

* Switch case statements :

day === 'monday'
~~'Monday' === 'monday'~~

* == → loose value ✓

* === → strict value ✓
datatype ✓

1 == '1' → true

1 === '1' → false

```
268 const day = "monday";
269
270 switch (day) {
271   case "monday":
272     console.log("plan course curriculum");
273     console.log("Arrange a team meeting");
274     break;
275
276   case "tuesday":
277     console.log("prepare videos for theory");
278     break;
279
280   case "wednesday":
281   case "thursday":
282     console.log("write examples for teaching");
283     break;
284
285   case "friday":
286     console.log("do a reharsal for the class");
287     break;
288
289   case "saturday":
290   case "sunday":
291     console.log("enjoy the weekend!!!!");
292     break;
293
294   default:
295     console.log("Not a valid day");
296 }
```

```
298 if (day === "monday") {
299   console.log("plan course curriculum");
300   console.log("Arrange a team meeting");
301 } else if (day === "tuesday") {
302   console.log("prepare videos for theory");
303 } else if (day === "wednesday" || day === "thursday") {
304   console.log("write examples for teaching");
305 } else if (day === "friday") {
306   console.log("do a reharsal for the class");
307 } else if (day === "saturday" || day === "sunday") {
308   console.log("enjoy the weekend!!!!");
309 } else {
310   console.log("Not a valid day!!!");
311 }
```

→ write switch when you have large if, else if, else ladders. to improve code structure, readability.

→ All statements below a matched case are executed until a break is occurred.

→ default is executed when no case is matched.

→ Case checking is strict '==='.

* Array Methods :

```
160 const arr = [1, 2, 3, 4];
161
162 // adding element at the end
163 arr.push(5);
164 console.log("Added 5 : ", arr);
165
166 // length
167 console.log(arr.length);
168
169 // remove last element
170 arr.pop();
171 console.log(arr);
172 arr.pop();
173 console.log(arr);
174
175 // remove first element
176 arr.shift();
177 console.log(arr);
178
179 // add elements at the start
180 arr.unshift(1);
181 console.log(arr);
182 arr.unshift(4);
183 console.log(arr);
```

```
185 // subarray
186 const animals = ["ant", "bison", "camel", "duck", "elephant", "lion"];
187 console.log("start 2 until end: ", animals.slice(2));
188 console.log(animals); // slice will not change the array unlike other methods
189 console.log("start 2 until 3: ", animals.slice(2, 4));
190 console.log("start 1 until 4: ", animals.slice(1, 5));
191 console.log("start -2 until end: ", animals.slice(-2)); // do not use negative
192
193 // splice method
194 const months = ["Jan", "Feb", "Mar", "Apr", "Jun"];
195 months.splice(1, 3); // (startIdx, deleteCount)
196 console.log(months);
197
198 // remove an element at specified index
199 const removeAt = 3;
200 months.splice(removeAt, 1);
201 console.log(months);
202
203 // add an element at specified index
204 const addAt = 1;
205 months.splice(addAt, 0, "Jul"); // (startIdx, delCount, item)
206 console.log(months);
207
208 // reversing an array
209 const actual = ["one", "two", "three", "four"];
210 actual.reverse(); // reverse is inplace
211 console.log(actual);
212
213 // sorting an array
214 const numberArr = [5, 3, 2, 1, 4];
215 numberArr.sort();
216 console.log(numberArr);
```

```
218 const strArr = ["March", "Jan", "Feb", "Dec"];
219 strArr.sort(); // sorts the string in dictionary (lexicographical) order
220 console.log(strArr);
221
222 const numberStrArr = [5, "abc", 3, "def"];
223 // ["5", "3", "abc", "def"]
224 numberStrArr.sort();
225 console.log(numberStrArr);
226
227 // join
228 const joined = strArr.join("");
229 console.log(joined);
```

*

$O(1)$ - push(nwm)

$O(1)$ - length

$O(1)$ - pop()

$O(n)$ - shift()

$O(n)$ - unshift()

$O(n)$ - slice(s, e)

$O(n^2)$ - splice(s, d, i)

$O(n)$ - reverse()

$O(n \log n)$ - sort()

└ merge sort

$O(n)$ - join()