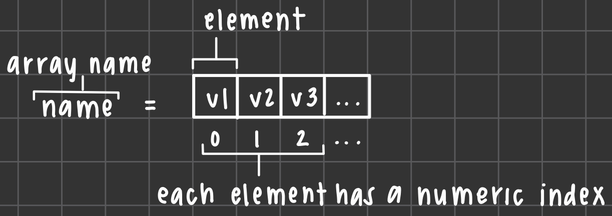
Lecture Materials – Week 3

JavaScript Arrays

- array = a collection of values
  - format: var name = [value1, value 2, value3, ...];
    - → (eg. var grades = [80, 87, 94, 82, 98, 73];)
    - → (eg. var foods = ["bananas", "apples", "pizza"];)
    - → (eg. var info = ["Erin", 23, "October", 2000];): elements don't have to be the same type
      uncommon
  - format using API methods: var name = document.method["selector"];
    - → (eg. var images = document.getElementsByClassName["imgs"];)
    - → (eg. var listItems = document.getElementsByTagName["li"];)

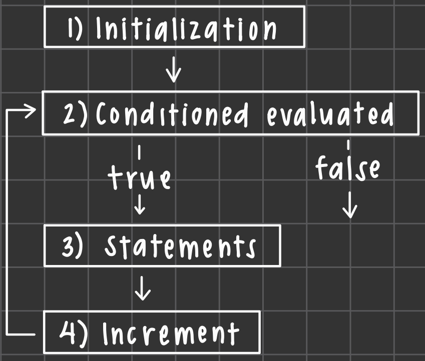


- element = each value in an array
- accessing an array: elements are referenced by their index
  index starts count at 0 → index = element placement - 1
- array attributes & methods: .length
  .sort()
  .push(element)
  - add element to array
  - alternative method: name[name.length] = element;

JavaScript Iteration

- iteration / looping = the best way to access all the data in an array
- for loop: 1. set a variable to the initial value
  2. run a boolean test case → true / false
  3a. true → run code
  3b. false → exit loop
  4. update your variable & go back to step 2

→ (eg. $\text{for } (\underbrace{i=0;}_{(1)}\ \underbrace{i < \text{array.length};}_{(2)}\ \underbrace{i{+}{+}}_{(4)})\ \{\underbrace{\ldots}_{(3a)}\})$



Flow of Control

- decision points: add variety to the program
  react to good / bad user input
  avoid potential errors

- flow of control = execution of only applicable & needed blocks of code → efficient

- if statement = evaluates a boolean expression before performing an action
  if expression = true → execute code
  if expression = false → skip over it

  - format:
```
if (boolean expression) {
   statement(s);
}
```

- if-else statement = if true → execute statement1
  else false → execute statement2

  - format:
```
if (boolean expression) {
    statement1;
} else {
    statement2;
}
```

- NAN = not a number

Advanced Conditionals

- complex boolean statements: if you need to check for 2 conditions
  if one condition depended on another

- nested if statement = to put one if statement inside another
  if-else statement → else statement matches most recent if statement

  - format:
```
if (boolean expression) {
   statement(s);
   if (new boolean expression) {
       statement(s) to execute if both true;
   }
}
```

Common Errors

- 2 classes of errors: syntatic
  logic

- syntatic errors = break the "rules" of JavaScript
  will appear in browser console on laptop/desktop

  → (eg. typed something wrong)

  → (eg. forgot to close a curly bracket)

  → (eg. using an undefined variable)

logical errors = code is valid, but something is wrong w/ your thought process
some are typos... typos that run

⟶ (eg. didn't check if dividing by 0)

- issues w/ comparisons: 5 == "5" → true
  if (name = "Erin") → value assignment not equality check
  (age < 18) && (age > 65) → will never be true
  (age > 18) || (age > 65) → choose one lol

- concatenation v. addition: 5+5 → 10
  "5"+"5" → 55
  "5"+5 → 55

  - the + operator performs different actions based on the type

- issues w/ nesting: else matches w/ wrong if
  misplacing semicolons

  ⟶ (eg. if (age < 18); {...}) :

  ⟶ (eg. for (c=0; c<5; c++); {...}) :

- use the console whenever possible
  save code often

[Q. built in JavaScript functions can be mixed in w/ other HTML code w/o <script> tag
by attaching an event to any element in the DOM
Q. if a function is defined twice, the second declaration will be used]