## Recordings

Why tell when you can show?

REPL-based programming often falls short of true interactivity. In most interpreted languages, constructs such as classes and methods require too much *top-down* design, creating unnecessary cognitive load in the prototyping phase.

Recordings are bottom-up functions. They can be created after the fact in a REPL, and can be reused, composed with other functions, or added to a module or class.

```
7. Use
the REPL

7. Tender about if path is '/about'

7. In the REPL

8. Select variables to abstract over

9. Server.addMethod handle_routes

9. Select variables to abstract over

9. Server.rb>
class server

1. Use
the REPL

9. Tender about if path is '/about'

1. Use
the REPL

9. Select variables to abstract over

9. Select variables to abstract over

1. Use
the REPL

9. Tender about if path is '/about'
end
end

1. Use
the REPL

2. Select
to commands to be recorded
to be recorded
the REPL

1. Use
the REPL

2. Select
to commands to be recorded
the REPL

1. Use
the REPL

2. Select
to commands to be recorded
the REPL

1. Use
the REPL

2. Select
to commands to be recorded
to be recorded
to be recorded
the REPL

2. Select
to commands to be recorded
to be recorde
```