# Class 6 R Funcions

My first function :-)

```
add <-  function(x,y){
  x+y
}
```

Can I just use it?

```
add(1,1)
```

```
[1] 2
```

```
add(x=1, y=100)
```

```
[1] 101
```

```
add(c(100,1,100),1)
```

```
[1] 101   2 101
```

**Assignment:** Make a function "generate_dna()" that makes a random nucleotide sequence of any length

```
bases <- c("A","T","C","G")
sequence <- sample(bases, size=5, replace=TRUE)
```

This is my wee working snippet now I can make it into a function.

```r
generate_dna <- function(length){
  bases <- c("A","T","C","G")
  sequence <- sample(x=bases,size=length, replace=TRUE)
  return(sequence)
  }
```

```r
generate_dna(20)
```

```
 [1] "G" "A" "G" "T" "G" "T" "C" "A" "G" "A" "A" "C" "T" "C" "C" "A" "G" "A" "C"
[20] "A"
```

```r
# install.packages("bio3d")
# Gives you access to protein amino acids, write, read sequences, run BLAST searches

# "::" allows you to import portion of a package
# Looks up amino acids in bio3d package
# Only use unique ones as there are repetitive aa that are chemically modified
# Only use 1-20 because X is last aa and not common/used
aa <- unique(bio3d::aa.table$aa1)[1:20]
```

```r
generate_prot <- function(length){
  aa <- unique(bio3d::aa.table$aa1)[1:20]
  protein_seq <- sample(x=aa,size=length, replace=TRUE)
  protein_seq <- paste(protein_seq, collapse="")
  return(protein_seq)
  }
```

```r
generate_prot(20)
```

```
[1] "FWGAAPCNVIWYSEYAEYQV"
```

**Assignment:** Generate random protein sequences of length 6 to 13

```r
# sapply() and other apply() functions can be used to run a function multiple times
# This takes advantage of the vectorization of R - if you are writing
# for loops in R you are typically doing it wrong and inefficiently
#apply() works with tables/data frames

answer <- sapply(6:12, generate_prot)
```

```
cat(paste(">id.", 6:12, "\n", answer, sep=""), sep="\n")
```

```
>id.6
QNVPQY
>id.7
LGRYVPK
>id.8
YLHHTIHW
>id.9
IDHFYQQPQ
>id.10
PEQLRFRWFW
>id.11
RPIICPIDYYT
>id.12
AGCSCGIHHQEP
```

```
paste(c("barry", "alice", "amy"), "loves R", sep=" ", collapse = " ")
```

```
[1] "barry loves R alice loves R amy loves R"
```

```
?cat()
```

`\n` is a new line