kopije?
6. Na spletu poiščite fotografijo mačke in ji dorišite brke v Slikarju. Sliko pošljite preko elektronske pošte na prijateljev elektronski naslov.
7. Na spletu poiščite in prekopirajte na prazen list v Wordu ter shranite:
 - vsa prosta delovna mesta v občini Maribor,
 - telefonsko številko Zdravstvenega doma Maribor,
 - fotografijo Koče na Arehu,
 - fotografijo vinske trte.Rezultate v Wordu pošljite preko e-pošte na prijateljev naslov.
8. Napišite kratko elektronsko sporočilo in ga skupaj s poljubnimi tremi slikami (s spleta) pošljite na štiri prijateljeve elektronske naslove.

5 DIGITALNI PODPIS

Digitalen podpis je podpis narejen z uporabo kriptografije in omogoča ugotavljanje verodostojnosti sporočila ter istovetnosti njegovega avtorja. Za digitalni podpis dokumenta potrebujemo osebni ključ oz. digitalno potrdilo. Osebni ključ je shranjen na računalniku kot kombinacija zasebnega ključa, ki mora ostati tajen, in javnega ključa, ki ga dodamo svojim dokumentom, ko jih podpišemo. Ko vnesemo digitalni podpis v dokument, se iz vsebine dokumenta in osebnega ključa izračuna neke vrste nadzorna vsota. Nadzorna vsota in javni ključ se skladiščita skupaj z dokumentom.

5.1 Odpiranje podpisanega dokumenta

Ko nekdo odpre dokument, ustrezen program ponovno izračuna nadzorno vsoto in jo primerja s shranjeno. Če sta vsoti enaki, bo program sporočil, da gledamo izvirni, nespremenjen dokument. Poleg tega lahko program iz digitalnega potrdila pokaže tudi podatke o javnem ključu. Javni ključ lahko primerjamo z javnim ključem, ki je objavljen na spletni strani pooblaščne institucije, ki je izdala potrdilo. Kadar koli nekdo spremeni kar koli v dokumentu, ta sprememba prelomi digitalni podpis. Po spremembi se ne bo pojavil znak, ki bi nam povedal, da gledamo izvirni dokument.

5.2 Varnostna opozorila

Ko prejmemo podpisan dokument in nam programska oprema sporoči, da je podpis veljaven, to še ne pomeni, da smo lahko povsem prepričani, da je dokument enak tistemu, ki ga je poslal pošiljatelj. Podpisovanje dokumentov s programskimi potrdili ni povsem varna metoda. Obstajajo načini izogibanja varnostni zaščiti.

5.3 Povzetek

V dobi spleta in elektronike je izjemnega pomena digitalen podpis, ki omogoča ugotavljanje verodostojnosti sporočila ter istovetnosti njegovega avtorja, zato bomo v tem poglavju spoznali definicijo digitalnega podpisa in odpiranje tako podpisanih dokumentov. Na koncu pa se bomo seznanili še z varnostnimi opozorili ob prejemu podpisanega dokumenta.

5.4 Vprašanja in naloge za preverjanje znanja

1. Kaj je digitalni podpis?
2. Kako deluje digitalni podpis?
3. Na kaj moramo paziti pri odpiranju dokumenta z digitalnim podpisom?
4. Ali je dokument podpisan s programskim potrdilom popolnoma varen?

6 ZAKLJUČEK

Z razvojem računalniških tehnologij se potreba po računalniški pismenosti in delu s spletom nenehno povečuje, zato so v gradivu predstavljene osnove. V vsakem izmed poglavji je na kratko predstavljena osnovna teorija izbrane tematike, ki je v nadaljevanju podkrepljena z različnimi praktičnimi primeri in vprašanji za preverjanje znanja. Predstavljene tematike so bile izbrane na osnovi kataloga znanja. V osnovi je gradivo namenjeno vsem začetnikom in prav tako vsem tistim, ki bi želeli nadgraditi oziroma obnoviti svoje znanje o spletu.