

Hands – On Lab

Workshop 3.

AREA OF TRIANGLE

Write a function that takes the base and height of a triangle and **return** its area.

Example:

Areaoftriangle (3, 4) —→ 6

Areaoftriangle (7, 8) —→ 28

Notes

- Area of triangle is $(\text{base} * \text{height})/2$
- Don't forget to return the result

→

The screenshot shows a web-based code editor with a file explorer on the left, a code editor in the center, and a console on the right. The file explorer shows a project named 'workshop 3' with files 'index.js' and 'package.json'. The code editor shows the following code in 'index.js':

```
1 //AREA OF TRIANGLE
2
3 let Areaoftriangle = () => {
4   let num1 = parseInt(prompt("Enter Base of Triangle"))
5   let num2 = parseInt(prompt("Enter Height of Triangle"))
6   console.log((num1 + num2) / 2)
7 }
8 Areaoftriangle()
```

The console on the right shows the output of the code:

```
Enter Base of Triangle> 2
Enter Height of Triangle> 3
2.5
Hint: hit control+c anytime to enter REPL.
```

RETURN SOMETHING TO ME!

Write a function that returns the string "something" joined with a space " " and the given argument.

Examples

giveMeSomething("is better than nothing") → "something is better than nothing"

giveMeSomething("Bob Jane") → "something Bob Jane"

giveMeSomething("something") → "something something"

→

The screenshot shows a web-based code editor with a file explorer on the left, a code editor in the center, and a console on the right. The file explorer shows a project named 'workshop 3' with files 'index.js' and 'package.json'. The code editor shows the following code in 'index.js':

```
1 // RETURN SOMETHING TO ME!
2
3 let giveMeSomething = () => {
4   let defn = ("Something")
5   let argmnt = prompt("Enter other half of your argument.")
6   console.log(defn + " " + argmnt)
7 }
8 giveMeSomething()
```

The console on the right shows the output of the code:

```
Enter other half of your argument.> is suspicious.
Something is suspicious.
Hint: hit control+c anytime to enter REPL.
```

BASKETBALL POINTS

You are counting points for a basketball game, given the amount of 2 – pointer scored and 3 – pointer scored, find the final points for the team and return the value.

Example:

points (3,5) $\longrightarrow 3*2 + 5*3 = 21$

points (1,1) $\longrightarrow 5$

\rightarrow



The screenshot shows a code editor with a file named 'index.js' containing the following JavaScript code:

```
1 //BASKETBALL POINTS
2
3 let points = () => {
4   let point2 = parseInt(prompt("Enter number of 2 pointer scored:"))
5   let point3 = parseInt(prompt("Enter number of 3 pointer scored:"))
6   console.log("The total score is: " + ((point2 * 2) + (point3 * 3)))
7 }
8
9 points();
10
```

The console output shows the execution of the function with the following prompts and results:

```
Enter number of 2 pointer scored:> 4
Enter number of 3 pointer scored:> 2
The total score is: 14
Hint: hit control+c anytime to enter REPL.
```

LESS THAN 100?

Given two numbers, return true if the sum of both numbers is less than 100.

Otherwise return false.

Examples

lessThan100(22, 15) \rightarrow true

// 22 + 15 = 37

lessThan100(83, 34) \rightarrow false

// 83 + 34 = 117

lessThan100(3, 77) \rightarrow true

\rightarrow



The screenshot shows a code editor with a file named 'index.js' containing the following JavaScript code:

```
1 //LESS THAN 100?
2
3 let lessThan100 = () => {
4   let num1 = parseInt(prompt("Enter First number:"))
5   let num2 = parseInt(prompt("Enter Second number:"))
6   if ((num1 + num2) < 100){
7     console.log("True");
8   }
9   else {
10    console.log("false");
11  }
12 }
13
14 lessThan100();
```

The console output shows the execution of the function with the following prompts and results:

```
Enter First number:> 23
Enter Second number:> 45
True
Hint: hit control+c anytime to enter REPL.
```



ADD UPTO THE NUMBER FROM A SINGLE NUMBER

Create a function that takes a number as an argument. Add up all the numbers from 1 to the number you passed to the function. For example, if the input is 4 then your function should return 10 because $1+2+3+4 = 10$



```
1 //ADD UPTO THE NUMBER FROM A SINGLE NUMBER
2 let add = () => {
3   let num = parseInt(prompt("Enter a number:"))
4   let sum = 0;
5   for (let i = 1; i <= num; i++) {
6     sum += i;
7   }
8   console.log("The sum of numbers up to " + num + " is: " + sum);
9 }
10
11 add();
12
```

Console output:

```
Enter a number:> 5
The sum of numbers up to 5 is: 15
Hint: hit control+c anytime to enter REPL.
>
```

ANY PRIME NUMBER IN RANGE

Create a function that return true if there is at least one prime number in the given range(n_1 to n_2) inclusive, false otherwise.

Example:

`primeInRange(10,15)` —→ `true`

// prime number is range : 11, 13

`primeInRange(3,1)` —→ `true`

// prime number is range : 3, 5



```
1 let primeInRange = () => {
2   let n1 = parseInt(prompt("Enter first number:"))
3   let n2 = parseInt(prompt("Enter second number:"))
4
5   for (let i = n1; i <= n2; i++) {
6     let isPrime = true;
7     if (i < 2) {
8       isPrime = false;
9     } else {
10      for (let j = 2; j < i; j++) {
11        if (i % j == 0) {
12          isPrime = false;
13          break;
14        }
15      }
16    }
17    if (isPrime) {
18      return true;
19    }
20  }
21  return false;
22 }
23
24 if (primeInRange()) {
25   console.log("True");
26 } else {
27   console.log("False");
28 }
```

Console output:

```
Enter first number:> 1
Enter second number:> 10
True
Hint: hit control+c anytime to enter REPL.
>
```

ODDISH VS. EVENISH

Create a function that determines whether a number is Oddish or Evenish. A number is Oddish if the sum of all of its digits is odd, and a number is Evenish if the sum of all of its digits is even. If a number is Oddish, return "Oddish". Otherwise, return "Evenish".

For example, `oddishOrEvenish(121)` should return "Evenish", since $1 + 2 + 1 =$

4. `oddishOrEvenish(41)` should return "Oddish", since $4 + 1 = 5$.

Examples

`oddishOrEvenish(43) → "Oddish"`

`// 4 + 3 = 7`

`// 7 % 2 = 1`

`oddishOrEvenish(373) → "Oddish"`

`// 3 + 7 + 3 = 13`

`// 13 % 2 = 1`

`oddishOrEvenish(4433) → "Evenish"`

`// 4 + 4 + 3 + 3 = 14`

`// 14 % 2 = 0`

→

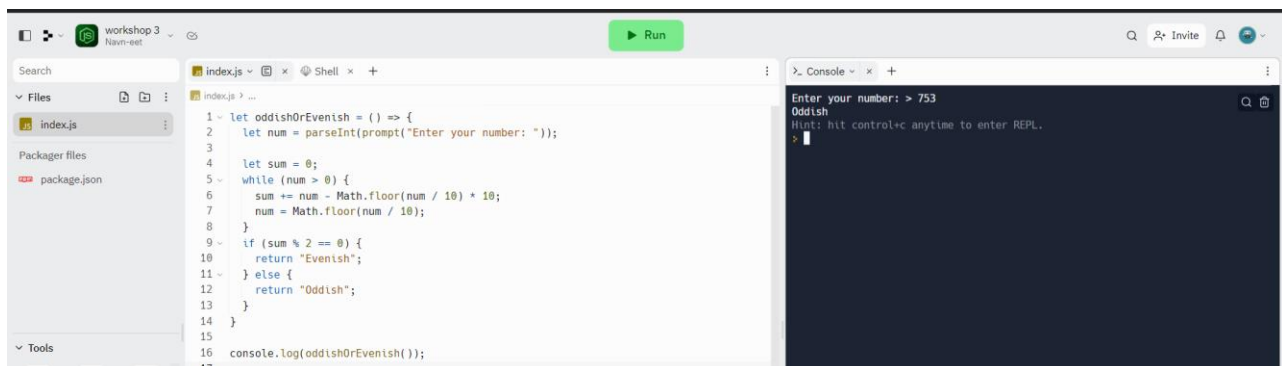


The screenshot shows a VS Code editor with a file named `index.js` containing the following code:

```
1 let oddishOrEvenish = () => {  
2   let num = parseInt(prompt("Enter your number: "));  
3  
4   let sum = 0;  
5   while (num > 0) {  
6     sum += num - Math.floor(num / 10) * 10;  
7     num = Math.floor(num / 10);  
8   }  
9   if (sum % 2 == 0) {  
10    return "Evenish";  
11  } else {  
12    return "Oddish";  
13  }  
14 }  
15  
16 console.log(oddishOrEvenish());  
17
```

The console output shows:

```
Enter your number: > 1232  
Evenish  
Hint: hit control+c anytime to enter REPL.
```



The screenshot shows the same VS Code editor with the same code as above. The console output now shows:

```
Enter your number: > 753  
Oddish  
Hint: hit control+c anytime to enter REPL.
```

LEFT SHIFT BY POWERS OF TWO

The left shift operation is similar to multiplication by powers of two.

Sample calculation using the left shift operator (\ll):

$$10 \ll 3 = 10 * 2^3 = 10 * 8 = 80$$

$$-32 \ll 2 = -32 * 2^2 = -32 * 4 = -128$$

$$5 \ll 2 = 5 * 2^2 = 5 * 4 = 20$$

Write a function that mimics (without the use of \ll) the left shift operator and returns the result from the two given integers.

Examples

`shiftToLeft(5, 2)` \rightarrow 20

`shiftToLeft(10, 3)` \rightarrow 80

`shiftToLeft(-32, 2)` \rightarrow -128

`shiftToLeft(-6, 5)` \rightarrow -192

`shiftToLeft(12, 4)` \rightarrow 192

`shiftToLeft(46, 6)` \rightarrow 2944

Notes

- There will be no negative values for the second parameter y .
- This challenge is more like recreating the left shift operation, thus, the use of the operator directly is prohibited.
- Alternatively, you can solve this challenge via recursion.



```
1 //LEFT SHIFT BY POWERS OF TWO
2
3 function shiftToLeft(x, y) {
4   var result = x;
5   for (var i = 0; i < y; i++) {
6     result *= 2;
7   }
8   return result;
9 }
10
11 var x = parseInt(prompt("Enter the value of x:"));
12 var y = parseInt(prompt("Enter the value of y:"));
13 var result = shiftToLeft(x, y);
14 console.log(result);
15
16 function shiftToLeft(x, y) {
17   if (y === 0) {
18     return x;
19   } else {
20     return shiftToLeft(x * 2, y - 1);
21   }
22 }
```

Enter the value of x: 4
Enter the value of y: 7
512
Hint: hit control+c anytime to enter REPL.

CONVERT A NUMBER TO BASE-2

Create a function that returns a base-2 (binary) representation of a base-10 (decimal) string number. To convertis simple: ((2) means base-2 and (10) means base-10)

$010101001_2 = 1 + 8 + 32 + 128$.

Going from rightto left,the value ofthe most right bitis 1, now from that every bitto the left will be x2. The values of an 8 bit binary number are (256, 128, 64, 32, 16, 8, 4, 2, 1).

Examples

`binary(1) → "1"`

`// 1*1 = 1 binary(5)`

`→ "101"`

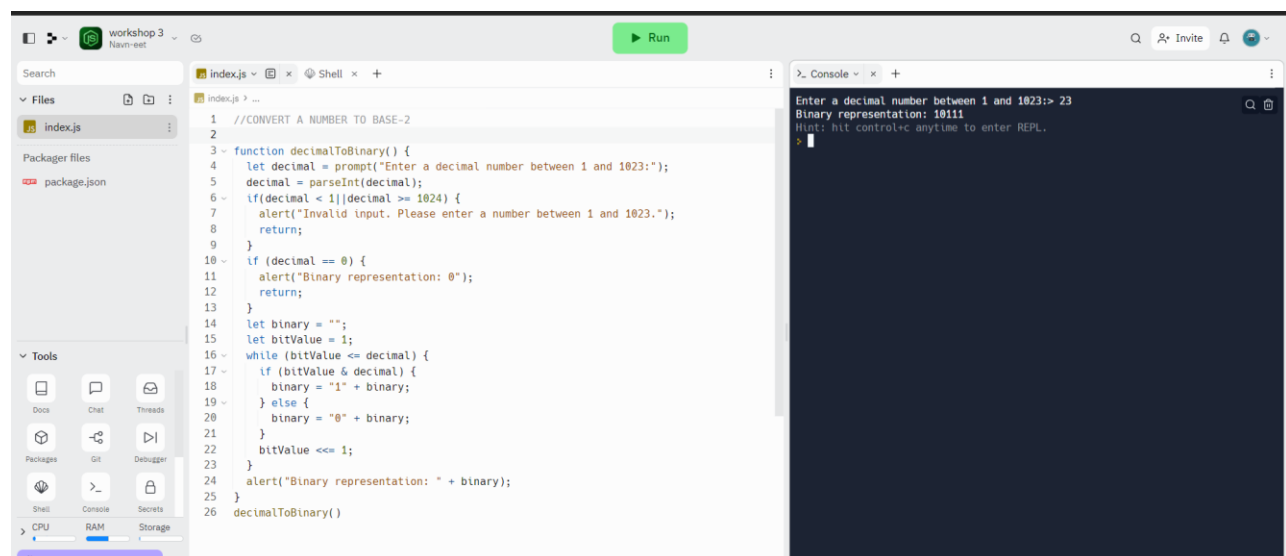
`// 1*1 + 1*4 = 5`

`binary(10) → "1010"`

`// 1*2 + 1*8 = 10`

Notes

- Numbers will always be below 1024 (notincluding 1024).
- The && operator could be useful.
- The strings will always go to the length at which the mostleft bit's value gets bigger than the number in decimal.
- If a binary conversion for 0 is attempted, return "0".



GUESSING GAME

Generate a random number (do research) and store it in a variable. Write a program to take input from the user and tell them whether their guessed number is correct, greater or lesser than the original number. $(100 - \text{number of guesses})$ is the score of user. The program is expected to terminate once the number is guessed. Number should be between 1 – 100.

Example:

Random number generated by computer: 54

User input: 34

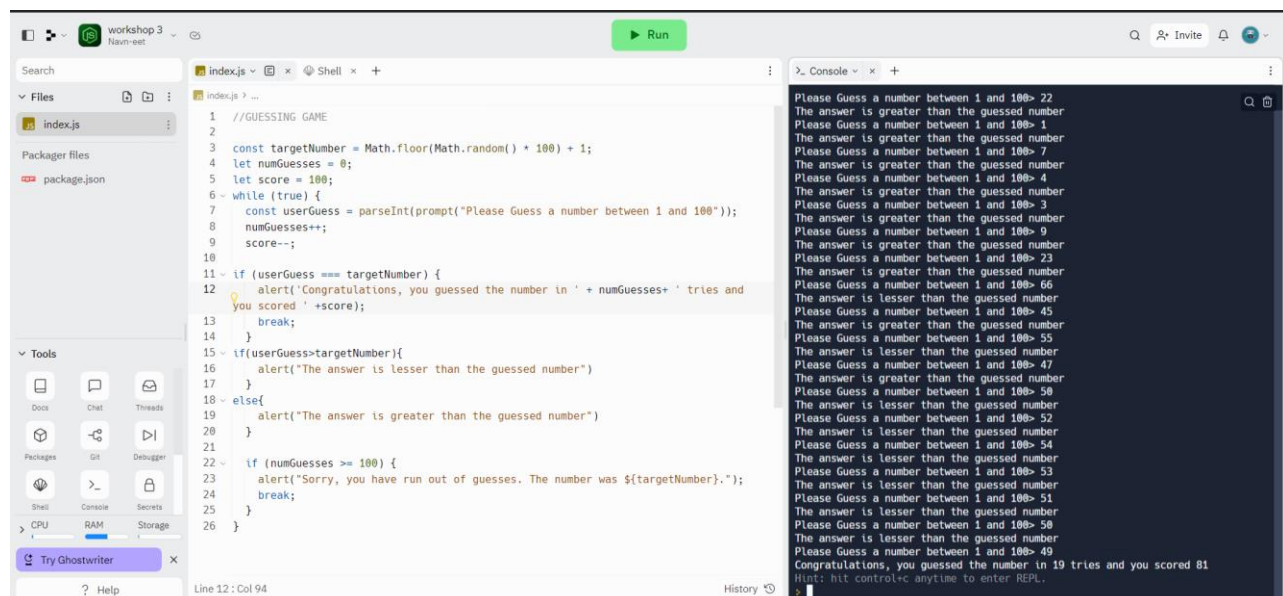
// lesser than original number

User input: 67

// greater than original number

User input: 54

// congratulations!!! The number you guessed matched the original number. Your score is 97!



The screenshot shows a web-based IDE with a file explorer on the left, a code editor in the center, and a console on the right. The code in the editor is a JavaScript program for a guessing game. The console shows the game's execution, including prompts for guesses, feedback messages, and the final score.

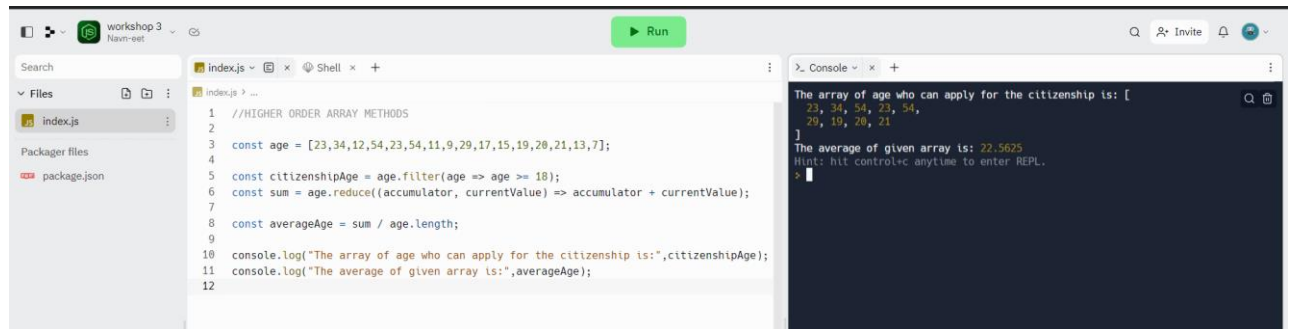
```
1 //GUESSING GAME
2
3 const targetNumber = Math.floor(Math.random() * 100) + 1;
4 let numGuesses = 0;
5 let score = 100;
6 while (true) {
7   const userGuess = parseInt(prompt("Please Guess a number between 1 and 100"));
8   numGuesses++;
9   score--;
10
11   if (userGuess === targetNumber) {
12     alert("Congratulations, you guessed the number in " + numGuesses + " tries and you scored " + score);
13     break;
14   }
15   if (userGuess > targetNumber) {
16     alert("The answer is lesser than the guessed number")
17   }
18   else {
19     alert("The answer is greater than the guessed number")
20   }
21
22   if (numGuesses >= 100) {
23     alert("Sorry, you have run out of guesses. The number was " + targetNumber);
24     break;
25   }
26 }
```

The console output shows the game's execution, including prompts for guesses, feedback messages, and the final score. The user guessed 22, 1, 7, 4, 3, 9, 23, 66, 45, 55, 47, 50, 52, 54, 53, 51, 50, 49, and finally 81, which was the correct number. The final score was 81.

HIGHER ORDER ARRAY METHODS

Const age = [23,34,12,54,23,54,11,9,29,17,15,19,20,21,13,7]

- Filter the array of age who can apply for citizenships
- Find the average age of a given array



```
1 //HIGHER ORDER ARRAY METHODS
2
3 const age = [23,34,12,54,23,54,11,9,29,17,15,19,20,21,13,7];
4
5 const citizenshipAge = age.filter(age => age >= 18);
6 const sum = age.reduce((accumulator, currentValue) => accumulator + currentValue);
7
8 const averageAge = sum / age.length;
9
10 console.log("The array of age who can apply for the citizenship is:",citizenshipAge);
11 console.log("The average of given array is:",averageAge);
12
```

The array of age who can apply for the citizenship is: [23, 34, 54, 23, 54, 29, 19, 20, 21]

The average of given array is: 22.5625

Hint: hit control+c anytime to enter REPL.

Const companies = [

- { name: "ABC", category: "Finance", start: 1981, end: 2004 },
- { name: "XYZ", category: "Retail", start: 1991, end: 20012 },
- { name: "DGF", category: "Finance", start: 1976, end: 2008 },
- { name: "LFT", category: "Retail", start: 1971, end: 1979 },
- { name: "MND", category: "Retail", start: 1995, end: 2010 },
- { name: "HCK", category: "Technology", start: 1987, end: 2011 },
- { name: "BMC", category: "Technology", start: 1989, end: 2009 },
- { name: "TIC", category: "Retail", start: 1993, end: 2005 },
- { name: "NAC", category: "Technology", start: 1991, end: 2010 },
- { name: "ITC", category: "Finance", start: 1998, end: 2016 }

];

- Filter the retail companies
- Get the 80s companies from the array
- Get the companies that lasted for 10 or more years

workshop 3
Naam-test

Run

Search

Files

index.js

Package files

package.json

Tools

Docs

Chat

Threads

Packages

Git

Debugger

Shell

Console

Secrets

CPU

RAM

Storage

index.js

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Console

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

const companies = [

{ name: "ABC", category: "Finance", start: 1981, end: 2004 },

{ name: "XYZ", category: "Retail", start: 1991, end: 2012 },

{ name: "DGF", category: "Finance", start: 1976, end: 2008 },

{ name: "LFT", category: "Retail", start: 1971, end: 1979 },

{ name: "MMD", category: "Retail", start: 1995, end: 2010 },

{ name: "HCK", category: "Technology", start: 1987, end: 2011 },

{ name: "BMC", category: "Technology", start: 1989, end: 2009 },

{ name: "TIC", category: "Retail", start: 1993, end: 2005 },

{ name: "NAC", category: "Technology", start: 1991, end: 2010 },

{ name: "ITC", category: "Finance", start: 1998, end: 2016 }

];

// Filter the retail companies

const retailCompanies = companies.filter(company => company.category === "Retail");

// Get the 80s companies from the array

const eightiesCompanies = companies.filter(company => company.start >= 1980 &&

company.start < 1990);

// Get the companies that lasted for 10 or more years

const tenYearCompanies = companies.filter(company => (company.end - company.start)

>= 10);

console.log(retailCompanies);

console.log(eightiesCompanies);

console.log(tenYearCompanies);

Hit: hit control+c anytime to enter REPL.

>

[

{ name: "XYZ", category: "Retail", start: 1991, end: 2012 },

{ name: "LFT", category: "Retail", start: 1971, end: 1979 },

{ name: "MMD", category: "Retail", start: 1995, end: 2010 },

{ name: "TIC", category: "Retail", start: 1993, end: 2005 }

]

[

{ name: "ABC", category: "Finance", start: 1981, end: 2004 },

{ name: "HCK", category: "Technology", start: 1987, end: 2011 },

{ name: "BMC", category: "Technology", start: 1989, end: 2009 }

]

[

{ name: "ABC", category: "Finance", start: 1981, end: 2004 },

{ name: "XYZ", category: "Retail", start: 1991, end: 2012 },

{ name: "DGF", category: "Finance", start: 1976, end: 2008 },

{ name: "MMD", category: "Retail", start: 1995, end: 2010 },

{ name: "HCK", category: "Technology", start: 1987, end: 2011 },

{ name: "BMC", category: "Technology", start: 1989, end: 2009 },

{ name: "TIC", category: "Retail", start: 1993, end: 2005 },

{ name: "NAC", category: "Technology", start: 1991, end: 2010 },

{ name: "ITC", category: "Finance", start: 1998, end: 2016 }

]

