



Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

# **Računalniški praktikum**

**Franc Solina**

## Namen predmeta

- Osvežiti in poenotiti znanje osnovnih programskih paketov, ki bodo potrebni med študijem.
- Uporaba odprte programske opreme (angl. Open Source).
- Nekaj teorije (1 ura predavanj), veliko prakse (3 ure laboratorijskih vaj).

## Vsebina

1. Uvod
2. Pravna zaščita programske opreme
3. Uporabniški vmesniki
4. Operacijski sistemi (Linux)
5. Urejanje besedil (OpenOffice.org Writer in  $\text{\LaTeX}$ )
6. Preglednice (OpenOffice.org Calc)
7. Računalniška grafika (GIMP in OpenOffice.org Draw)
8. Govorne predstavitve (OpenOffice.org Impress)
9. Matematični programi (Mathematica in Matlab)
10. Medmrežje in svetovni splet (HTML in XML)

## Izvajanje predmeta

- obvezen zagovor laboratorijskih vaj v dveh delih:
  1. paket: Linux, Writer, L<sup>A</sup>T<sub>E</sub>X, Calc v sredini semestra
  2. paket: GIMP, Draw, Mathematica, HTML na koncu semestra
- pisni izpit z nalogami iz snovi s predavanj

## Literatura

- Učbenik *Uporabniška programska oprema*, 3. izdaja izzide meseca oktobra 2004
- priložena zgoščenska z vsemi odprto kodnimi programi
- prosojnice v pdf formatu na spletni strani predavatelja

## Pravna zaščita programske opreme

- prosto programje / odprto programje (dostop do izvorne kode!)
- neprosto oz. lastniško

## Svoboščine prostega programja

**prostost 0** : Program lahko svobodno poganjamo za kakršenkoli namen.

**prostost 1** : Lahko proučujemo delovanje programa in ga prilagajamo svojim potrebam. Za to je seveda nujen dostop do izvorne kode (ang. *source code*).

**prostost 2** : Program lahko prosto (brezplačno ali proti plačilu) razširjamo naprej.

**prostost 3** : Program lahko izboljšujemo in po želji javno objavljamo izboljšane verzije za korist skupnosti. Programe, ki so prosti, lahko po mili volji uporabljamo, spreminjamo in javno objavljamo in nikogar ni potrebno vprašati ali obvestiti. Lahko jo distribuiramo brezplačno ali proti plačilu.

## Ohranjanje prostega programja

- Programi, ki so prosti in niso zakonsko zaščiteni, spadajo v kategorijo **programja v javni lasti** (angl. *Public Domain Software*).
- Če je program v javni lasti, lahko kdorkoli napravi lastniško verzijo prej prostega programa.
- Das se status prostega programja ohrani, ga je potrebno ustrezno zaščititi.



## Licenca GNU General Public Licence

- leta 1984 na pobudo Richarda Stallmana,
- razlog za ustanovitev projekta je bil soliološko-etični, v smislu pozitivnega prispevka k skupnosti,
- vse nadaljnje verzije prostega programa morajo ostati proste: metoda **“Copyleft”**
- prepoveduje kombiniranje prostih programov zaščitenih s copyleftom s programi, ki niso prosti, saj je taka kombinacija nujno neprost program,
- ko je program enkrat zaščiten po načelu copyleft, ostane prost za vedno.

## Kako zaščitimo program z licenco GNU GPL?

Če ste torej razvili svoj program in želite, da bi se distribuiral kot prosta programska oprema z licenco GNU GPL, na začetek vsake datoteke z izvorno kodo dodajte naslednje vrstice:

- V prvi vrstici navedite ime programa in kratek opis njegovega delovanja.
- *Copyright (C) ime avtorja*  
*This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.*  
*This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.*  
*You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.*
- Na koncu pripišite tudi svoj kontaktni naslov: elektronski in/ali navadni naslov.

## Odprto programje (angl. Open Source)

- filozofija podobna kot pri prostem programju: dati ljudem možnost, da pregledajo in popravijo ter dopolnijo programe, saj to vodi h kvalitetnejši programski kodi,
- vendar ne zahteva distribucije identičnih ali spremenjenih kopij pod isto licenco, kot je določeno pri prostem programju,
- možno je vgraditi kodo odprtega programja v program prostega programja, obratno ni možno,
- odprta koda se lahko povezuje z lastniško kodo

## Lastniško programje (angl. proprietary software)

- prepoveduje oziroma omejuje uporabo, razširjanje in spreminjanje programov.
- pojma lastniško programje ne smemo zamenjevati s pojmom komercialno programje (angl. *commercial software*),
- cilj komercialnega programja je zaslužek.
- Komercialni programi so lahko prosti ali neprosti. Prav tako obstaja tudi nekomercialno programje, ki ni prosto.

## Komercialna programska licenca

- daje pravico do uporabe programskega izdelka in hkrati določa, na kakšen način lahko program uporabljamo in distribuiramo. izdelek,
- na koliko računalnikov ga program lahko namestimo,
- kako pogosto želimo verzijo programa nadgraditi, ali želimo o izdelku kakšno dodatno izobraževanje,
- kakšen bo način plačila itd.
- V nekaterih primerih pa programsko opremo že kupimo skupaj s strojno opremo, primer so IBM-ovi prenosniki, kjer poleg računalnika kupimo tudi Microsoftov operacijski sistem Windows.

## Ugodnosti za izobraževalne ustanove

- FRI je pristopila k programu Microsoft Developer Network Academic Alliance (MSDNAA), ki omogoča pridobitev Microsoftovih razvojnih orodij, platform in strežnikov za uporabo v izobraževalne in raziskovalne namene.
- Do uporabe programske opreme iz programa MSDNAA so upravičeni vsi redni študenti Fakultete za računalništvo in informatiko ter zaposleni na Fakulteti za računalništvo in informatiko.
- Vsi, ki so upravičeni do uporabe programske opreme iz programa MSDNAA, lahko dobijo programsko opremo iz programa MSDNAA tudi za uporabo na osebem računalniku doma v pedagoške in/ali nekomercialne raziskovalne namene.

## Še druge vrste programov

- Programi na pokušino (angl. *Shareware*) so delno ali popolnoma funkcionalni javno dostopni programi (ali podatki), ki jih lahko razširjamo naprej.
- demonstracijske različice (angl. *demo version*):
  1. Zgodnja ne povsem funkcionalna verzija programa v razvoju.
  2. Posebna, navadno okrnjena verzija že končanega programa, ki jo lahko dobimo brezplačno ali pa za minimalno protiplačilo.
- Poskusna različica (angl. *trial version*) programov lahko brezplačno namestimo na računalnik. Njihovo delovanje je največkrat časovno omejeno na nekaj ur ali nekaj dni.
- Zastonjsko programje (angl. *freeware*)

- Programi, ki prikazujejo reklamne oglase (angl. *adware*) so brezplačni programi. Finančno so podprti s pomočjo oglasov, ki so vključeni v program.



## Avtorska zaščita programske opreme

- Računalniški programi sodijo po slovenski in mednarodni zakonodaji med avtorska dela.
- Programsko opremo obravnava:
  - 60. člen Ustave Republike Slovenije
  - Zakon o avtorskih in sorodnih pravicah (kratica ZASP), UL RS št. 21/95 s spremembami in z dopolnitvami, UL RS št. 9/2001, UL RS št. 43/2004
  - Kazenski zakonik Republike Slovenije, UL RS št. 63/94 s spremembami in dopolnitvami, UL RS št. 23/99, UL RS št. 40/2004
  - Zakon o preprečevanju omejevanja konkurence, UL RS št. 56/99
  - Zakon o sodiščih, UL RS št. 19/94, 38/99 in 28/2000

- Zakon o dohodnini, UL RS št. 71/93 s spremembami in dopolnitvami, UL RS št. 7/95, 44/96

## Sankcije

- Sankcije za nelegalno uporabo so podrobneje definirane v Kazenskem zakoniku RS in v Zakonu o avtorski in sorodnih pravicah.
- Kazenski zakonik določa do 8 let zaporne kazni za nelegalno izkoriščanje računalniških programov.
- ZASP med drugim določa tudi kazen za nelegalno uporabo avtorsko zaščitene del v znesku od 400.000 do 10.000.000 SIT in pooblašča tržno inšpekcijo za nadzor nad legalno uporabo.

## Kaj je piratstvo?

Obstaja pet osnovnih oblik programskega piratstva:

1. **Ponarejanje** (angl. *Counterfeiting*), pri katerem gre za neavtorizirano reproduciranje in distribuiranje zavarovanih programov na disketah ali CD-ROM-ih v opremi (embalaži), ki je ponavadi tudi ponarejena;
2. **Nalaganje na disk** (angl. *Hard Disc Loading*), pri katerem gre za to, da proizvajalci ali prodajalci pri prodaji računalnikov brezplačno opremijo računalnik z neavtoriziranimi programi, s čimer želijo pridobiti kupce;
3. **Mehko piratstvo** (angl. *Softlifting*), pri katerem podjetje nabavi samo eno legalno kopijo programa in jo neavtorizirano reproducira na vse svoje računalnike oziroma svoje delavce;
4. **Dajanje v najem** (angl. *Software Rental*) primerkov avtorskega dela, npr. prodajalec posodi programsko opremo za uporabo na

izposojevalčevem domačem ali službenem računalniku;

5. **Internetno piratstvo**, kjer gre za neavtorizirano naložitev računalniškega programa na spletno stran oziroma nezakonito razpečavanje s pomočjo interneta (angl. slengovski izraz za piratske programe, filme ipd. je *Warez*).

## Nadzor programske opreme v Sloveniji

- BSA (*Business Software Alliance*) je mednarodna organizacija, glasnik vodilnih proizvajalcev in prodajalcev programske in strojne opreme
- Več let deluje tudi v Sloveniji. Njen glavni namen je obveščanje in izobraževanje javnosti o upravljanju s pravicami na računalniških programih, avtorsko pravno zaščito, zaščito v kibernetnem prostoru, elektronskem poslovanju in z internetom povezanimi vprašanji ter podpora organom pregona pri aktivnostih za zatiranje nedovoljene reprodukcije, distribucije in uporabe nelicenčnih računalniških programov.

## Zakonsko določene avtorske pravice

- Ker računalniški programi sodijo med avtorska dela, programerejm pripadajo vse pravice iz vsebine avtorskega prava (ZASP). To so moralne avtorske pravice, materialne avtorske pravice in druge pravice avtorja.
- **Moralne avtorske pravice** varujejo avtorja glede njegovih osebnih vezi do dela. Tako ima avtor pravico določiti, kje in kdaj bo njegovo delo prvič objavljeno. Avtor ima pravico določiti, s kakšno oznako naj se navede njegovo avtorstvo, pravico, da se upre skazitvi ali kakšnemu drugemu posegu v njegovo delo.
- **Materialne avtorske pravice** varujejo premoženjske interese avtorja s tem, da avtor izključno dovoljuje ali prepoveduje uporabo svojega dela.

## ZASP

ZASP ima tudi posebej oddelek namenjen izključno računalniškim programom (četrto poglavje, 2. oddelek). Tu je definiran pojem računalniškega programa in njegovega varstva na naslednji način:

1. Računalniški programi po tem zakonu so programi v vsaki izrazni obliki, vključno s pripravljalnimi gradivom za njihovo izdelavo.
2. Ideje in načela, ki so osnova nekemu elementu računalniškega programa, vključno s tistimi, ki so osnova njegovim vmesnikom, ne uživajo varstva.
3. Računalniški programi uživajo varstvo, če so individualna dela v tem smislu, da pomenijo lastno intelektualno stvaritev njihovega avtorja.
4. kadar računalniški program ustvari delojemalec pri izpolnjevanju svojih obveznosti ali po navodilih delodajalca, ali ga ustvari avtor po avtorski pogodbi o naročilu, se šteje, da so materialne avtorske pravice in druge



pravice avtorja na tem programu izključno in neomejeno prenesene na delodajalca ali naročnika, če ni s pogodbo drugače določeno.

## Programski patenti

- Programski patenti so zadnjem času zelo vroča tema v Evropskem parlamentu.
- Patent definiramo kot pogodbo med izumiteljem in javnostjo. Izumitelj v zameno za časovno in prostorsko omejen monopol objavi svoj izum. Na ta način naj bi se vzpodbujala izumiteljska dejavnost, saj z objavo svojega znanja omogoči drugim, da to razvijajo in izboljšujejo naprej.
- Za programske patente se zavzemajo velike korporacije, ki si patente lahko privoščijo. Patentiranje namreč ni poceni, se pa splača, saj kasneje lahko prinaša velike dobičke.
- Pristaši odprtega programja zelo intenzivno nasprotujejo programskim patentom. Trdijo, da bi patenti preprečili zdrav konkurenčni razvoj programja.

- Nekateri scenariji, ki jih navajajo kot možne posledice legaliziranja programskih patentov, so: Evropski patentni urad (angl. European Patent Office, EPO) je že podelil več kot 30.000 programskih patentov, ki pa jih zaenkrat še ni mogoče uveljaviti.
- Že podeljeni patenti vsebujejo tudi operator XOR, prikaz poteka (angl. progress bar), uporaba t.i. null operatorjev za upočasnitev dogajanja, uporaba dveh barv za prikaz popravkov v dokumentu, drsni trak (angl. scroll bar), uporaba miškega klika za nakupovanje preko spleta itd.
- Vse te metode so vključene v velik del že obstoječe odprte programske kode. Legalizacija takih programskih patentov bi ogrozila obstoj odprtega programja, saj je uporabo patentov potrebno plačati.
- Posledica programskih patentov v ZDA se kaže tudi v skrajnostih, kot je npr. primer formatov MP3 in GIF. V ZDA je zakonsko prepovedano

pisanje komercialnih programov, ki bi shranjevali zvok v format MP3 oz. sliko v format GIF brez plačila ustrezne licenčnine.

## Razvoj uporabniških vmesnikov

- lučke, papirnati trak in kartice,
- monitor, tipkovnica
- računalniška miška (1968, Douglas Englebart)
- virtualna realnost
- sinteza in razpoznavanje govora
- interpretacija slik
- ambient intelligence

## Vrste uporabnikov

- nekdam le posebej usposobljeni strokovnjaki
- danes široka množica različnih uporabnikov:
  - začetniki
  - občasni uporabniki
  - redni uporabniki
- kako ustreči zahtevam vseh vrst uporabnikov?

## **Abstrakcijski nivoji uporabniških vmesnikov**

- konceptualni nivo
- semantični nivo
- sintaktični nivo

## Lastnosti uporabniških vmesnikov

**popolnost:** uporabnik lahko izrazi vse ukaze,

**skladnost:** v podobnih situacijah naj se uporablja podobna zaporedja akcij,

**učinkovitost:** hitra izvedba s čim manj možnimi napakami (možnost definiranja bližnjic),

**razširljivost:** dodajanje novih pojmov s pomočjo obstoječih,

**odzivnost:** računalnik se mora ustrezno odzvati na uporabnikov ukaz,

**možnost popravljanja napak:** obrnljivost akcij, detekcija napak,

**nudenje pomoči:** enostavna sporočila na zaslonu, na zahtevo daljša razlaga, predvsem o sintaksi vhodnih ukazov.



## Osnovni načini interakcije z računalnikom

- ukazni jezik
- izpolnjevanje formularjev
- izbira preko menujev
- neposredna manipulacija

## Ukazni jezik

**Zahtevana oprema:** tipkovnica in računalniški zaslon.

**Način dela:** uporabnik tipka vse ukaze.

**Prednosti:**

- hitrost (pomembna za pogoste uporabnike!),
- fleksibilnost (možnost tvorjenja makro ukazov),
- najmanj zahtev glede strojne in programske opreme.

**Slabosti:**

- dolgo šolanje,
- veliko pomnjenja in zato hitro pozabljanje ukazov.

## Primeri:

- operacijska sistema DOS in Unix,
- urejevalnik besedila vi na Unixu.

## Izpolnjevanje formularjev

**Zahtevana oprema:** tipkovnica in računalniški zaslon.

**Način dela:** uporabnik s tipkanjem vnaša podatke v vnaprej pripravljena polja. Med polji se premika s pomočjo tipke Enter>.

### Prednosti:

- poenostavljen vnos podatkov,
- posnemanje formularja na papirju,
- kratko šolanje.

### Slabosti:

- namenjeno specifični, ponavljajoči se nalogi (npr. kadrovska evidenca),
- požrešno glede prostora na zaslonu.

## Primeri:

- različni uporabniški programi,
- izpopolnjena oblika formularjev so **preglednice**, kjer imajo celice več funkcionalnosti (matematične in logične operacije nad več celicami). Dodani so elementi direktne manipulacije. Preglednice obravnavamo v 4. poglavju.

## Izbira preko menijev

**Zahtevana oprema:** tipkovnica, računalniški zaslon in miška.

**Način dela:** uporabnik s pomočjo miške ali tipk izbira med ponujenimi ukazi.

### Prednosti:

- kratko šolanje (namesto pomnjenja ukazov je potrebno le njihovo prepoznavanje),
- strukturirano (hierarhično) odločanje.

### Slabosti:

- zavzame veliko prostora in včasih oteži izbiro zaradi prevelikega števila ukazov (zato izvlečni in dvižni meniji)
- upočasnuje pogoste uporabnike.

## Primeri:

- bančni avtomati,
- okenski uporabniški vmesniki,
- mobilni telefoni.

## Neposredna manipulacija

**Zahtevana oprema:** tipkovnica, zaslon (stereo očala, čelada VR), miška (podatkovna rokavica).

**Način dela:** uporabnik s pomočjo miške ali drugih interaktivnih vmesnikov izbira objekte na zaslonu (ali v virtualnem prostoru) in z njimi upravlja.

Vizualno je potrebno predstaviti vse objekte. Okolja, v katero so objekti postavljeni, imenujemo namizje in okna; datoteke, imeniki in posamezni programi pa so na namizju predstavljeni z ikonami.

### Prednosti:

- hitro učenje,
- uporabnik neposredno vidi, če akcije vodijo k cilju,
- uporabnik pridobi samozaupanje in nadzor.



## Slabosti:

- zahtevna za programiranje,
- zmogljiva strojna oprema,
- vseh nalog ne moremo opisati s konkretnimi objekti in neposredno akcijo.

## Primeri:

- Xerox Star,
- Apple Macintosh,
- Microsoft Windows,
- X Windows
- video igre.

## Elementi grafičnih uporabniških vmesnikov

- namizje (angl. *desktop*),
- ikone (programi, datoteke, mape – direktoriji),
- okna (imenik ali področje izvajanja programa), drsniki za določanje izreza, in druge kontrole, ki omogočajo premikanje in spreminjanje velikosti oken,
- meniji (izvlečni, dvižni, kontekstni),
- miška, ki omogoča klik, dvojni klik, izbiro in premik,
- animacije.

## Oblikovanje uporabniških vmesnikov

- problem (industrijskega) oblikovanja
- realizirajo jih programerji, ki se za to niso šolali
- oblikovanja se učijo predvsem na umetniških šolah, ti pa tehnologije do popolnosti ne obvladajo
- sklepanje kompromisov
- ni preprostih receptov

## E. S. Raymond: The Art of Unix Programming

1. Modularnost: Pišite enostavne dele povezane z jasnimi vmesniki.
2. Jasnost: Jasnost je boljša kot domiselnost.
3. Kompozicija: Oblikujte programe tako, da bodo povezani z drugimi programi.
4. Separacija: Ločite politiko od mehanizma; ločite vmesnik od motorja.
5. Enostavnost: Oblikujte za enostavnost; kompleksnost dodajte samo tam, kjer je ta nujna.
6. Minimalnost: Program naj bo velik le, če lahko demonstrirate, da ne bo šlo drugače.
7. Transparentnost: Oblikujte za transparentnost, da bosta nazdor in razhroščevanje lažja.

8. Robustnost: Robustnost je otrok transparentnosti in enostavnosti.
9. Reprezentacija: Shranite znanje v podatke, da bo program lahko neumen in robusten.
10. Najmanjše presenečenje: Pri oblikovanju vmesnikov vedno naredite najmanj presenetljivo stvar.
11. Tišina: Če program nima nič presenetljivega za povedati, naj bo tiho.
12. Popravljanje: Če je nujno, da se naloga prekine, potem naj se to zgodi čimprej in opazno.
13. Ekonomičnost: Programerjev čas je dragocen, varčujte z njim bolj kot z računalnikovim časom.
14. Generiranje: Izogibajte se ročnemu delu; pišite programe ki pišejo programe, kadar lahko.

15. Optimiziranje: Naredite prototip pred končno verzijo. Poskrbite, da najprej dela, šele potem optimizirajte.
16. Različnost: Ne verjemite trditvam o eni pravi-in-edini metodi.
17. Razširljivost: Oblikujte za prihodnost ker bo tu prej kot si mislite.

## Nadzor nad okoljem osrečuje uporabnika

- Uporabnik je zadovoljen takrat, ko *uspešno* opravi neko *nalogo*
- Uporabniški vmesnik je dober takrat, ko se program obnaša natanko tako, kot to pričakuje uporabnik
- Nepričakovano obnašanje in presenečenja uporabnika ne razveselijo.
- Vsakič, ko neka uporabnikova akcija ne privede do pričakovanega rezultata, doživi uporabnik majhno frustracijo.
- Vsakič, ko program ponudi izbiro med več možnostmi, od uporabnika zahteva, da sprejme odločitev.

## Uporabniki ne berejo navodil

- Uporabniki posežejo po navodilih šele takrat, ko se kaj zaplete
- Za programe naj sploh ne bi potrebovali navodil, kakor jih ne potrebujemo za televizor ali pralni stroj.
- V grafičnem okolju naj bi programi delovali na podoben način
- Čarovniki nalogo razdelijo na korake, ob vsakem koraku uporabnik vnese samo bistvene informacije



## Operacijski sistemi

Operacijski sistem je poseben program, ki upravlja z delovanjem računalnika in skrbi za

- uporabniški vmesnik,
- organizacijo datotek na pomnilnih enotah in
- za nadzor nad vhodno-izhodnimi enotami.

## Delitev operacijskih sistemov

**število uporabnikov:** enouporabniški ali večuporabniški sistemi,

**število hkratnih procesov:** enoopravilni ali večopravilni sistemi,

**vrste obdelav:** paketne, interaktivne, paralelne, v realnem času.

**uporabniški vmesnik:**

- ukazni (MS-DOS, Unix),
- grafični (Macintosh OS, X Window System, MS Windows, OS/2),

**strojna platforma:**

- na različnih platformah (Unix),
- samo na istovrstni strojni platformi (npr. Microsoft Windows na PC kompatibilnih računalnikih),

- možna pa je programska emulacija operacijskega sistema (npr. MS Windows na računalnikih Macintosh).

**cena:**

- brezplačni (Linux),
- plačljivi (MS Windows).

## Unix

- Unix so razvili v Bellovih laboratorijih že pred desetletji
- v 80-tih letih se je Unix razširil na Sun-ovih delovnih postajah
- na različnih platformah so se razvile različne med seboj nekompatibilne različice Unixa (HP, IBM, Silicon Graphics, Apollo ...)
- nov zagon razvoju in uporabi Unixa je dal Linux
- Linus Torvalds, ki je razvil jedro operacijskega sistema, je izvorno kodo v celoti objavil in omogočil njeno brezplačno razširjanje in dopolnjevanje
- različice Linuxa (RedHat, Fedora, Suse, Mandrake, Debian, ...) so preko interneta dostopne brezplačno, kupiti pa se jih da tudi na zgoščenkah
- na Unixu temelji tudi Macintoshov operacijski sistem OS X, ki je nadgrajen s preverjenim Macintoshovim grafičnim vmesnikom.

## Urejanje besedil

- računalniki so se koncem 70-tih let začeli uporabljati za oblikovanje besedil
- danes so povsem izrinili pisalne stroje
- računalniki so povzročili revolucijo v tiskarski in založniški industriji
- obstaja cela vrsta aplikacij za oblikovanje besedil
- *namen dobre tipografije je lažje **razumevanje** vsebine besedila!*

## Načini urejanja besedil

**vizualno** urejanje po načelu “kar vidiš, to dobiš” (angl. WYSIWYG):

- rezultati vseh uporabnikovih akcij takoj vidni na zaslonu
- delo z njimi je konceptualno zelo podobno pisanju na pisalnem stroju
- PRIMERI: Wordstar, Write, MS Word, OpenOffice.org Writer

**logično** urejanje besedil:

- izhaja iz tradicije tiskarstva
- v rokopisu je potrebno označiti tip in velikost črk, poravnavo besedila
- s pomočjo stilskih datotek ločitev vsebine od oblike
- PRIMERI: SGML, Scribe, Nroff, Troff, T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, HTML in XML
- nekatere preproste elemente logičnega urejanja s pomočjo stilskih datotek poznajo tudi vizualni urejevalniki besedil

## Pisave

- pisava je eden največjih človeških izumov
- velik del sodobnega sveta uporablja latinico za pisanje svojega jezika
- latinsko pisavo izumili v starem Rimu na prehodu 7. v 6. stoletje pred našim štetjem
- najprej se je razvilo več vrst velikih črk ali majuskul
- zaradi večje hitrosti pisanja in različnih vrst pisal (dleto, čopič, trstika) se je v 3. stoletju iz velike pisave razvila nova mala pisava ali minuskula
- velike pisave so oblikovane v dvolinijskem sistemu, male pa v štirilinijskem
- z renesanso in iznajdbo tiska pa se je pričel razvoj pisav, ki jih uporabljamo še danes, ko imamo na voljo v digitalni obliki več kot 100.000 različnih vrst latinskih pisav

## Proporcionalne pisave in pisave z enako širino črk

- v proporcionalni pisavi imajo pri isti velikosti različne črke različno širino, ki je odvisna od oblike črke
- obstajajo pa pisave (angl. monospaced fonts), ki imajo vse črke enako široke

iiiiiiiiii

mmmmmmmmmm

iiiiiiiiiiii

mmmmmmmmmm



## Serifne in tehnične pisave

Latinico delimo na dva glavna črkovna sklopa:

- sklop pisav s tankimi in podebeljenimi potezami (angl. *serif fonts*)
- ter sklop pisav s skoraj enako ali z enako debelimi črtami (tehnični slog ali angl. *sans serif fonts*)

S S m m p p t t 9 9

## Prvi sklop pisav s serifi

delima na štiri skupine:

**I** skupina: beneške renesančne pisave (od 1470 do 1500),

**II** skupina: francoske renesančne pisave (od 1495 do 1757),

**III** skupina: baročne pisave (od 1757 do 1790),

**IV** skupina: klasicistične pisave (od 1790 od 1900).

Kasneje v 20. stoletju so se uporabljale vse skupine pisav s serifi in njihova uporaba ni bila omejena le na tisto časovno obdobje, v katerem so pisave prvotno nastale. To velja v še večji meri tudi danes, ko se s pomočjo nove računalniške tehnologije porajajo nove pisave ali se ponovno obujajo pisave, ki imajo značilnosti pisav izpred nekaj stoletij.

## Drugi sklop pisav tehničnega sloga

delimo na dve skupini:

**V** skupina: egipčanske pisave oziroma pisave z oglatimi serifi (angl. *square serif types*) (od 1815 dalje). V teh pisavah so serifi pravokotne črte.

**VI** skupina: linearne pisave (od 1819 dalje) (angl. *sans serif types*). Te pisave nimajo serifov.

## Obstajajo še štiri samostojne skupine črkovnih slogov

**VII** skupina: dekorativne oziroma akcidenčne pisave (votle, črtane, okrašene, osenčene, podvojene itd). Običajno so v takih pisavah le velike črke. Uporabljamo jih za vabila, poročna naznanila, vstopnice, oglase, plakate, naslove ipd.

**VIII** skupina: rokopisne pisave, ki so posnete po klasičnih *KALIGRAFSKIH PISAVAH*. Uporabljamo jih le za posamezne vrstice besedila ali za naslove.

**IX** skupina: risane pisave, ki so posnete po pisavah, napisanih s čopičem ali trsko.

**X** skupina: gotske pisave (tekstura, rotunda, schwabacher in fraktura). Prvi dve slovenski knjigi iz leta 1550, sta bili natisnjeni z gotskimi pisavami.

Trubar je kasneje raje uvedel renesančne pisave. Črki  $\mathcal{R}$  in  $\mathcal{I}$  uporabljamo za označevanje množic realnih in imaginarnih števil.

## Katere vrste črk so lažje berljive?

- Izkušnje in raziskave o hitrosti in berljivosti natisnjenih besedil kažejo, da je pisave s serifi možno hitreje in lažje brati.
- Zato za daljša besedila običajno uporabljamo serifne pisave, za bolj poudarjene dele besedil, kot so naslovi in mednaslovi, pa se dobro obnesejo neserifne pisave.

## Družine pisav

- Črke ene pisave so navadno izdelane v več različicah, ki sestavljajo družino te pisave.
- Družino pisav sestavljajo pisave, ki so med seboj oblikovno usklajene, ločijo pa se po velikosti, teži, širini in obliki.
- V družini pisav so običajno pokončna pisava (angl. *roman*), *kurzivna pisava* (angl. *italic*), **krepka pisava** (angl. *bold*) in MALE KAPITELKE (angl. *small caps*) v nekaj standardnih velikostih.

## Kurzivna pisava

- V kurzivni pisavi so črke nagnjene v desno in v pravi kurzivni pisavi je oblika posameznih črk običajno tudi nekoliko drugačna od pokončnih črk.
- V tako imenovani elektronski kurzivi pa so običajne pokončne črke *le nagnjene v desno*.

A B C a b c č e g @

*A B C a b c č e g @*

*A B C a b c č e g @*



## Velikost pisav

- tipografska enota za velikost črk je pika (angl. *point*), ki meri  $1/72$  palca ali  $0,352778\text{mm}$  in jo je vpeljal Adobov jezik Postscript [?].
- Tipične velikosti črk so 7, 8, 9, 10, 11, 12, 14, 16, 18, 20, 24, 30 in 36 pik
- v klasičnem tiskarstvu poseben nabor črk za vsako velikost
- na računalniku pa lahko večamo ali manjšamo črke
- VENDAR poljubno pomanjšanje ali povečanje črk ne daje dobrih rezultatov, saj se podoba črk za človeško oko ne spreminja linearno. Črke, načrtovane za večjo velikost, so običajno ožje in imajo več podrobnosti kot črke, ki so bile načrtovane za manjšo velikost, katerih poteze so širše.

Pisava velikosti 12 pik in pisava velikosti 6 pik, povečana na velikost 12 pik.

## Rastrske črke in obrisne črke

- Črke prvih pisav, ki so se začele uporabljati na računalnikih, so bile sestavljene iz pik, ker je tako narekovala tehnologija
- kasneje se je pojavil nov način definiranja oblik črk, in sicer z matematičnim zapisom krivulj, s katerimi so definirani obrisi črk (angl. outline fonts).
- Formati: PostScript podjetja Adobe, TrueType, OpenType

## Osnovna izhodišča za oblikovanje besedil

- *dokument moramo oblikovati glede na njegov namen in funkcijo!*
- uporabimo standardne vzorce za osnovne vrste dokumentov, ki so pogosto na voljo v posameznih programskih orodjih
- Uporabljajmo standardne nastavitve širine robov, razmika med vrsticami, razmika med črkami (spiranje ali angl. *kerning*).
- Glede velikosti črk in širine vrstic moramo paziti, da bo v vrsticah deset do dvanajst besed oziroma največ okoli 60 znakov.

## Standardna struktura strokovnih besedil

1. **Naslov** dela, imena avtorjev, institucija in njen naslov, naslovi elektronske pošte avtorjev, datum in kraj nastanka dela.
2. **Povzetek**, v katerem v sto do dvesto besedah opišemo vsebino celotnega dela (v slovenščini in tujem jeziku).
3. **Uvod**, s katerim uvedemo bralca v delo in podamo pregled celotnega dela.
4. **Pregled** področja, s katerim se delo ukvarja, in sorodnih rešitev, če predlagamo v delu neko novo rešitev. Če gre za kratko besedilo, sta uvod in pregled lahko združena.
5. **Glavni del**, ki je običajno sestavljen iz večih poglavij (npr. zajem podatkov, uporabljene metode in tehnologija, opis lastnih rešitev, rezultati).

6. **Zaključek**, kjer povzamemo glavne rezultate naloge.
7. **Zahvala**, v kateri navedemo imena vseh, ki so nam pri delu pomagali, oziroma so delo omogočili. To so lahko fizične osebe ali tudi institucije, ki so financirale naše delo.
8. **Seznam** virov oziroma literature, na katero se sklicujemo v svojem delu.
9. **Dodatki** (npr. algoritmi, daljše matematične izpeljave itd.).

## Prednosti vizualnega urejanja besedil

- Orodja za vizualno urejanje besedil je lažje uporabljati in se jih uporabniki hitreje naučijo.
- Z vizualnimi orodji je lažje izvajati zahtevno grafično oblikovanje.
- Primerna so predvsem za kratka besedila, za dolga besedila pa kmalu postanejo preokorna.
- WYSIWYG pravzaprav pomeni “What you see is ONLY what you get”.

## Prednosti logičnega urejanja besedil

- Logično urejanje zaradi ločitve vsebine (logične strukture besedila), od oblike omogoča konsistentno oblikovanje celotnega besedila na osnovi njegove logične strukture.
- Logično strukturirana besedila lahko prevedemo iz ene strukturirane oblike v drugo strukturirano obliko (npr.  $\text{T}_{\text{E}}\text{X}$  v HTML ali obratno), ali pa jih ustvarimo z drugimi računalniškimi orodji (npr. enačbe v formatu  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  s programom Mathematica).
- Dosežemo lahko veliko višjo in konsistentno tipografsko kvaliteto.
- Lažje prenosljive in veliko manjše datoteke (ASCII).

## Primer fleksibilnosti logičnega urejanja besedil

- Fleksibilnosti logičnega urejanja besedil omogoča ločitev strukture in oblike (stilske datoteke).
- V izvornem besedilu v formatu  $\text{\LaTeX}$  bomo pisali notranji produkt na naslednji način:  $\text{\np}\{A\}\{B\}$ .

- Z definicijo makro ukaza  $\text{\np}$  za notranji produkt:

```
 $\text{\newcommand}\{2\}\{\text{\np}\}\{(\#1,\#2)\}$ 
```

notranji produkt  $A$  in  $B$  izpišemo kot  $(A, B)$ .

- Le z ustrezno spremembo makro ukaza pa lahko notranji produkt povsod v besedilu izpišemo tudi na druge načine:

$(A, B)$ ,  $(A|B)$  ali  $\langle A|B \rangle$ .



## Preglednice

- Preglednice so v osnovi elektronska oblika računovodske glavne knjige.
- Računovodske glavne knjige, oblikovane v vrstice in stolpce, omogočajo pregleden vnos finančnih podatkov, ki so omogočali medsebojno kontrolo vsot po vrsticah in stolpcih
- V preglednice v elektronski obliki lahko vnašamo ne le podatke, pač pa tudi formule, s katerimi izračunamo (obdelamo) vnešene podatke.
- Ob spremembi podatkov v celicah, ki jih obsegajo posamezne formule, se samodejno izračunajo in prikažejo novi rezultati.
- današnje preglednice omogoča celovito urejanje, manipulacijo, analizo in grafičen prikaz podatkov.

## Razvoj preglednic

- Prva elektronska oblika preglednice *VisiCalc*, ki sta jo razvila Dan Bricklin in Bob Frankston, predstavlja prvo resnično uspešno aplikacijo za osebne računalnike.
- *VisiCalc* je pripomogel, da so se mnogi ljudje, predvsem poslovneži, odločili za prvi nakup osebnega računalnika.
- Tudi naslednik *Lotus-1-2-3*, ki ga je v 80-ih letih razvil Mitch Kapor, ostaja eden najbolj prodajanih produktov programske opreme vseh časov.
- Naslednji mejnik v razvoju preglednic predstavlja Microsoftovo orodje *Excel*, prvotno napisan za 512K Apple Macintosh v letih 1984–1985.

## Računalniška grafika

- Računalniška grafika obsega ustvarjanje raznovrstnih vizualnih informacij s pomočjo računalniške tehnologije.
- S pomočjo računalniške grafike je danes možno sintetizirati realistične upodobitve bodisi realnih ali povsem namišljenih predmetov in okolij.
- Ker ljudje zlahka in zelo hitro razumemo vizualne upodobitve, postaja računalniška grafika vse pomembnejše računalniško področje, ki se uporablja na vseh možnih področjih od uporabniških vmesnikov do skoraj vseh sodobnih uporabniških programov.

## Vrste računalniške grafike

- glede na vrsto slik in objektov ločimo **2D** in **3D** računalniško grafiko,
- glede na to, ali gre za ustvarjanje posameznih slik ali za dinamično zaporedje slik, kot ga potrebujemo za animacijo, pa ločimo **statično** in **dinamično** računalniško grafiko
- glede na vrsto uporabniške interakcije ločimo **interaktivne** programe in take, predvsem računsko zahtevne, ki za podane vhodne parametre izračunajo zahtevano sliko v času, ki ga merimo v minutah, za zelo zahtevne upodobitve pa celo v urah.
- glede na vlogo slike ločimo grafične programe, kjer je **slika končni cilj**, kot na primer pri grafičnem oblikovanju, in programe, kjer je **slika le del nekega produkcijskega procesa**, kot so na primer modeli CAD (Computer Aided Design).

## Področja uporabe računalniške grafike

- računalniški uporabniški vmesniki,
- znanstvena vizualizacija in simulacija,
- inženirsko načrtovanje (CAD/CAM),
- kartografija in geografski informacijski sistemi (GIS),
- medicina,
- grafično oblikovanje in umetnost,
- animacija,
- navidezna resničnost (angl. *virtual reality*).

## Vektorski in rasterski zasloni

- Glede na način generiranje slike na računalniškem monitorju ločimo vektorske in rastrske zaslone.
- Vektorski zasloni, ki so v preteklosti prvi omogočali grafičen prikaz, oblikujejo sliko na zaslonu iz črt poljubne dolžine in naklona s pomočjo neposrednega krmiljenja elektronskega žarka v katodni cevi.
- Na rastrskem zaslonu je slika podobno kot na televizijskem ekranu sestavljena iz polja slikovnih pik (angl. *pixel*).
- Kvaliteta rastrske slike je odvisna od razsežnosti slikovnega polja (na primer  $512 \times 512$ ,  $1024 \times 1024$ ) in globine zapisa posameznega polja (ponavadi 8-bitni za črnobeke zaslone in 24-bitni za barvne, po 8 bitov za rdečo, zeleno in modro barvo).

## Vektorski in rasterski zapis

- Rastrska računalniška grafika (angl. *bitmap graphics*) je sestavljena iz pik, organiziranih v obliki 2D polja določene dimenzije.
- Če tako sliko ali njen izsek povečamo, postanejo posamezne pike vidne in zato postanejo objekti na sliki zrnati.
- Pri vektorski računalniški grafiki pa so posamezni elementi definirani v matematični obliki s pomočjo vnaprej določenih geometrijskih objektov (daljice, krogi, elipse, pravokotniki, fonti, Bézierjeve krivulje ipd.) in so zato neodvisni od povečave.
- Vektorsko zapisane grafične objekte sicer tudi prikazujemo na rastrskih zaslonih, vendar se ti objekti vsakokrat prilagodijo najvišji ločljivosti izhodne naprave, bodisi računalniškega zaslona ali tiskalnika.

## Glavni koncepti računalniške grafike

- **Uporabniški grafični program**, ki omogoča konstrukcijo ter spreminjanje in manipulacijo z grafičnimi objekti.
- **Grafični sistem**, ki igra funkcijo “kamere” in ki določa, kaj se vidi in kako se vidijo objekti na grafičnem zaslonu.
- **Grafična podatkovna struktura**, ki predstavlja posamezne grafične elemente, ki sestavljajo na primer neko sliko, in njihove lastnosti (oblika, velikost, barva ipd.) ter njihova medsebojna razmerja (položaj, orientacija).



## Grafični formati

- V računalniški grafiki poznamo celo vrsto standardnih grafičnih zapisov: png, gif, tiff, pict, jpeg, ps, eps, pdf za 2D podatke, vrml za 3D podatke, mpeg in avi za video posnetke,
- Med grafičnimi formati velja posebej omeniti PostScript, saj je to pravzaprav programski jezik, ki je posebej namenjen vektorskemu opisu grafičnih elementov.  
Osnovna struktura jezika PostScript je sklad (angl. *stack*).

## Skupine uporabniških grafičnih programov

- **Barvanje in obdelava fotografij.** Ti programi uporabljajo rastrski zapis (na primer Adobe Photoshop in GIMP).  
Tipične operacije: razne vrste filtriranja, kot so glajenje, izostritev in iskanje robov, ter spreminjanje barv in razne vrste preobrazb.
- **Risanje.** Ti programi običajno uporabljajo vektorski zapis (na primer Adobe Illustrator in OpenOffice.org Draw).  
Tipične operacije: premik, rotacija, skaliranje, zrcaljenje, barvanje, senčenje ipd.
- **Načrtovanje.** Ti programi so bodisi splošni 2D ali 3D načrtovalski programi, kot je na primer AutoCAD, ali pa so namenjeni določenemu uporabniškemu področju, kot na primer program ArchiCad za arhitekturo.

- **Izdelava grafov in diagramov.** Številni programi na osnovi numeričnih podatkov izdelajo različne vrste diagramov (točkovne, linijske, stolpične, tortne, polarne, kombinirane ipd.).  
To so lahko samostojni programi ali pa funkcije v sklopu drugih programov, kot so na primer OpenOffice.org Calc, Excel, Mathematica itd.
- **Izdelava predstavitev.** Za izdelavo prosojnic in računalniških predstavitev poznamo celo vrsto programov (na primer Microsoft PowerPoint in OpenOffice.org Impress).
- **3D animacija,** ki postaja za filmsko industrijo vse bolj pomembna. Tipični programi za 3D animacijo so Maya, Lightwave in 3D Studio.
- **Obdelava videa.** Filmska in video tehnika se digitalizira. Na močnejšem osebнем računalniku se načeloma že da obdelati in zmontirati celovečerni film.

## Govorne predstavitve

- v poslovnem in akademskem svetu moramo pogosto predstaviti rezultate svojega dela
- na predstavitev se moramo dobro pripraviti
- upoštevati moramo predvsem, kdo nas bo poslušal in koliko časa imamo na voljo za predstavitev
- za svoj nastop moramo pripraviti ustrezna gradiva in svoj nastop vaditi
- spontan nastop zahteva veliko vaje!

## Gradiva za govorno predstavitev

- gradiva za udeležence
- gradiva za govornika
- projekcijska gradiva

## Gradiva za udeležence

- gradiva, ki jih razdelimo udeležencem že pred ali med samim nastopom
- vabilo (in povzetek)
- izčrpna gradiva (članki, konferenčni zbornik)
- gradiva v elektronski obliki (CD rom, internet)

## **Gradiva za govornika:**

- namenjena nastopajočemu bodisi pri pripravi na nastop ali za sam nastop
- besedilo referata za branje
- skript – besedilo z dodatnimi navodili za podajanje

## Brati ali ne brati?

- odvisno od vrste dogodka
- nastop je bolj spontan in neposreden, če prosto govorimo in ne beremo
- za pravilni vrstni red podajanja in kot pomoč pri našem podajanju si pomagamo s projekcijskimi gradivi



## Projekcijska gradiva

- tabla in kreda
- bela tabla in barvni svinčniki
- grafoskop in folije (napisane z roko ali oblikovane na računalniku)
- diaproyektor in 36mm ali  $6 \times 6$ cm diapozitivi
- računalnik in video projektor
- posebni programi za oblikovanje prosojnic

## Kako pripraviti projekcijska gradiva?

- vsebinski izbor tega, kar nameravamo povedati
- daljši dokument (npr. 40 strani) → 10 minut časa za predstavitev?
- ena minuta govora → ena prosojnica
- na eno prosojnico:  
kolikor lahko napišemo s tiskanimi črkami na listek  $8 \times 8$  cm
- čimveč informacij podamo na grafičen način

## Oblikovanje prosojnic

- temna pisava na svetlem ozadju ali svetla pisava na temnem ozadju?
- ozadje naj bo uniformno!
- črke dovolj velike, da lahko berejo poslušalci iz zadnje vrste
- ne pretiravajmo z efekti (animacije, zvok, video), da ne zasenčijo govornika
- že pripravljene oblikovne predloge ali oblikovanje lastnih?

## Orodja za izdelavo prosojnic

- posebni programi za izdelavo multimedijskih predstavitev
- spletna predstavitev
- PDF dokument za največjo možno neodvisnost od platforme
- PDF $\LaTeX$  besedilo oblikovano v sistemu  $\LaTeX$  pretvori v PDF, stilna datoteka `pdfslide.sty` pa omogoči dodatne ukaze za izdelavo interaktivne predstavitve.

## Matematična orodja

- orodja za reševanje matematičnih problemov: *Mathematica*, *Matlab*, *Maple*, *Octave*, *Derive*
- Matlab: predvsem za numerično reševanje problemov
- Mathematica: simbolično reševanje problemov
- interaktivno delo
- programiranje novih funkcij
- grafična predstavitev rezultatov
- podpora različnih platform

## Uporabnost matematičnih orodij

- učenje matematike (koncepti, primerjalne študije),
- učenje in raziskave na področju tehnike, računalništva, fizike, avtomatike, ekonomije, kemije, biologije itd.,
- raziskave in razvoj v industriji.

## Mathematica

- Mathematica je učinkovit, morda celo revolucionaren programski paket oziroma jezik, ki omogoča integracijo simboličnih, numeričnih in grafičnih funkcij
- omogoča ustvarjanje dokumentov, ki so na las podobni besedilom (knjige, seminarske naloge)
- vendar ti dokumenti omogočajo dinamično prevajanje, saj so v osnovi sestavljeni iz živih enačb in animacij
- Žive enačbe so celice dokumenta, ki jih Mathematica ovrednoti in izpiše rezultat pod ustrezno celico.
- Mathematica je plačljiv program, zato ga žal na zgoščenki Slix ni.

## Medmrežje in svetovni splet

- Internet ali medmrežje je nastalo v 60-ih letih 20. stoletja, kot Arpanet
- povezoval je le nekaj ameriških univerz in raziskovalnih inštitutov
- protokol TCP/IP, ki omogoča delovanje omrežja brez centralnega nadzora
- delovanje brez centralnega nadzora je bila pomembna funkcionalna zahteva njegovega naročnika, ameriškega obrambnega ministrstva
- Internet je že od samega začetka omogočal izmenjavo elektronske pošte in delo na oddaljenih računalnikih (protokol telnet)
- štejejo ga za najpomembnejši in najbolj presenetljiv izum 20. stoletja



## WWW ali svetovni splet

- s pojavom svetovnega spleta je internet stopil iz znanstvenih in strokovnih krogov v širšo javnost in se bliskovito razširil
- s preprostim grafičnim vmesnikom, ki ga imajo današnji spletni brskalniki, lahko vsakdo sam išče po internetu raznovrstne informacije
- jedro spletnih strani predstavljajo dokumenti, napisani v jeziku HTML
- HTML omogoča povezovanje med točkami v besedilu z drugimi besedili, slikami, zvokom ali video sekvencami, ki so lahko na istem računalniku ali pa kjerkoli na svetovnem spletu
- ta princip povezljivosti se imenuje **hipertekst**

## Hipertekst

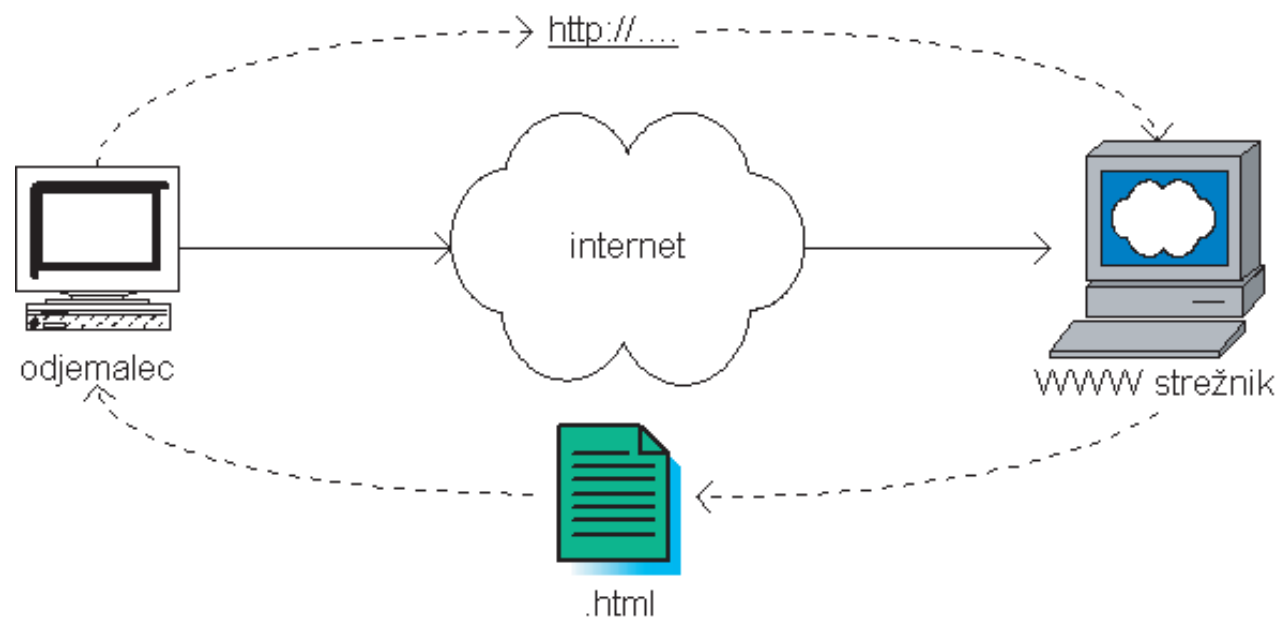
- pojem hiperteksta je definiral Ted Nelson leta 1965
- o konceptu hipertekstovnih povezav je v svojem projektu Memex govoril že Vannevar Bush leta 1945
- prva uspešna izvedba hipertekstovnega sistema je bil Applov Hypercard leta 1987
- jezik HTML, ki ga je leta 1991 razvil Tim Berners-Lee v jedrskem raziskovalnem centru CERN v Ženevi, je možnost povezav med točkami v besedilih, slikah, zvočnih in video zapisih, in to neodvisno od tega, kje se ti zapisi nahajajo, ponesel na internet

## Uspeh WWW

- poleg interaktivnosti in grafičnega uporabniškega vmesnika pa je k uspehu svetovnega spleta pripomogla tudi neodvisnost od strojne in systemske računalniške opreme
- za uporabo svetovnega spleta ni pomembna ne vrsta računalnika, tip procesorja ali operacijskega sistema
- dovolj je le to, da ima računalnik dostop do interneta in da je na računalniku nameščen spletni brskalnik
- povezave: telefon (ADSL), televizijski kabel, brezžične povezave
- svetovni splet sestavljajo vsi dokumenti, ki so dosegljivi preko internetnih protokolov in so medsebojno povezani preko imenovanih povezav (URL) v dokumentih

## Kako deluje svetovni splet? I

- iz uporabniškega vidika si lahko predstavljamo delovanje spleta kot interakcijo med odjemalci in strežniki, ki poteka prek internetnih povezav



## Kako deluje svetovni splet? II

- odjemalec je spletni brskalnik, ki teče na računalniku uporabnika
- brskalnik iz naslova v obliki URL pridobi naslov strežnika in iz njega internetni naslov
- potem vzpostavi internetno povezavo do strežnika in pošlje zahtevek, ki vsebuje URL naslov zahtevanega dokumenta in odvisno od uporabljenega protokola še nekaj dodatnih informacij.
- strežnik prebere zahtevek odjemalca in odgovori z glavo in vsebino dokumenta
- odjemalec nato interpretira vsebino dokumenta in jo prikaže na zaslonu
- poleg protokola HTTP na spletu pogosto srečamo tudi varnejšo različico HTTPS, ter protokol za prenos datotek FTP

## Izdelava spletnih strani

- poleg tehničnega znanja, je pomembna predvsem vsebina spletnih strani (izbor besedil ter grafične in zvočne opreme)
- vsebina mora biti ustrezno strukturirana za medij medmrežja, grafika in zvok pa morata biti optimizirana za hiter prenos po računalniškem omrežju
- vse skupaj mora biti ustrezno oblikovano in povezano v tako strukturo, po kateri se je možno hitro premikati in poiskati informacije
- Izdelava spletnih strani zahteva dobro koordinacijo dela, saj so potrebni različni strokovnjaki
- spletne strani je potrebno tudi redno vzdrževati

## Potek izdelave spletne strani

1. Določitev ciljnih uporabnikov spletne strani, to so lahko:
  - spletni deskarji in naključni uporabniki ali ciljna publika (naše stranke, kupci itd.),
  - začetniki ali izkušeni uporabniki.
2. Kaj je osnovni namen spletne strani (na primer informiranje, prodaja, zabava ipd)?
3. Na osnovi namena oblikovanje konkretnih ciljev (na primer podpora strankam, informiranje zaposlenih ipd).
4. Določitev obsega informacij na spletni strani (koliko besedila, koliko slik).
5. Kakšen bo osnovni način komuniciranja z obiskovalci spletne strani (tekstovni, vizualni)?

6. Na osnovi obsega vsega gradiva in načina komuniciranja se določi strukturo spletne strani.
7. Na osnovi strukture spletne strani se določi primeren način navigacije po spletni strani.
8. Sledi grafično oblikovanje spletne strani.
9. Izvedba spletne strani.
10. Vzdrževanje spletne strani.



## Pisanje za svetovni splet

- branje z zaslona je počasnejše kot branje s papirja zaradi višjega kontrasta tiska na papirju pri različnih svetlobnih pogojih in zaradi večje fleksibilnosti papirnatega medija
- pri branju informacij na računalniku bralci računalniški zaslon najprej preletijo podobno kot pri branju časopisa, šele nato začno z branjem
- težišče informacije, podane preko zaslona, mora biti zato v vizualni organizaciji strani in v slikovnih informacijah
- tipični uporabnik posveti svojo pozornost eni spletni strani v povprečju le okoli osem sekund
- zato je potrebno najvažnejše informacije podati hitro in na najbolj vidnem mestu (na vrhu strani, z veliko in krepko pisavo)!

- tudi v globino spletne strani se tipični uporabnik sprehodi le do dva pritiska na miško daleč
- informacije morajo biti zato strukturirane v majhne pakete, da posamezne strani niso predolge in da ni potrebno predolgo listanje

## Izražanje na svetovnem spletu

- pisati moramo natančno in jasno, tako da uporabljamo kratke stavke ter jasne in aktivne glagole
- besedilo poskušamo jasno strukturirati s pomočjo spiskov ali tabel
- ne uporabljajmo žargona, fraz in kratic, ki niso razumljive širši javnosti
- zavedati se moramo, da je za svetovni splet javnost lahko kar cel svet
- vnaprej si postavimo splošne omejitve, kot so na primer največje število pritiskov na miško do določene vrste informacij, povprečna dolžina besedila o določeni vrsti vsebine, dolžina odstavkov (na primer 5 vrstic)

## Slikovna oprema spletnih strani

- zaradi neposrednosti spletnega medija je uporaba slikovnih informacij zelo zaželena
- numerične vrednosti lahko predstavimo kot:
  - kolač kadar gre za dele celote,
  - histogram če želimo predstaviti razmerja med vrednostmi.
- kvalitativne informacije lahko predstavimo v obliki
  - urnika ali tabele,
  - potek dogajanja kot diagram s puščicami,
  - organizacijo ali hierarhijo z drevesnim diagramom.
- s pomočjo tabel in spiskov je možno razčleniti in nadgraditi besedilo

- s pomočjo puščic, vodorovnih in navpičnih črt in različnih barv ozadja pa je možno hitreje voditi pogled bralca med posameznimi informacijskimi enotami
- grafična oprema naj ne le polepša, temveč predvsem izboljša informacijo
- upoštevati moramo tudi čas nalaganja in barvno ločljivost slik
- najbolj razširjeni grafični formati, ki se uporabljajo na spletu, so:
  - GIF (256 barv), ki je primeren za risbe in diagrame,
  - JPEG, ki omogoča 24-bitne barve in različne stopnje stiskanja ter je zato najbolj primeren za fotografije, ter
  - PNG, ki omogoča brezizgubno stiskanje slik s poljubno barvno globino in v zadnjem času tudi zaradi težav s patenti nadomešča format GIF.

## Komuniciranje z obiskovalci spletnih strani

- na spletni strani naj bi bile informacije podane v obliki dialoga, ki obiskovalca strani zvabi v interakcijo
- komunikacija naj bi bila intimna, s tem da se prilagodi uporabnikovim potrebam in pričakovanjem
- na obiskovalca se obračamo v drugi osebi in tako kot v pogovoru uporabljamo le kratke stavke ali celo le dele stavkov
- pri naslavljanju obiskovalcev ne smemo delati omejujočih predpostavk, saj ne vemo vnaprej, kdo so, koliko so stari, kakšno izobrazbo imajo, ali dobro razumejo jezik spletne strani

## Interakcija s spletnimi obiskovalci

- pri interaktivnem komuniciranju moramo biti vljudni ter upoštevati uporabnikove odgovore
- interakcija mora biti za uporabnika tehnično nezahtevna
- odgovore obiskovalcev lahko klasificiramo tako, da že vnaprej predvidimo najbolj značilne odgovore in tako grupiramo uporabnike
- vnaprej pripravljenimi odgovori na najbolj pogosta vprašanja (angl. *FAQ*)
- individualizirana obravnava uporabnikov (spletna trgovina si zapomni, kaj smo že prej kupili, katero številko nosimo, kakšen okus imamo itd.)





## Struktura in vrsta informacij

<i>struktura</i>	<i>čas interakcije</i>	
	<i>kratek</i>	<i>dolg</i>
linearna	urjenje	poučevanje
nelinearna	viri	izobraževanje

- za **urjenje in izpite** moramo omejiti gibanje na naprej - nazaj,
- za **poučevanje** so poleg povezave naprej pomembne tudi povezave v globino na podrobnejše razlage,
- za **izobraževanje** je potrebna večja fleksibilnost
- pri **virih** je pomembno predvsem hitro iskanje določenih informacij