```
# first part of rintro.pdf...

# Basics

x <- 3
x
print(x)

x = 5     # NOTE: the rintro.pdf document uses "_" instead of "=".
x         # "=" is the correct symbol! (synonym for "<-")

y <- "Hello there"
y

y <- sqrt(10)
y

z <- x + y
z

q()       # use this to quit

################

# A basic R object is the "vector". R always prints some index hints
# to the left of a printout of a vector.

# A scalar in R is just a vector of length 1, so we still get the
# index hint to the left of the printout

x <- 1:5
print(x)

xx <- 1:50
xx

x[1] <- 17
x

x[3:5] <- 0
x

w <- x[-3]
w

y <- c(1,5,2,4,7)
y
y[2]
```

```
y[-3]
y[c(1,4,5)]

i <- 1:3
z <- c(9,10,11)
y[i] <- z
y

y <- y^2
y

y <- 1:10
y <- log(y)
y

y <- exp(y)
y

x <- 1:10
y <- c(5,4,3,2,1,5,4,3,2,1)
z <- x + y
z

x <- 1:10
y <- c(5,4,3,2,1,5,4,3,2,1)
x == 2
z <- (x == 2)
z

z <- (x<5); z       # ";" separates commands on the same line

                    # you can break an expression across lines
                    # if the expression is incomplete on the current line
x[x<5] <- y[x<5]
x

############

# matrices are 2-dimensional arrays of elements of the same type

junk <- c(1,2,3,4,5, 0.5, 2, 6, 0, 1, 1, 0)
m <- matrix(junk,ncol=3)
m

m <- matrix(junk,ncol=3,byrow=T)
m

dim(m)
```

```r
y <- m[,1]
y

x <- m[,2]
x

z <- m[1,2]
z

m
zz <- t(m)       # matrix transpose!
zz

new <- matrix(1:9,ncol=3)
new
hello <- z + new
hello

m[1,3]
subm <- m[2:3,2:4]
m[1,]
m[2,3] <- 7
m
m[2,3]
m[,c(2,3)]
m[-2,]

############

# Something slightly statistical to illustrate built-in functions

x <- runif(100,0,1) # random sample of size 100 from Unif[0,1]
mean(x)
y <- mean(y)
y
help(mean)
min(x)
max(x)
summary(x)
help(summary)       # most commands in R have a help page!


############

# aside: a vector can only have elements of one type in it.
#        a matrix can only have elements of one type in it.
#
```

```r
# if you want a vector-like object with different data types in it,
# use a **list**.
#
# if you want a matrix-like object with different data types in it,
# use a **data.frame** (which is basically a list of columns).
# data read into R from external files is usually stored in a
# data frame, by default.

# Here are som examples involving lists

who <- list(name="Joe", age=45, married=T)
who

who$name
who[[1]]

who$age
who[[2]]

who$married
who[[3]]
names(who)

who$name <- c("Joe", "Steve", "Mary")
who$age <- c(45,23)
who$married <- c(T,F,T)
who

#########

# there is more useful info, on loops and creating and using
# functions, in the rintro.pdf handout!

#########
```