

What We Expect You to Know

- **Classes** specify the structure of **Objects** which have **fields** and **methods**
- OOP in R:
 - S3: copy-semantics, implicit structure
 - R6: reference-semantics, explicit structure
 - some others

R6

- Define class:
`R6Class(<name>, public = list(..), active = list(..), private = list(..))`
 - **public**: accessible from outside
 - **private**: accessible through special variable "private"
 - **active**: "active binding" function, looks like a field from outside
- Instantiate: `<Object Generator>$new()`
- Inheritance:
 - `R6Class(..., inherit = <superclass>, ...)`
 - Methods and fields from superclass if not overwritten
- Special variables inside methods: `self`, `private`, `super`.
- Special methods: `initialize()`, `deep_clone()`.
- Deep copy: `<Object>$clone(deep = TRUE)` , calls `deep_clone()` for all fields & methods

What We Expect You to Know

S3

- **attributes:** Additional information hanging on to objects in R
 - "names": names of lists / vectors, "dim": dimension of matrix / array
 - "class": S3 class
 - Access through `attr(<obj>, <name>)` or `attributes(<obj>)$<name>`
 - Set conveniently with `<-` or `structure(<obj>, <name> = <value>, ...)`
- Create S3-object by setting "class" attribute
- Define class: Constructor function

```
<ClsName> <- function(..) { .. structure(list(..), class = "<ClsName>" ) }
```
- **Generic function:** `<fname> <- function(...) UseMethod("<fname>")`
- **S3 Method:** `<fname>.<ClsName> <- function(...) { .. }`
 - Should have compatible signature (i.e. arguments) with generic
- Special method `print.<ClsName> <- function(x, ...)`
 - called automatically when object is displayed.
 - should have "x" and "..." arguments and must return `invisible(x)`.
- Inheritance through multiple entries in "class" attribute vector. Subclass first, superclass next.
- `NextMethod()` : call to superclass method
- assert through `assertClass(<obj>, "<ClsName>")` and assume internal structure is valid