What We Expect You to Know

Functions

Know the following about functions

- How to define functions
- Control flow, `return` statement
- Functions are objects, and can be defined as anonymous functions
- Arguments, argument default values, missing arguments, and when arguments may be missing
- Function argument matching by position and by name
- How variable scoping works. What variables can a function access, what is variable shadowing, and what variables can a function modify?
- Copy semantics prevents functions from changing their arguments directly
- how the `...`-arguments work, how to get the number and values of `...`-arguments, how to pass them on to other function calls
- `do.call` to call functions with a list of arguments

What We Expect You to Know

`lapply()` and similar Functions

- sapply(): simplify to a vector / matrix, if possible
- vapply(): sapply, but with *type safety*: ensure the return-type of a function!
- mapply(): sapply, but going along more than one input list/vector
  - Map(): like mapply(), but not simplifying (like the difference sapply <--> lapply)
- apply(): sapply, but to to each row / column of a matrix
- tapply(): INDEX argument indicates *group*. apply group-wise:
- rapply(): recursive lapply (don't need to know this one)
- Reduce(): Apply a two-argument successively to elements and previous result
- Filter(): return only members for which a "predicate" is TRUE
- Find(): Like "Filter()" but return only first element
- Position(): Like "Find()", but return position instead of element

What We Expect You to Know

`checkmate` Functions

- Difference between testXxx, checkXxx, assertXxx functions, and why they are useful
- For a given constraint (type, minimum, length, null / NA ok) you should be able to find the corresponding checkmate assertXxx-function.
  - In particular: vectors, scalars, sets (i.e. atomic vectors with ordering disregarded), data.frame, matrix, function, NULL
- use assert(check(...), check(...), ..) to assert "or"-conditions
- `%???%`-operator to get first non-NULL value

If You Want to Go Beyond This

What we are not covering here: environments, frames and call stacks, closures, promises and lazy argument evaluation. If you want to learn more about that:

- check out Advanced R's chapters on functions and on environments.
- look at the R Language Definition sections 3.5, 4.3.3 and the R help pages of the functions mentioned there