# BASH SCRIPTING

# RECAP: SO FAR

- Creating and running
- Poundbang, comments
- Arguments, variables
- Arithmetic
- Conditionals
- Loops

# ARRAYS

- Similar to arrays/lists in other languages
- Initializing:
  - `arr=(hello world)`
  - `arr_ex2[2]=hello`
    `arr_ex2[3]=world`
  - `arr_ex3=()`
  - `declare -a arr_ex4`

# ARRAYS (CONT.)

- Accessing elements:
  - Bracket notation
  - `${arr[0]}`
- Appending elements:
  - `arr+=(goodbye bonjour)`
- Accessing slice:
  - `${arr[@]:start:num}`

# ARRAYS (CONT.)

- Special Accessing
  - `${arr[@]}` - all elements
  - `${arr[*]}` - all elements as one string
- Special Values
  - `${!arr[@]}` - array indices
  - `${#arr[@]}` - number of elements in array

# ASSOCIATIVE ARRAYS

- Can use non-integer indices
- aka maps, dictionaries in other languages
- Example:

```
declare -A aa_ex1
aa_ex1=(
  [rainy]=6
  [cloudy]=7
  [warm]=1
)
```

# FUNCTIONS

- Definition:

```
function_name () {
  commands
}
```

- Calling: just `function_name`
- Parameters:
  - Call function name with arguments (like we'd call bash script with arguments)
  - Access within function like you access arguments to bash script
  - Aka: use `$1`, `$2`, ...

# MINILAB 09

- Same as last time -- ran out of time