Asymptote Reference Card

Program structure/functions
import "filename"
import module
import "filename" as name
include "filename"
include verbatim text from file

type f (type arg, ...) {
  statements
  return value;
}

Data types/declarations
boolean (true or false) bool
true, default, or false bool3
integer int
float (double precision) real
ordered pair (complex number) pair
tri-state boolean (true, default, or false) bool3
character string string
fixed piecewise cubic Bezier spline path
triple ordered triple triple
unresolved piecewise cubic Bezier spline guide
color, line type/width/cap, font, fill rule pen
label with position, alignment, pen attributes label
affine transform transform
canvas drawing canvas
constant (unchanging) value const
allocate in higher scope static
no value void
inhibit implicit argument casting explicit
structure 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
\TeX string constant "abc\ldots\de"
\TeX strings: special characters \\
C strings: constant 'abc\ldots\de'
C strings: special characters \\
C strings: newline, cr, tab, backspace \n \r \t \b
C strings: octal, hexadecimal bytes \0-\377 \X0-\XFF

Operators
arithmetic operations + - * /
modulo (remainder) %
comparisons == ! > >= <=
and or (conditional evaluation of RHS) !
& & ||
and or xor (type) expr
++ --
+= -= *= /= %=
expr1 ? expr2 : expr3
name.member

Flow control
statement terminator ;
block delimiters {
comment delimiters /* */

exit from while/do/for
next iteration of while/do/for
return value from function
terminate execution

abort execution with error message
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);

1
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]
test whether element n is initialized
array of indices of initialized elements
complement of int array in {0, . . . , n-1}
deep copy of array a
array {0, 1, . . . , n-1}
array {n, n+1, . . . , m}
array {n-1, n-2, . . . , 0}
array {f(0), f(1), . . . , 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 ordering less
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
Bezier 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 s with relative position and alignment
Label s with absolute position and alignment
Label s 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(n, m)
reverse(n)
sequence(f, n)
map(f, a)
uniform(a, b, n)
concat(a, b, . . .)
sort(a)
sort(a, less)
search(a, key)
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 pen
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 butt 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
scaled \TeX pen

\TeX pen from string
\TeX pen from strings
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(dir($p$))
acceleration of path $p$ at time $t$
radius of curvature of path $p$

arclength of path $p$
time at which arclength($p$) = $L$
point on path $p$ at arclength $L$
first value $t$ at which 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 "$a$--$b$--$c$--$

intersection times of path $p$ crossing $y = z$
y scale
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 $n$th cut of knife

pair $z$ lies within path $p$
pair $z$ lies within or on path $p$
path surrounding region bounded by paths
path filled by 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 pic to currentpicture
add picture pic about pair $z$

length($p$)
size($p$)
cyclic($p$)
straight($p$, $i$)
piecewisestraight($p$)
point($p$, $t$)
dir($p$, $t$)
dir($p$)
dir($p$, $q$)
accel($p$, $t$)
radius($p$, $t$)
precontrol($p$, $t$)
postcontrol($p$, $t$)
arc($p$)
arcpoint($p$, $L$)
arcpoint($p$, $L$)
dir($p$)
dactime($p$, $t$)
dirtime($p$, $z$)
reltime($p$, $l$)
relpoint($p$, $l$)
midpoint($p$)
reverse($p$)
subpath($p$, $a$, $b$)

intersect($p$, $q$)

mintimes($p$)

maxtimes($p$)

intersections($p$, $q$)

intersections($p$, $a$, $b$)
times($p$, $x$)
times($p$, $z$)

intersectionpoint($p$, $q$)

intersectionpoints($p$, $q$)

extension($P$, $Q$, $p$, $q$)

min($p$)

max($p$)
cut($p$, knife, $n$)
windingnumber($p$, $z$)

interior($p$, $z$)
inside($p$, $z$)
bucycle(...)
strokepath($g$, $p$)

unitsquare
ununitcircle
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 ($x$, $y$) $\rightarrow (x+ay, y)$
rotate by real angle in degrees around point $z$
reflect about line from $P--Q$

reflect($P$, $Q$)

string operations

concatenator 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 before changed to after

string $s$ translated via {{before,after},...}

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

identity($t$)
shift(real, real)$t$
shift(pair)$t$

xscale($x$)$t$
yyscale($y$)$t$

scale($x$)$t$
scale($x, y$)$t$
slant($s$)$t$

rotate(angle, z=(0,0))$t$

reflect($P$, $Q$)$t$

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.