```
'use strict';
// Lecture 46: Iteration-  The for Loop

// console.log('Lifting weights repetition 1 🏋');
// console.log('Lifting weights repetition 2 🏋');
// console.log('Lifting weights repetition 3 🏋');
// console.log('Lifting weights repetition 4 🏋');
// console.log('Lifting weights repetition 5 🏋');
// console.log('Lifting weights repetition 6 🏋');
// console.log('Lifting weights repetition 7 🏋');
// console.log('Lifting weights repetition 8 🏋');
// console.log('Lifting weights repetition 9 🏋');
// console.log('Lifting weights repetition 10 🏋');

// for loop keeps running while condition is TRUE
for (let rep = 1; rep <= 30; rep++) {
  console.log(`Lifting weights repetition ${rep} 🏋`);
}
// for loop syntax
for (init ; condition ; increment)
```

```
{
    statements
}
////////////////////////////////////////
// Lecture 47: Looping Arrays, Breaking
  and Continuing
const jonas = [
  'Jonas',
  'Schmedtmann',
  2037 - 1991,
  'teacher',
  ['Michael', 'Peter', 'Steven'],
  true
];
const types = [];

// console.log(jonas[0])
// console.log(jonas[1])
// ...
// console.log(jonas[4])
// jonas[5] does NOT exist

// Looping through an array
// Remember arrays are zero based.
for (let i = 0; i < jonas.length; i++) {
  // Reading from jonas array
  console.log(jonas[i], typeof jonas[i]);

  // Filling types array
  // types[i] = typeof jonas[i];
  types.push(typeof jonas[i]);
}
```

```
}

console.log(types);

const years = [1991, 2007, 1969, 2020];
const ages = [];

// Remember the array is zero based.
for (let i = 0; i < years.length; i++) {
  ages.push(2037 - years[i]);
}
console.log(ages);

// continue and break
console.log('--- ONLY STRINGS ---')
for (let i = 0; i < jonas.length; i++) {
    // Note that the typeof operator
      returns a string.
  if (typeof jonas[i] !== 'string')
    continue;
// We only want to log strings to the
  console.
  console.log(jonas[i], typeof jonas[i]);
}

console.log('--- BREAK WITH NUMBER ---')
for (let i = 0; i < jonas.length; i++) {
  if (typeof jonas[i] === 'number')
    break; // this will stop the
    execution of the for loop the moment
    the first number is encountered!
  console.log(jonas[i], typeof jonas[i]);
}
```

```
77  }
78
79
80  ////////////////////////////////////////
81  // Lecture 48: Looping Backwards and
      Loops in Loops
82  const jonas = [
83    'Jonas',
84    'Schmedtmann',
85    2037 - 1991,
86    'teacher',
87    ['Michael', 'Peter', 'Steven'],
88    true
89  ];
90
91  // 0, 1, ..., 4
92  // 4, 3, ..., 0
93
94  for (let i = jonas.length - 1; i >= 0; i--
      ) {
95    console.log(i, jonas[i]);
96  }
97  // We want to do 3 exercises, and 5 reps
      per exercise
98  for (let exercise = 1; exercise < 4;
      exercise++) {
99    console.log(`-------- Starting exercise
        ${exercise}`);
100   // loop within loop!
101   for (let rep = 1; rep < 6; rep++) {
102     console.log(`Exercise ${exercise}:
          Lifting weight repetition ${rep}
```

```
      🏋️`);
  }
}
////////////////////////////////////
// Lecture 49: The while Loop
for (let rep = 1; rep <= 10; rep++) {
  console.log(`Lifting weights repetition
    ${rep} 🏋️`);
}

let rep = 1; // note that the rep
  variable above is already out of scope
  - so it can be redefined here.
// The while loop is more versatile than
  the for loop, since the while loop
  doesn't need a counter.
while (rep <= 10) {
  console.log(`WHILE: Lifting weights
    repetition ${rep} 🏋️`);
  rep++;
}
// Application: Rolling a dice until you
  get a value 6
// Math.random() will create a number
  between 0 and 1.
// Math.trunc() will get rid of the
  decimal.
let dice = 0; // initialise this to 0
  first, since 0 is an impossible value
// while dice not equal to 6
while (dice !== 6) {
  console.log(`You rolled a ${dice}`);
```

```
    dice = Math.trunc(Math.random() * 6) +
      1;
    if (dice === 6)
    {
        console.log('Loop is about to
          end...');
        break;
    }
}
```