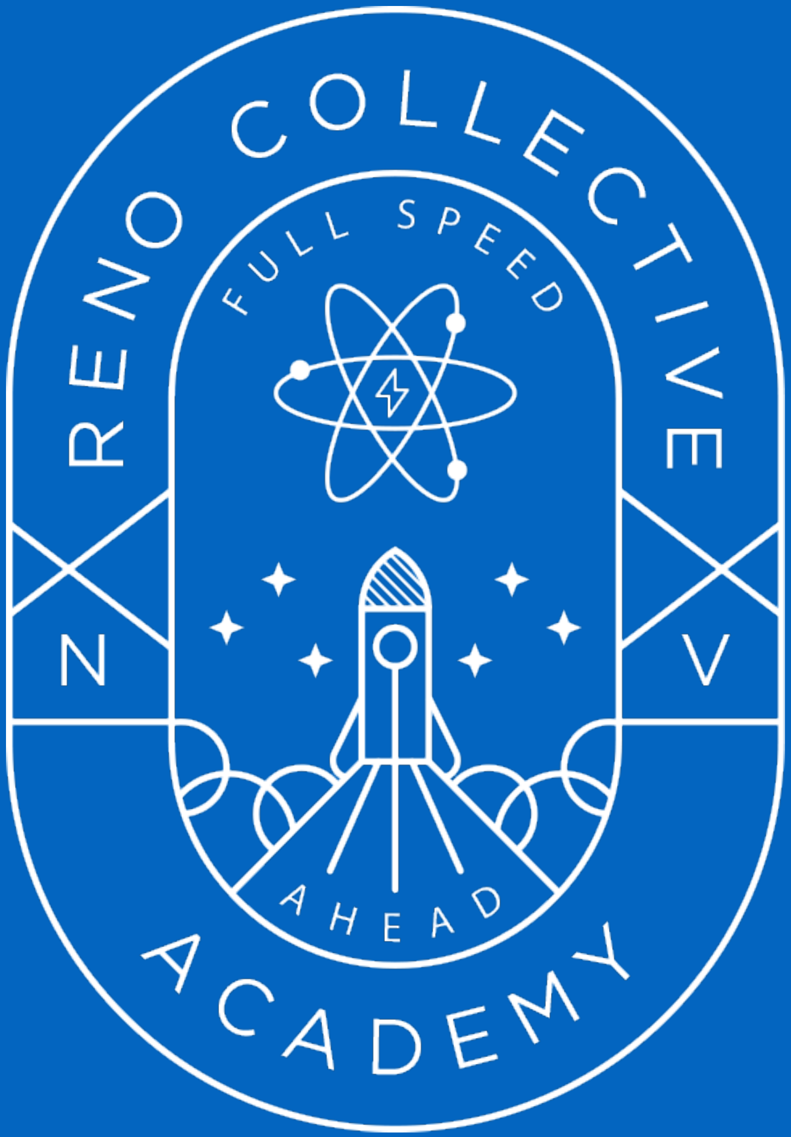
RENO COLLECTIVE
FULL SPEED
N
V
AHEAD
ACADEMY

Leveling up our Javascript

Install Node

https://nodejs.org

Arrays

```
var cities = ['RNO', 'LAS', 'NYC'];

var testScores = [70, 80, 90, 100];
```

Using an Index

```
var cities = ['RNO', 'LAS', 'NYC'];

               0      1      2
```

```
cities[0] // RNO
cities[1] // LAS
cities[2] // NYC
```

Strings are arrays of characters

```
var name = 'Colin';
            01234
```

```
name[0] // C
name[1] // o
name[2] // l
```

Access & Assigment

Creating an empty array named ‘cities’

```
var cities = [];
```

```
cities[0] = 'Reno';
cities[1] = 'Las Vegas';
console.log(cities);
```

```
> [ 'Reno', 'Las Vegas' ]
```

```
console.log(cities[0]);
```

```
> Reno
```

array.push()

Let’s create an empty array of names.

```
var names = [];
```

We can use **push** to add something to the end of the array.

```
names.push('Colin');
names.push('Josh');

console.log(names);
```

```
> [ 'Colin', 'Josh' ]
```

array.pop()

We can use **pop** to remove the last item from the array and return it to the caller.

```
var myFish = ['angel', 'clown', 'mandarin',
'sturgeon'];

var popped = myFish.pop();
```

array.pop()

```
var myFish = ['angel', 'clown', 'mandarin',
'sturgeon'];

console.log(myFish);
// ['angel', 'clown', 'mandarin', 'sturgeon']

var popped = myFish.pop();

console.log(myFish);
// ['angel', 'clown', 'mandarin' ]

console.log(popped);
// 'sturgeon'
```

Useful Array Methods

```
var cities = ['RNO', 'LAS', 'NYC'];
```

```
cities.sort();

cities.length;
cities.indexOf('LAS');
```

indexOf

```
var cities = ['RNO', 'LAS', 'NYC'];
```

```
if(cities.indexOf('LAS') === -1) {
 console.log('City does not exist');
} else {
 console.log('That city exists');
}
```

Using Arrays

```
var cities = ['RNO', 'LAS', 'NYC'];

      var i = 0;
      cities[i];     // ??

      cities[1];     // ??

      cities.pop();  // ??

      cities.length; // ??
```

Iteration

For Loop

```
for(var i = 0; i < 10; i++) {
   // your code here
   console.log(i);
}
```

For Loop & Arrays

```
var cities = ['RNO', 'LAS', 'NYC'];

for(var i = 0; i < cities.length; i++) {

    console.log(cities[i]);

}
```

While Loop

```
while (condition) {
    //code block to be executed
}
```

While Loop + Arrays

```
var cities = ['RNO', 'LAS', 'NYC'];
```

```
var i = 0;
while (i < cities.length) {
  console.log(cities[i]);
  i++;
}
```

Objects

Let’s create an empty object.

```
var flightBooking = {};
```

Object Properties

```
var flightBooking = {
 departingCity : 'RNO',
 destinationCity : 'LAS',
 price : 79.00
}
```

```
var flightBooking = {
 'departingCity' : 'RNO',
 'destinationCity' : 'LAS',
 'price' : 79.00
}
```

Property Assignment

```
 var flightBooking = {
  departingCity : 'RNO',
  destinationCity : 'LAS',
  price : 79.00
 }

flightBooking['destinationCity'] = 'RNO';

flightBooking.destinationCity = 'RNO';
```

Accessing Properties

```
var cityAirportCodes = {
 'RNO' : 'Reno',
 'LAS' : 'Las Vegas',
 'SFO' : 'San Francisco'
};

console.log(cityAirportCodes['RNO']);

console.log(cityAirportCodes['LAS']);
```

Using a Variable

```
var cityAirportCodes = {
 'RNO' : 'Reno',
 'LAS' : 'Las Vegas',
 'SFO' : 'San Francisco'
};

var destinationCity = 'RNO';

console.log(cityAirportCodes[destinationCity]);
```

Property Values

```
var ourCar = {
  make: 'Toyota',
  model: '4Runner',
  features: ['power windows', 'wifi', 'gps'],
  style: { color: 'green', racingStrip: false }
};
```

Array of Objects

```
var cars = [
  { make: 'Toyota', model: '4Runner' },
  { make: 'Tesla', model: 'S' },
  { make: 'Subaru', model: 'Forester' },
 ];
```

Accessing objects in arrays

```
var cars = [
  { make: 'Toyota', model: '4Runner' },
  { make: 'Tesla', model: 'S' },
  { make: 'Subaru', model: 'Forester' },
 ];
```

```
console.log(cars[0].make);

console.log(cars[1]['model'];

cars[2].model = 'Outback';
```

array.join()

```
var words = ['Hello', 'World!'];


var sentence = words.join(' ');

    console.log(sentence);
```

```
> 'Hello World!'
```

Object Methods

```
var ourCar = {
  make: 'Toyota',
  model: '4Runner',
  features: ['power windows', 'wifi', 'gps'],
  style: { color: 'green', racingStrip: false },
  start : function() {
    console.log('Started car');
  }
};

ourCar.start();

ourCar.features[2];          //gps
ourCar.color.racingStripe    //false
ourCar.model                 //4Runner
```

Another way to write functions

```
var start = function() {
   console.log(‘start car’);
};
```

Arguments Keyword

```
var addTwo = function(num1, num2) {
  return num1 + num2;
}
```

Arguments Keyword

```
var add = function() {
  console.log(arguments);
}
```

```
var add = function() {
  var sum = 0;
  for(var i = 0; i < arguments.length; i++) {
    sum += arguments[i];
  }
  return sum;
}
```

More Helpful Methods & Operators

```
array.join();

array.split();

% modulus
```

array.split()

```
var sentence = 'Hello World!';

var words = sentence.split(' ');

console.log(words);
```

```
> ['Hello', 'World!']
```

modulus operator %

```
if( 10 % 2 ) {
  console.log('Odd number');
else {
  console.log('Even number');

}
```