

Práctica 4. El lenguaje Postscript

Duración: 2 sesiones

1. Objetivos

Los objetivos concretos de esta práctica son:

- Conocer la sintaxis básica del lenguaje Postscript.
- Crear varios archivos y probar el funcionamiento con un visor Postscript.
- Transformar los ejemplos creados a otros formatos de archivo, como PDF.

2. ¿Qué es lenguaje PostScript?

El lenguaje PostScript (PS) es el código de programación que se usa para especificarle a una impresora qué y cómo debe imprimir, paso a paso. Fue creado por John Warnock y Charles Geschke en 1984 en Adobe Systems.

PS es un lenguaje de descripción de página (LDP), y un documento PS es un pequeño programa que le dice a una máquina qué, cómo y dónde imprimir: cualquier documento PS que imprimamos (p.ej. PDF o EPS) se interpreta en el procesador interno de la impresora.

Existen varios LDP, pero PS es el que ha tenido más éxito y el que se ha convertido en un estándar (casi todas las impresoras entienden PS).

3. Cómo funciona PostScript

PS es un lenguaje interpretado, de forma que se necesita un procesador o intérprete (un dispositivo físico o un programa en el ordenador) entre el código PS y la impresora física.

Si abrimos un archivo PS, veremos una serie de simples instrucciones escritas en formato texto plano. Esas instrucciones, habitualmente "vectoriales", vienen a decir cosas como "*traza un círculo de 3 cm. de radio*", o "*dibuja una línea de 4,5 cm. en un ángulo de 45 grados*".

4. Estructura de un fichero PostScript

Todo fichero PS tendrá un encabezamiento que lo identificará como fichero de este tipo. Habitualmente será: **%!PS-Adobe-3.0** y terminará con una marca de final de fichero del tipo **%%EOF** , **%EPS grestore** , **%EndPhotoshop**

Los ficheros PS (del tipo que sean) pueden contener además de texto, objetos vectoriales y mapas de imágenes.

Para crear ficheros PS se puede usar un simple editor de textos. Es lo que haremos aquí, aunque lo habitual es utilizar editores de texto que se ocupan de codificar en instrucciones PS la maquetación del texto o imágenes que se hayan incluido en el documento (es lo que hace MSWord o Writer cuando imprimimos a una impresora real o a un archivo PDF).

5. Características del lenguaje PostScript

Como lenguaje de programación, PS es muy sencillo, y sólo hay que conocer unos pocos conceptos:

- **Comentario (*Comment*):** Un comentario en PS es cualquier texto precedido por un "%". El comentario especial "%!" situado justo al principio de un archivo PS lo identifica como este tipo de archivo.
- **Pila (*Stack*):** Existen varias pilas en un sistema PS, aunque las más importantes son la de operandos (donde se ponen los argumentos para los procedimientos) y la de diccionario (proporciona almacenamiento para las variables).
- **Nombre (*Name*):** Un nombre es cualquier secuencia de caracteres que no es un número.
- **Número (*Number*):** Se pueden usar enteros y reales.
- **Cadena (*String*):** La sintaxis fuerza a especificar las cadenas de caracteres entre paréntesis.
- **Matriz (*Array*):** Los arrays son similares a los de cualquier otro lenguaje (se usan los corchetes []).
- **Procedimiento (*Procedure*):** Se usan para definir nuevos operadores.
- **Camino (*Path*):** Es un conjunto de líneas o elementos que definen un objeto a dibujar. Se inicia con la instrucción "newpath" y se finaliza con "stroke", que es la instrucción que pasa a representarlo realmente.

El objetivo último de PS es dibujar gráficos en la página. Así, para dibujar un **rectángulo**, podemos usar los operadores "moveto" y "lineto" dentro de un path:

```
%!
%% Draws a square box at (x,y)=(10,200) from bottom-left
newpath                % Start a new path
10 200 moveto          % move pointer to (x,y)=(10,200)
90 0 rlineto
0 90 rlineto
-90 0 rlineto
closepath
5 setlinewidth
stroke                 % finish the path (paint the lines)
showpage              % We're done
```

Ahora guardamos esas instrucciones en un archivo de texto plano al que le pondremos extensión .PS y lo visualizamos.

Si queremos dibujar arcos o **círculos**, usaremos la instrucción "arc":

```
%!PS

newpath

0 setgray              % poner color de lápiz a negro
0.1 setlinewidth       % poner grosor de línea (muy fina)

% x y R angI angF      % dar coordenada, radio y ángulos
40 70 30 0 360 arc

stroke                 % no se pinta nada hasta esta instrucción

showpage              % mostrar la página
```

Si queremos rellenar una figura que hayamos dibujado, usaremos "fill":

```
%!PS

newpath
% x y R angI angF
40 70 30 0 360 arc % dibujar un círculo gris
0.7 setgray        % poner el color a gris claro
fill               % rellenamos la figura recién terminada

0 setgray
1 setlinewidth
40 70 30 0 360 arc % dibujar una circunferencia negra

stroke             % no se pinta nada hasta esta instrucción

showpage          % mostrar la página
```

Para usar otros **colores**, usaremos la instrucción "setrgbcolor". Así, para escribir algo en color rojo (u otros colores), usaremos la instrucción indicando la cantidad de rojo, verde y azul que queremos usar en ese color:

```
1 0 0 setrgbcolor % poner color rojo
0.5 0 0.5 setrgbcolor % poner color amarillo oscuro
0 0 0 setrgbcolor % poner color negro
1 1 1 setrgbcolor % poner color blanco
```

Para mostrar **texto** en una página, debemos seguir los siguientes pasos:

- Establecer la fuente de letra.
- Indicar el punto exacto respecto a la esquina inferior izquierda para mostrar el texto.
- Especificar la cadena a imprimir.

Si por ejemplo queremos escribir en "Times Roman" con tamaño de 20 puntos, usaremos las instrucciones "setfont", "moveto" y "show" como sigue:

```
%!
% Sample of printing text

/Times-Roman findfont % Get the basic font
20 scalefont          % Scale the font to 20 points
setfont               % Make it the current font

newpath               % Start a new path
72 72 moveto          % Lower left corner of text at (72, 72)

(Hello, world!) show  % Typeset "Hello, world!"
stroke

showpage
```

Ahora guardamos esas instrucciones en un archivo de texto plano al que le pondremos extensión .PS y lo visualizamos. Veremos la cadena "Hello, world!" en "Times Roman" con tamaño de 20 puntos, impresa en la coordenada (72,72) desde la esquina inferior izquierda.

6. Comandos del lenguaje PS

A continuación se detallan los principales comandos de dibujo:

- **%!** un programa PS comienza con estos dos símbolos
- **x y moveto** mueve el "lápiz" a (x,y) sin dibujar nada
- **x y lineto** mueve el lápiz hasta (x,y), añadiendo el segmento de línea recta al path actual
- **closepath** añade un segmento de línea recta que conecta el punto actual con el punto de inicio del path actual (normalmente el punto especificado más recientemente por moveto), "cerrando" así el path actual
- **stroke** dibuja una línea alrededor del path actual (y lo finaliza, por lo que para empezar a dibujar un nuevo path hay que iniciar uno nuevo)
- **fill** pinta toda la región delimitada por el trazado actual (usando el color actual)
- **showpage** finaliza la página ("expulsa la página de la impresora"). Suele ser la última instrucción del archivo PS
- **x setlinewidth** cambia el grosor de línea
- **x setgray** cambia el color del lápiz a ese tono de gris (0 = negro, 1 = blanco)
- **r g b setrgbcolor** cambia el color del lápiz a un color (r,g,b), con valores entre 0 y 1
- **x y r ang1 ang2 arc** añade un arco de un círculo al path actual

En la última referencia (al final de este documento) se ofrece la URL de una página donde se detallan todos los comandos disponibles en el lenguaje PS:

add	end	moveto	setfont
arc	exch	mul	setgray
begin	fill	newpath	setlinewidth
bind	for	pop	show
clip	findfont	restore	showpage
charpath	grestore	rlineto	stroke
closepath	gsave	rmoveto	sub
curveto	if	rotate	translate
def	ifelse	save	
div	index	scale	
dup	lineto	scalefont	

7. Convertir los archivos a PDF

Existen diversas herramientas para convertir los archivos PS a un formato más universal como el PDF.

La suite de herramientas de LaTeX incluye diversas herramientas, como por ejemplo el comando: `ps2pdf`

Por otro lado, si no disponemos de LaTeX, o estamos usando un dispositivo móvil, la alternativa pasa por usar un servicio online, como <https://online2pdf.com/en/ps2pdf>



Para usarlo, sólo tenemos que subir el archivo PS a la web e indicar el formato al que queremos convertirlo (PDF, jpg, png, epub, etc).

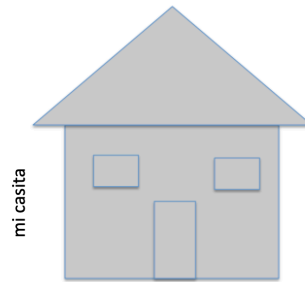
Cuestiones a resolver

El objetivo principal es conocer cómo utilizar el lenguaje Postscript a nivel básico.

El estudiante debe estudiar la sintaxis del lenguaje y crear varios archivos Postscript para aprender cómo crear figuras, líneas, textos y establecer propiedades de esos elementos. A continuación comprobará su correcto funcionamiento con un visualizador de este tipo de archivos.

Concretamente, en esta práctica se realizarán los siguientes archivos PS:

1. Página en la que se muestre el dibujo de una casita, similar al siguiente:



2. Página en la que diseñes una tarjeta de visita, similar a la siguiente:



Como resultado **se mostrará** al profesor el funcionamiento de cada archivo propuesto. Adicionalmente se convertirán a formato PDF y JPG para subirlos al repositorio. Se aconseja sacar diversas capturas de pantalla de los programas en las que se muestre el funcionamiento de cada uno.

En el documento a entregar se describirá cómo se han creado los diferentes ejemplos y se mostrará el correcto funcionamiento con capturas de pantalla.

Normas de entrega

La práctica o seminario podrá realizarse de manera individual o por grupos de hasta 2 personas.

Se entregará como un archivo de texto en el que se muestre la información requerida. También se puede utilizar la sintaxis de Markdown para conseguir una mejor presentación e incluso integrar imágenes o capturas de pantalla. La entrega se realizará subiendo los archivos necesarios al repositorio “**PDIH**” en la cuenta de GitHub del estudiante, a una carpeta llamada “**P4**”.

Toda la documentación y material exigidos se entregarán en la fecha indicada por el profesor. No se recogerá ni admitirá la entrega posterior de las prácticas/seminarios ni de parte de los mismos.

La detección de copias implicará el suspenso inmediato de todos los implicados en la copia (tanto de quien realizó el trabajo como de quien lo copió).

Las faltas de ortografía se penalizarán con hasta 1 punto de la nota de la práctica o seminario.

Referencias

<https://es.wikipedia.org/wiki/PostScript>
<https://staff.science.uva.nl/a.j.p.heck/Courses/Mastercourse2005/tutorial.pdf>
<https://www.diligent.es/lengiaje-postscript-021studio-diseno-grafico-barcelona/>
http://gusgsm.com/que_es_el_lenguaje_postscript
<https://www2.cs.duke.edu/courses/fall03/cps100/assign/recursivegraph/postscript-help.htm>
<http://pages.cs.wisc.edu/~ghost/>
<https://www.ghostscript.com/releases.html>
<http://www.ghostgum.com.au/software/gsview.htm>
<http://www.tailrecursive.org/postscript/postscript.html>
<http://www.tailrecursive.org/postscript/examples/box1.html>
<http://www.tailrecursive.org/postscript/examples/hello-text.html>
<http://www.tailrecursive.org/postscript/examples/newsprint.html>
<http://www.tailrecursive.org/postscript/operators.html#opindex>
<https://online2pdf.com/en/ps2pdf>