

# Linguaggi e formati per la grafica vettoriale

- Esistono molti formati per i disegni in grafica vettoriale:
  - .ai – Adobe Illustrator
  - .eps – Encapsulated Postscript
  - .cdr – Corel Draw
  - .svg – Structure Vector Graphics
  - .cgm – Computer Graphics Metafile
  - .dxf – Autocad
  - .wmf – Windows Metafile
  - ...
- Esamineremo **Postscript** e **SVG**

# Il linguaggio Postscript

- **Postscript** è un vero e proprio linguaggio di programmazione, pensato per descrivere l'aspetto di pagine stampate
- Il risultato dell'esecuzione di un programma Postscript è un disegno
  - in senso lato: include grafica vettoriale, grafica raster, testo
  - il motore di rendering è detto **interprete Postscript**

# Postscript – struttura

- Un programma Postscript (ovvero, un'immagine vettoriale salvata in Postscript) è un file di testo
- Il simbolo “%” introduce commenti
- Il resto del programma descrive (“calcola”) il disegno
- Alla fine, il comando **showpage** stampa la pagina calcolata (e passa alla successiva)

# Postscript – pila

- La valutazione delle espressioni viene fatta su una **pila**
- Gli operandi (valori) vengono posti sulla pila
- Gli operatori prendono valori sulla pila e lasciano l'eventuale risultato sulla pila
- L'effetto è un po' “arabo” -- si legge da destra a sinistra...

# Postscript – pila

- Esempio: l'operatore **mul** moltiplica due numeri (operandi) dallo stack
  - 2 5 mul ▶ 10
  - 10 2 5 mul mul ▶ 100
- Per ogni operazione o comando di disegno, prima si calcolano gli operandi, poi si applica il comando

# Postscript – coordinate

- Le coordinate dei punti sono date in Postscript in unità di **1/72 di pollice**
- Il pollice è pari a 2,54 cm; l'unità (punto Postscript) è pari a 0,035277778 cm.
- L'origine (0,0) è nell'angolo in basso a sinistra della pagina
- Le misure sono **indipendenti** dai pixel di schermo e dalla risoluzione della stampante

# Postscript – path

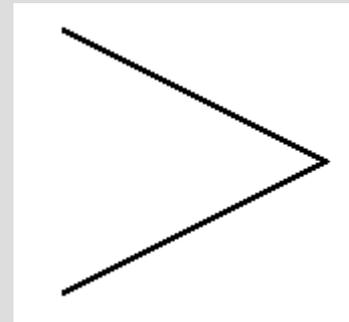
- I path in Postscript sono composti da una sequenza di punti
- Fra un punto e il successivo si può avere:
  - una linea (comando **lineto**)
  - una curva di Bezier cubica (comando **curveto**)
  - niente del tutto (comando **moveto**)
- Il comando **closepath** chiude un path

# Postscript – esempio

- Altri comandi consentono di definire le proprietà dei path
  - es.: **setlinewidth**
- Il comando **stroke** disegna il bordo del path, **fill** disegna l'area l'interna

```
% Esempio di path
```

```
newpath  
100 200 moveto  
200 250 lineto  
100 300 lineto  
2 setlinewidth  
stroke
```



# Postscript – esempio

```
% Esempio di path
```

```
newpath
```

```
100 200 moveto
```

```
200 250 lineto
```

```
100 300 lineto
```

```
closepath
```

```
gsave
```

```
0.5 setgray
```

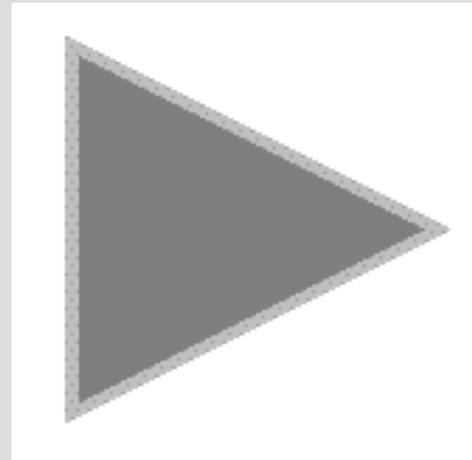
```
fill
```

```
grestore
```

```
4 setlinewidth
```

```
0.75 setgray
```

```
stroke
```



- I comandi **gsave** e **grestore** salvano e ripristinano, rispettivamente, lo stato corrente
- Funzionano come delle “parentesi” grafiche

# Postscript – testo e font

- Altri comandi consentono di definire il font, la dimensione, e altri attributi di un testo
- I testi sono delimitati da ( )
- **show** disegna un testo

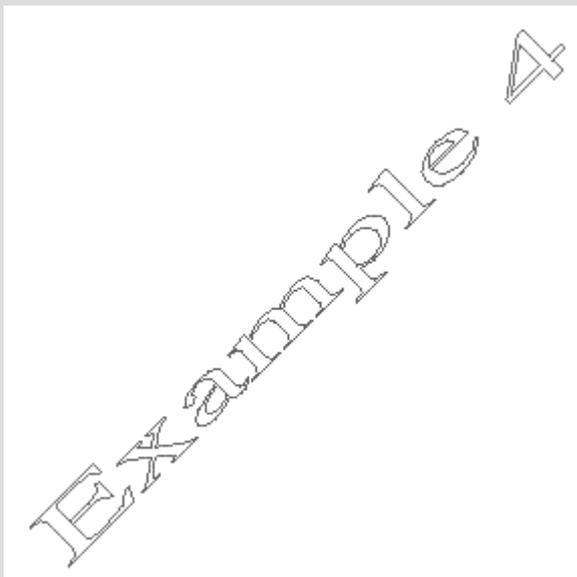
```
% Esempio di testo
```

```
/Times-Roman findfont  
12 scalefont  
setfont  
newpath  
100 200 moveto  
(Esempio di testo) show
```

Esempio di testo

# Postscript – trasformazioni

- I comandi **translate**, **scale** e **rotate** calcolano le matrici di trasformazione e le moltiplicano per la matrice corrente



```
/Times-Roman findfont
32 scalefont
setfont
100 200 translate
45 rotate
2 1 scale
newpath
0 0 moveto
(Example 4) true
charpath
0.5 setlinewidth
0.4 setgray
stroke
```

# Postscript – definizioni

- Il comando **def** consente di associare un pezzo di codice (ovvero, un disegno) a un nome
- I nomi iniziano per /
  - esattamente come abbiamo visto per **findfont**

```
/linea {  
    newpath  
    0 0 moveto  
    100 0 lineto  
    stroke  
} def
```

e poi:

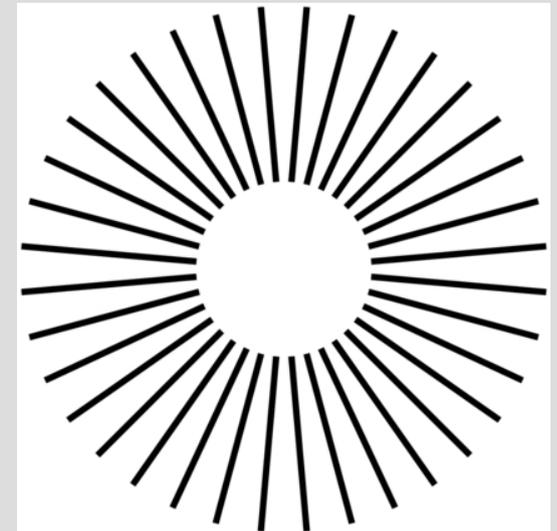
```
10 setlinewidth  
0 setlinecap  
0 setgray  
linea
```



# Postscript – costrutti di controllo

- Come tutti i linguaggi di programmazione, Postscript include un certo numero di **costrutti di controllo**
- Stessi principi che in altri linguaggi:
  - **if**, **for** (comando **repeat**), **while**
- Esempio:

```
gsave
36 { 10 rotate
      10 10 moveto
      30 30 lineto } repeat
stroke
grestore
```



# Postscript come formato di file

- Oltre che per la stampa, Postscript può essere utilizzato come **formato di memorizzazione** per immagini vettoriali **editabili**
- In genere, si usa un sottoinsieme ristretto del Postscript
  - il linguaggio intero è “troppo”!
- È (relativamente) facile interpretare un file Postscript e disegnarne il contenuto, ma è (molto) difficile **editare** questi file!

# Il formato di Adobe Illustrator

- I file di **Adobe Illustrator** (.ai), per esempio, sono file Postscript che usano un insieme di comandi particolari
  - procedure di Adobe, definite con **def**
  - Illustrator sa poi come interpretare questi comandi speciali, e li trasforma in documenti vettoriali editabili
  - Il programma non saprebbe editare file Postscript generici!

# Il formato EPS

- Il linguaggio Postscript è quello interpretato nativamente da quasi tutte le stampanti laser e le fotocomposizioni
- Postscript gestisce naturalmente documenti multi-pagina
- Per la memorizzazione di singole immagini, si usa spesso il formato **EPS** (Encapsulated Postscript)

# Il formato EPS

- Un file EPS consiste di un programma Postscript, con in più alcuni **commenti** dal formato speciale
- I commenti speciali di EPS iniziano per %%
  - si ricordi che i normali commenti Postscript iniziano con %, i commenti EPS sono quindi anche commenti Postscript
  - la prima riga del file contiene informazioni sulla versione di EPS usata
  - ci sono alcune (piccole) restrizioni sul contenuto della parte Postscript

# Il formato EPS

- Esempio:

```
%!PS-Adobe-3.0 EPSF-3.0
```

```
%%Creator: programma che ha creato il file
```

```
%%BoundingBox: 0 0 larghezza altezza
```

```
%%LanguageLevel: 2
```

```
%%Pages: 1
```

```
%%DocumentData: Clean7Bit
```

```
... codice Postscript ...
```

```
%%EOF
```

# Il formato EPS

- I file EPS *possono* contenere una preview (miniatura) in un formato raster, codificata come una immagine Postscript
  - Alcuni programmi, quando importano un file EPS, mostrano questa preview – o, se non c'è alcuna preview nel file, mostrano un'icona standard (“X” rossa)
  - Altri sono in grado di disegnare essi stessi il Postscript contenuto nel file
  - In ogni caso, al momento della stampa viene usata la versione Postscript

# Adobe DSC

- I file Postscript e EPS *possono* aderire alle **Document Structuring Conventions** (Convenzioni sulla struttura dei documenti) definite da Adobe
  - Anche DSC si basa commenti speciali introdotti da %% e sparsi all'interno del file Postscript
  - I commenti speciali dichiarano quali font sono usati, quali insiemi di procedure sono necessarie, dove inizia e dove finisce il codice di ciascuna pagina

# Adobe DSC

**%!PS-Adobe-3.0**

**%%Pages:** *numero di pag.  
altri commenti iniziali*

**%%EndComments**

**%%BeginProlog**

*definizioni iniziali*

**%%EndProlog**

**%%BeginSetup**

*impostazioni iniziali*

**%%EndSetup**

**%%Page:** *n. pagina*

**%%BeginPageSetup**

*impostazioni per questa pagina*

**%%EndPageSetup**

*codice Postscript della pagina*

**%%PageTrailer**

*codice di chiusura della pagina*

**%%Trailer**

*codice di chiusura del documento*

**%%EOF**

- Parte iniziale
- Ripetuto per ogni pagina
- Parte finale

## Spool Management

*(Spooling, Banner and Trailer Pages, and Print Logging)*

%%Creator: %%PageMedia:  
%%CreationDate: %%PageRequirements:  
%%DocumentMedia: %%Requirements:  
%%DocumentPrinterRequired: %%Routing:  
%%For: %%Title:

## Resource Management

*(Resource Inclusion, Downloading, and Optimization)*

%%DocumentNeededResources: %%IncludeResource:  
%%DocumentSuppliedResources: %%Begin(End)Resource:  
%%PageResources:

## Error Management

*(Error Reporting and Recovery)*

%%Extensions: %%ProofMode:  
%%LanguageLevel:

## Printer Management

*(Printer Rerouting, Feature Inclusion, Parallel Printing, Color Breakout)*

%%Begin(End)Feature: %%IncludeFeature:  
%%Begin(End)Resource: %%IncludeResource:  
%%DocumentMedia: %%LanguageLevel:  
%%DocumentNeededResources: %%PageMedia:  
%%DocumentPrinterRequired: %%PageRequirements:  
%%DocumentSuppliedResources: %%PageResources:  
%%Extensions: %%Requirements:

## Page Management

*(Page Reversal, N-up Printing, Range Printing, Collation, Underlays)*

%%Pages: %%Page:  
%%EndComments %%Begin(End)PageSetup  
%%Begin(End)Setup %%PageTrailer  
%%Begin(End)Prolog %%Trailer

# Adobe DSC

- 
- Un gruppo di direttive, quelle relative alla risorse, consentono al documento di dichiarare di quali **risorse** ha bisogno:
  - font
  - file esterni (immagini)
  - librerie di procedure

Se compare (**atend**), i dati sono forniti nel Trailer (in fondo al file)

Si può andare a capo con %%+ (simbolo “segue”)

# Adobe DSC

- Spesso è utile guardare direttamente nel file Postscript quali font o altri file sono necessari
  - Per esempio, in caso di immagini o documenti che non stampano correttamente

```
%%DocumentNeededResources: file /home/gervasi/file.eps
%%+ file (Logo \042ganzo\042.eps)
/usr/share/clipart/pen.tiff
%%+ file C:\LIB\LOGO.EPS
```

```
%%DocumentNeededResources: font Times-Roman Palatino-Bold
%%+ font Helvetica Helvetica-Bold NewCenturySchoolbook-
Italic
```

# Portable Document Format

- La diffusione del Postscript a metà anni '80 ha **rivoluzionato** tutto il mondo dell'editoria elettronica
- Formato vettoriale: precisione infinita, ottima qualità, largamente indipendente da una particolare periferica (stampante)
- Tuttavia, la dipendenza da risorse esterne – soprattutto da font esterni – ha reso il formato poco portabile

# Portable Document Format

- Il formato **PDF** (Portable Document Format) è una variante di Postscript ottimizzata per l'interscambio di documenti
  - i file PDF contengono al loro interno i font utilizzati, i dati delle immagini, ecc.
  - i dati binari (inclusi i font) vengono compressi con un algoritmo (lossless) in modo da ridurre la dimensione dei file
  - il tutto è poi incapsulato in ASCII per non avere problemi di encoding



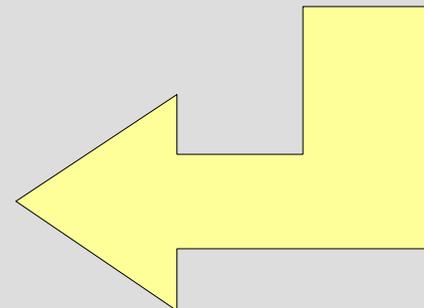
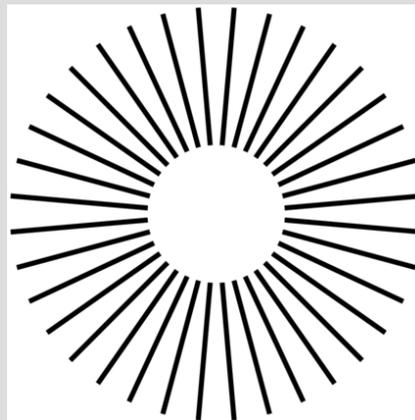
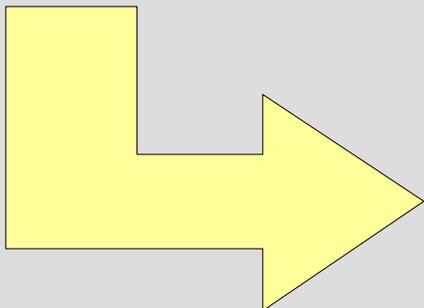
# Portable Document Format

- I file PDF contengono **le sole istruzioni di disegno** processate da un programma Postscript
  - Non ci sono procedure definite, né costrutti di controllo, niente variabili o altri calcoli...
  - In pratica, viene eseguito il programma Postscript, e ogni comando di “stampa” (**stroke**, **show**, ...) viene copiato, con i suoi parametri, nel PDF
- Meno generale di Postscript, ma più pratico!

# Portable Document Format

- Esempio: Postscript

```
/segm { 10 rotate
        10 10 moveto
        30 30 lineto
} def
gsave
36 segm repeat
stroke
grestore
```



- Esempio: PDF

```
gsave 10 rotate 10 10
moveto 30 30 lineto 10
rotate 10 10 moveto 30 30
lineto 10 rotate 10 10
moveto 30 30 lineto ...
altre 32 volte ... 10
rotate 10 10 moveto 30 30
lineto stroke grestore
```

# Portable Document Format

- Tipicamente, un file PDF comincia con un'intestazione ASCII, seguita vari blocchi (*oggetti*) composti da alcuni comandi e, opzionalmente, da dati binari compressi
  - l'operatore **stream** consente di incorporare **dati binari**
  - Questi ultimi, una volta decompressi, diventano ulteriori comandi di disegno PS
- Alla fine del file, ulteriori comandi compongono il disegno usando gli oggetti definiti in precedenza

# Portable Document Format

```
%PDF-1.4
%Ã€Ãƒ€Ã¶Ã¶.
2 0 obj
<< /Length 3 0 R
  /Filter /FlateDecode
>>
stream
x.iX[.^]7^L~_8ÿÃ.ÎZÿ^O,^K9.m
...
.fz^Ly>^T^X0t^o^Uÿ}^o^#.#.üQü^CÜ^K"üendstream
endobj

42 0 obj
<< /Length 43 0 R
  /Filter /FlateDecode
  /Length1 17632
>>
stream
x.Y|y|T*xœø9çb;^|*îw^XAt.R.æ%œ4K.6K_^ZûÒ_.ø6ë¯5l.ãõ^Y'
...
ZûÒ_.ø6ë¯5
endstream
endobj

44 0 obj
<< /Type /FontDescriptor
  /FontName /BAAAAA+Arial-Black
  /Flags 4
  /FontBBox [ -193 -306 1686 1083 ]
  /ItalicAngle 0
  /Ascent 1100
  /Descent -309
  /CapHeight 1083
  /StemV 80
  /FontFile2 42 0 R
>>
endobj ecc. ecc.
```

- L'intestazione `%PDF-versione` identifica un file PDF
- Gli **oggetti** possono essere immagini, font, comandi compressi

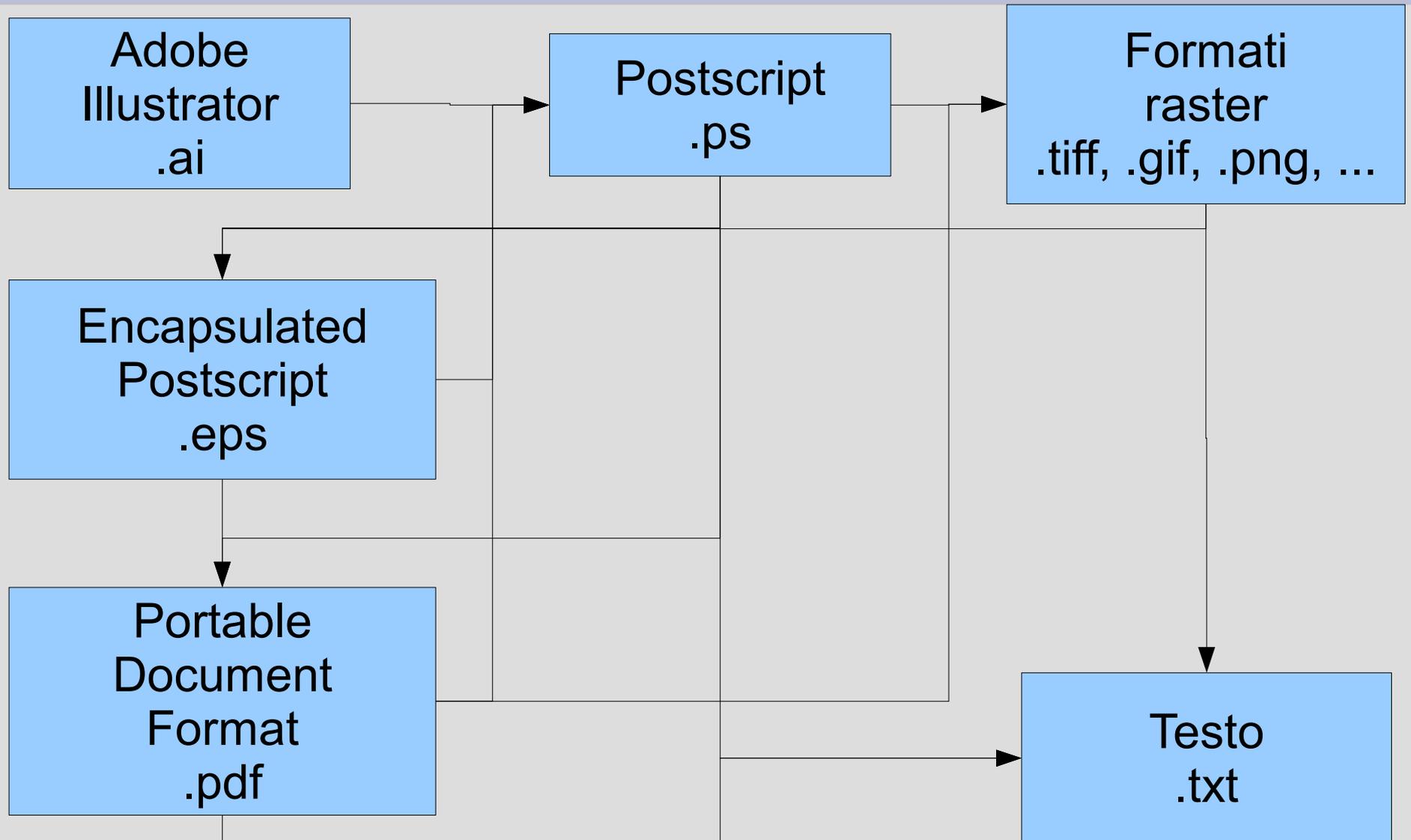
# Portable Document Format

- È possibile **recuperare** il testo o la grafica da un file PDF
  - caratteristica utilissima!
- PDF dispone anche di alcune impostazioni per il **controllo dei diritti** di accesso al documento
  - È possibile “vietare” la copia, la stampa, la modifica, ecc.
    - Il programma di rendering può però ignorare questi divieti... si tratta di una protezione assai debole

# Portable Document Format

- Infine, versioni recenti di PDF incorporano una serie di altre caratteristiche:
  - collegamenti ipertestuali sul web
  - FORM, con possibilità di compilarle e spedire i dati via Internet
  - supporto alla grafica 3D
  - supporto ai libri digitali (“eBook”)
  - supporto Javascript / ActionScript
  - ricerche in collezioni di documenti
  - ecc.

# Conversioni di formato



# Conversioni di formato

- .ai e .eps sono già .ps
- **Adobe Acrobat** o **Ghostscript** possono trasformare un file .pdf in uno .ps
- **Adobe Acrobat** o **pdftotext** possono trasformare un file .pdf in uno .txt (estraendo il testo, con errori)
- **ps2txt** può trasformare un file .ps in uno .txt (estraendo il testo, con errori)
- **Adobe Distiller** o **ps2pdf** possono trasformare un file .ps in uno .pdf
- **Ghostscript** può trasformare un file .pdf o .ps in un'immagine in grafica raster
- I programmi di **Optical Character Recognition** (OCR) possono estrarre il testo (con errori) da immagini raster
- I programmi di **tracing** (incluso GIMP e Photoshop) possono convertire un'immagine raster in una vettoriale (di solito .eps)
- Altre combinazioni sono fatte importando file di vario tipo in programmi di grafica che poi producono .ps o .pdf

# Esercizio

- Scegliete, create o scaricate da Internet un qualunque file .eps e uno .pdf
- Con un editor di testi, aprite entrambi i file e scorretene il contenuto
- Con Adobe Acrobat (Reader), aprite il PDF e leggetene le Proprietà (menu File) – ci saranno varie schede di dati
- Riuscite a trovare le proprietà in questione nel file PDF “crudo”? Se no, da dove pensate che le legga Acrobat?

# Riferimenti

- Sulla parte teorica il testo di riferimento è J.D. Foley, A. van Dam, et al., *Introduction to Computer Graphics*, Addison Wesley Publishing Company. Le trasformazioni sono nel Capitolo 5, le curve parametriche nel Capitolo 9.
- Le matrici di trasformazione si usano molto in ambito grafico: per esempio, la pagina [http://www.adobe.com/it/devnet/flash/articles/image\\_api\\_03.html](http://www.adobe.com/it/devnet/flash/articles/image_api_03.html) discute il loro uso in Flash 8 (per i disegni e i filmati)
- Un tutorial basico su Postscript si trova a <http://a2.pluto.it/a2402.htm>; una versione più completa è [http://a2.pluto.it/postscript\\_un\\_linguaggio\\_per\\_la\\_composizione\\_finale.htm](http://a2.pluto.it/postscript_un_linguaggio_per_la_composizione_finale.htm)
- L'interprete Postscript interattivo più diffuso è Ghostscript; la sua home page è <http://www.cs.wisc.edu/~ghost/>
  - con Ghostscript è possibile scrivere semplici programmi Postscript e vederne il risultato senza dover mandare il disegno su una stampante
- Si può scaricare Adobe Reader, nella versione più aggiornata, dalla pagina <http://www.adobe.com/products/acrobat/readstep2.html>
- Le specifiche ufficiali dei vari linguaggi e formati che abbiamo visto sono:
  - [http://partners.adobe.com/public/developer/ps/index\\_specs.html](http://partners.adobe.com/public/developer/ps/index_specs.html) (PS e EPS)
  - [http://partners.adobe.com/public/developer/en/ps/5001.DSC\\_Spec.pdf](http://partners.adobe.com/public/developer/en/ps/5001.DSC_Spec.pdf) (DSC)
  - [http://partners.adobe.com/public/developer/pdf/index\\_reference.html](http://partners.adobe.com/public/developer/pdf/index_reference.html) (PDF)