Program structure/functions

import "filename"
import module
import "filename" as name
include "filename"
include verbatim text from file
type f(type,...); optional function declaration
type name; variable declaration
type f(type arg,...) { function definition
  statements
  return value;
}

Data types/declarations

boolean (true or false)
bool

tri-state boolean (true, default, or false)
bool3

integer
int

float (double precision)
real

ordered pair (complex number)
pair

character string
string

fixed piecewise cubic Bezier spline
path

unresolved piecewise cubic Bezier spline
guide

color, line type/width/cap, font, fill rule
pen

label with position, alignment, pen attributes
Label

drawing canvas
picture

affine transform
transform

constant (unchanging) value
const

allocate in higher scope
static

no value
void

inhibit implicit argument casting
explicit

structure
struct
typedef type name

3D data types (import three)

ordered triple
triple

3D path
path3

3D guide
guide3

3D affine transform
transform3

Constants

exponential form
6.02e23

\la\text{TeX} string constant
"abc\text{de}"
\la\text{TeX} strings: special characters
\\, \`
C strings: constant
'abc\text{de}'
C strings: special characters
\\, \`
C strings: newline, cr, tab, backspace
\n \r \t \b
C strings: octal, hexadecimal bytes
\0-\377 \x0-\x7f

Operators

arithmetic operations
+ - * /
%
== != > >= <=
!
&& ||
(type) expr
++ --
+= -= *= /= %=

and or (conditional evaluation of RHS)
and or xor
cast expression to type
increment decrement prefix operators
assignment operators
conditional expression
structure member operator
expression evaluation separator

Flow control

statement terminator
;
block delimeters
{
}
comment delimeters
/* */
//
exit from while/do/for
break;
continue;
return expr;
exit();
abort(string);

Flow constructions (if/while/for/do)

if(expr) statement
else if(expr) statement
else statement

while(expr)
statement

for(expr1; expr2; expr3)
statement

for(type var : array)
statement

do statement
while(expr);
**Arrays**

array
array element i
array indexed by elements of int array A
anonymous array
array containing n deep copies of x
length
cyclic flag
pop element x
push element x
append array a
insert rest arguments at index i
delete element at index i
delete elements with indices in [1,j]
delete all elements
test whether element n is initialized
array of indices of initialized elements
complement of int array in \( \{0, \ldots, n-1\} \)
deep copy of array a
array \( \{0,1,\ldots,n-1\} \)
array \( \{n,n+1,\ldots,m\} \)
array \( \{n-1,n-2,\ldots,0\} \)
array \( \{f(0),f(1),\ldots,f(n-1)\} \)
array obtained by applying \( f \) to array a
uniform partition of \([a,b]\) into \( n \) intervals
concat specified 1D arrays
return sorted array
return array sorted using key
search sorted array a for key
index of first true value of bool array a
index of nth true value of bool array a

**Initialization**

initialize variable
initialize array

**path connectors**

straight segment
Bezzi segment with implicit control points
concatenate
lift pen
..tension at least 1...
..tension at least infinity...

**Labels**

implicit cast of string s to Label
Label a with relative position and alignment
Label a with absolute position and alignment
Label a with specified pen

**draw commands**

draw path with current pen
draw path with pen
draw labeled path
draw arrow with pen
draw path on picture
draw visible portion of line through two pairs

type[] name;
name[1]
name[A]
new type[dim] array(n,x)
name.length
name.cyclic
name.pop()
name.push(x)
name.append(a)
name.insert(1,...)
name.delete(i)
name.delete(i,j)
name.delete()
name.initialized(n)
name.keys
complement(a,n)
copy(a)
sequence(a)
sequence(n)
reverse(n)
sequence(f,n)
map(f,a)
uniform(a,b,n)
concat(a,b,...)
sort(a)
sort(a,less)
search(a,keys)
find(a)
find(a,n)

**fill commands**

fill path with current pen
fill path with pen
fill path on picture

**label commands**

label a pair with optional alignment z
label a path with optional alignment z
add label to picture

**clip commands**

clip to path
clip to path with fill rule
clip picture to path

**pens**

Grayscale pen from value in \([0,1]\)
RGB pen from values in \([0,1]\)
CMYK pen from values in \([0,1]\)
RGB pen from hexadecimal string
hexadecimal string from rgb pen
 hsv pen from values in \([0,1]\)
 invisible
 default pen
 current pen
 solid pen
 dotted pen
 wide dotted current pen
 wide dotted pen
 dashed pen
 long dashed pen
 dash dotted pen
 long dash dotted pen
 PostScript but line cap
 PostScript round line cap
 PostScript projecting square line cap
 miter join
 round join
 bevel join
 pen with miter limit
 zero-winding fill rule
 even-odd fill rule
 align to character bounding box (default)
 align to TeX baseline
 pen with font size (pt)
 TeX pen from encoding,family,series,shape
 TeX pen
 scaled TeX pen
 PostScript font from strings
 pen with opacity in \([0,1]\)
 construct pen nib from polygonal path
 pen mixing operator

\[ + \]
path operations

- number of segments in path \( p \)
- number of nodes in path \( p \)
- is segment \( i \) of path \( p \) straight?
- is path \( p \) cyclic?
- coordinates of path \( p \) at time \( t \)
- direction of path \( p \) at time \( t \)
- direction of path \( p \) at length \( p \)
- unit \( \text{dir}(p)+\text{dir}(q) \)
- acceleration of path \( p \) at time \( t \)
- radius of curvature of path \( p \) at time \( t \)
- precontrol point of path \( p \) at time \( t \)
- postcontrol point of path \( p \) at time \( t \)
- arclength of path \( p \)
- time at which arclength \( p \)=L
- point on path \( p \) at arclength \( L \)
- first value \( t \) at which \( \text{dir}(p,t)=z \)
- time \( t \) at relative fraction \( L \) of arclength \( p \)
- point at relative fraction \( L \) of arclength \( p \)
- point midway along arclength of \( p \)
- path running backwards along \( p \)
- subpath of \( p \) between times \( a \) and \( b \)
- times for one intersection of paths \( p \) and \( q \)
- times at which \( p \) reaches minimal extents
- times at which \( p \) reaches maximal extents
- intersection times of paths \( p \) and \( q \)
- intersection times of path \( p \) with \( \text{arc}(\text{at}=(0,0)) \)
- intersection times of path \( p \) crossing \( x=x \)
- intersection times of path \( p \) crossing \( y=y \)
- intersection point of paths \( p \) and \( q \)
- intersection points of \( p \) and \( q \)
- intersection of extension of \( P--Q \) and \( p--q \)
- lower left point of bounding box of path \( p \)
- upper right point of bounding box of path \( p \)
- subpaths of \( p \) split by \( n \)
- cut of knife \( p \) about pair \( z \)
- pair \( z \) lies within path \( p' \)
- pair \( z \) lies within or on path \( p' \)
- path surrounding region bounded by paths
- path filled by \( \text{draw}(g,p) \)
- unit square with lower-left vertex at origin
- unit circle centered at origin
- circle of radius \( r \) about \( c \)
- arc of radius \( r \) about \( c \) from angle \( a \) to \( b \)
- unit \( n \)-sided polygon
- unit \( n \)-point cyclic cross

pictures

- add picture \( \text{pic} \) to currentpicture
- add picture \( \text{pic} \) about pair \( z \)

length \( p \)
size \( p \)
cyclic \( p \)
straight \((p,i)\)
week plates \( (p) \)
dir \((p,q)\)
clock \((p)\)
radius \((p,t)\)
acell \((p,t)\)
precontrol \((p,t)\)
postcontrol \((p,t)\)
arcnumber \((p)\)
arcarray \((p,L)\)
arcpoint \((p,L)\)
dirtime \((p,z)\)
reltime \((p,l)\)
relpoint \((p,l)\)
midpoint \((p)\)
reverse \((p)\)
subpath \((p,a,b)\)
intersect \((p,q)\)
imaintimes \((p,q)\)
maxtimes \((p,q)\)
intersections \((p,q)\)
times \((p,x)\)
times \((p,z)\)
tangentpoint \((p,q)\)
tangentpoints \((p,q)\)
extension \((P,Q,p,q)\)
min \((p)\)
max \((p)\)
cut \((p,knife,n)\)
interior \((p,z)\)
inside \((p,z)\)
bubblecycle \((\ldots)\)
strokepath \((g,p)\)
unitsquare
unitcircle
circle \((c,r)\)
arc \((c,r,a,b)\)
polygon \((n)\)
cross \((n)\)

affine transforms

identity transform
shift by values
shift by pair
scale by \( x \) in the \( x \) direction
scale by \( y \) in the \( y \) direction
scale by \( x \) in both directions
scale by real values \( x \) and \( y \)
map \((x,y) \to (x+ay,y)\)
rotate by real angle in degrees about pair \( z \)
reflect about line from \( P--Q \)

string operations

concatenate operator
string length
position \( \geq \) pos of first occurrence of \( t \) in \( s \)
position \( \leq \) pos of last occurrence of \( t \) in \( s \)
string with \( t \) inserted in \( s \) at pos
substring of string \( s \) of length \( n \) at pos
string \( s \) reversed
string \( s \) with \( \text{before} \) changed to \( \text{after} \)
string \( s \) translated via \( \{\text{before,after},\ldots\} \)
format \( x \) using \( C \)-style format string \( s \)
casts hexadecimal string to an integer
casts \( x \) to string using precision digits
current time formatted by format
time in seconds of string \( t \) using format
string corresponding to seconds using format
split \( s \) into strings separated by delimiter

May 2014 v1.1. Copyright © 2014 John C. Bowman

Permission is granted to make and distribute copies of this card, with or without modifications, provided the copyright notice and this permission notice are preserved on all copies.