

COMP2140 – Web/Mobile Programming

Lecture Week 3: Recursion vs Iteration & Fetch API

Dr Aneesha Bakharia (Senior Lecturer, EECS)
Email: a.bakharia1@uq.edu.au

Contents

-
- 01 Course Updates

 - 02 JavaScript Object Notation (JSON)

 - 03 Processing, Filtering and Joining JSON with Iteration

 - 04 Recursion

 - 05 Getting JSON from an API

 - 06 Fetch API and Async & Await
-

Course Updates

Issue/Feedback	Change
Lectures not live streamed	Live stream available – accessed via Lecture Recordings on Blackboard Course site
Upcoming Ekka Public Holiday – What's happening with Contacts	<p>Two replacement Contacts will be scheduled for the Tuesday</p> <ul style="list-style-type: none">• 12:00 – 15:30• 14:00 – 17:30 <p>Rooms will be confirmed in the next Blackboard Announcement</p>

JavaScript Object Notation

JavaScript Object Notation (JSON)

```
{  
    "student_id": "123456",  
    "student_firstname": "John",  
    "student_surname": "Doe"  
}
```

- JSON is a lightweight data interchange format that's easy for humans to read and write and easy for machines to parse and generate.
- The structure of JSON includes two types of structures:
 - a collection of key/value pairs and an ordered list of values (array)
- The JSON format is text only and can be easily sent to and from a server (post to an API and get from an API)

JavaScript Object Notation (JSON)

```
{  
    "student_id": "123456",  
    "student_firstname": "John",  
    "student_surname": "Doe"  
}
```

- All keys and string values are written in double quotes
- A colon separates each key from its value.
- The entire set of key/value pairs is enclosed in curly braces {}.

JavaScript Object Notation (JSON)

```
{  
  "students": [  
    {  
      "student_id": "001",  
      "student_firstname": "John",  
      "student_surname": "Doe"  
    },  
    {  
      "student_id": "002",  
      "student_firstname": "Jane",  
      "student_surname": "Smith"  
    },  
    {  
      "student_id": "003",  
      "student_firstname": "Bob",  
      "student_surname": "Johnson"  
    }  
]  
}
```

```
const students = [  
  {  
    "student_id": "001",  
    "student_firstname": "John",  
    "student_surname": "Doe"  
  },  
  {  
    "student_id": "002",  
    "student_firstname": "Jane",  
    "student_surname": "Smith"  
  },  
  {  
    "student_id": "003",  
    "student_firstname": "Bob",  
    "student_surname": "Johnson"  
  }  
];
```

Storing Arrays of Objects

JSON can be Nested

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 85},
      {"quiz_name": "History Quiz", "score": 90}
    ]
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 92},
      {"quiz_name": "History Quiz", "score": 88}
    ]
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 78},
      {"quiz_name": "History Quiz", "score": 80}
    ]
  }
];
```

Using Iteration to Process JSON

Printing all student_id's

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe"
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith"
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson"
  }
];
```

Loop - Imperative

```
for (let i = 0; i < students.length; i++) {
  console.log(students[i].student_id);
}
```

.forEach - Functional

```
students.forEach((student) => {
  console.log(student.student_id);
});
```

Filter by a key value

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe"
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith"
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson"
  }
];
```

Loop - Imperative

```
let filteredStudents = [];

for (let i = 0; i < students.length; i++) {
  if (students[i].student_id === "001") {
    filteredStudents.push(students[i]);
  }
}

console.log(filteredStudents);
```

.filter - Functional

```
const filteredStudents = students
  .filter(student => student.student_id === "001");
console.log(filteredStudents);
```

Get an array of studentIds

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe"
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith"
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson"
  }
];
```

Loop - Imperative

```
let studentIds = [];

for (let i = 0; i < students.length; i++) {
  studentIds.push(students[i].student_id);
}

console.log(studentIds);
```

.map - Functional

```
const studentIds = students.map(student => student.student_id);
console.log(studentIds);
```

Do all students have a student_id

Loop - Imperative

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe"
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith"
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson"
  }
];
```

```
let allHaveIds = true;

for (let i = 0; i < students.length; i++) {
  if (students[i].student_id === "") {
    allHaveIds = false;
    break;
  }
}

console.log(allHaveIds);
```

.every - Functional

```
const allHaveIds = students.every(student => student.student_id !== "");
console.log(allHaveIds);
```

Average Math Quiz scores

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 85},
      {"quiz_name": "History Quiz", "score": 90}
    ]
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 92},
      {"quiz_name": "History Quiz", "score": 88}
    ]
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 78},
      {"quiz_name": "History Quiz", "score": 80}
    ]
  }
];
```

Loop - Imperative

```
let sumOfMathQuizScores = 0;
let countOfMathQuizScores = 0;

for (let i = 0; i < students.length; i++) {
  for (let j = 0; j < students[i].quiz_scores.length; j++) {
    if (students[i].quiz_scores[j].quiz_name === "Math Quiz") {
      sumOfMathQuizScores += students[i].quiz_scores[j].score;
      countOfMathQuizScores++;
    }
  }
}

const avgMathQuizScore = sumOfMathQuizScores / countOfMathQuizScores;

console.log(avgMathQuizScore);
```

Average Math Quiz scores

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 85},
      {"quiz_name": "History Quiz", "score": 90}
    ]
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 92},
      {"quiz_name": "History Quiz", "score": 88}
    ]
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson",
    "quiz_scores": [
      {"quiz_name": "Math Quiz", "score": 78},
      {"quiz_name": "History Quiz", "score": 80}
    ]
  }
];
```

.reduce - Functional

```
const mathQuizScores = students
  .map(student => student.quiz_scores
    .filter(quiz => quiz.quiz_name === "Math Quiz")
    .map(quiz => quiz.score))
  .flat();

const avgMathQuizScore = mathQuizScores.reduce((sum, score) => sum + score, 0) / mathQuizScores.length;

console.log(avgMathQuizScore);
```

1. We first use map to create an array that contains, for each student, the scores of the quizzes named "Math Quiz". This results in an array of arrays.
2. We then use flat to flatten this array of arrays into a single array that contains all of the "Math Quiz" scores.
3. We finally use reduce to calculate the sum of these scores, and divide by the length of the array (i.e., the number of scores) to get the average.

Join related objects

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe"
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith"
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson"
  }
];
```

```
const addresses = [
  {
    student_id: "001",
    street: "123 Main St",
    postcode: "12345",
    state: "CA"
  },
  {
    student_id: "002",
    street: "456 Market St",
    postcode: "67890",
    state: "NY"
  },
  {
    student_id: "003",
    street: "789 Broadway St",
    postcode: "11223",
    state: "TX"
  }
];
```

Join related objects

```
const students = [
  {
    "student_id": "001",
    "student_firstname": "John",
    "student_surname": "Doe"
  },
  {
    "student_id": "002",
    "student_firstname": "Jane",
    "student_surname": "Smith"
  },
  {
    "student_id": "003",
    "student_firstname": "Bob",
    "student_surname": "Johnson"
  }
];
```

```
const addresses = [
  {
    "student_id": "001",
    "street": "123 Main St",
    "postcode": "12345",
    "state": "CA"
  },
  {
    "student_id": "002",
    "street": "456 Market St",
    "postcode": "67890",
    "state": "NY"
  },
  {
    "student_id": "003",
    "street": "789 Broadway St",
    "postcode": "11223",
    "state": "TX"
  }
];
```

```
[
  {
    "student_id": '001',
    "student_firstname": 'John',
    "student_surname": 'Doe',
    "postcode": '12345'
  },
  {
    "student_id": '002',
    "student_firstname": 'Jane',
    "student_surname": 'Doe',
    "postcode": '67890'
  },
  {
    "student_id": '003',
    "student_firstname": 'Mary',
    "student_surname": 'Johnson',
    "postcode": '11223'
  }
]
```

Join related objects

```
const studentsWithAddresses = students.map(student => {
  const address = addresses.find(addr => addr.student_id === student.student_id);
  return {
    student_id: student.student_id,
    student_firstname: student.student_firstname,
    student_surname: student.student_surname,
    postcode: address ? address.postcode : null
  };
);

console.log(studentsWithAddresses);
```

Or using ... (spread operator)

```
const studentsWithAddresses = students.map(student => {
  const address = addresses.find(addr => addr.student_id === student.student_id);
  return {
    ...student,
    postcode: address ? address.postcode : null
  };
);

console.log(studentsWithAddresses);
```

.Find vs .Filter

- filter() returns a new array that includes all elements that pass the test implemented by the provided function. If no elements pass the test, filter() returns an empty array.
- find() returns the first element in the array that passes the test implemented by the provided function. If no elements pass the test, find() returns undefined.

```
const numbers = [1, 2, 3, 4, 5];

// filter all even numbers
const evenNumbers = numbers.filter(num => num % 2 === 0);
console.log(evenNumbers); // prints: [2, 4]

// find the first even number
const firstEvenNumber = numbers.find(num => num % 2 === 0);
console.log(firstEvenNumber); // prints: 2
```

Can we make our own SQL like join method?

```
/**  
 * @param {Array} array1 First array of objects.  
 * @param {Array} array2 Second array of objects.  
 * @param {String} joinOnField Field name on which join has to be performed.  
 * @param {Array} fieldsToMerge Fields from the second array to be merged.  
 */  
function innerJoin(array1, array2, joinOnField, fieldsToMerge) {  
    return array1.map(item1 => {  
        const item2 = array2.find(item2 => item1[joinOnField] === item2[joinOnField]);  
  
        // If matching item found in array2  
        if (item2) {  
            const merged = {};  
            // Merge specified fields from item2  
            for (const field of fieldsToMerge) {  
                if (item2.hasOwnProperty(field)) {  
                    merged[field] = item2[field];  
                }  
            }  
            return {  
                ...item1,  
                ...merged  
            };  
        }  
        return null;  
    }).filter(item => item !== null); // Keep only items that found a match in array2  
}
```

Can we make our own SQL like join method?

```
const array1 = [{id: 1, name: 'Alice'}, {id: 2, name: 'Bob'}, {id: 3, name: 'Charlie'}];
const array2 = [{id: 1, age: 25}, {id: 2, age: 30}, {id: 4, age: 45}];
const joinOnField = 'id';
const fieldsToMerge = ['age'];

const result = innerJoin(array1, array2, joinOnField, fieldsToMerge);

console.log(result);
// Outputs: [{id: 1, name: 'Alice', age: 25}, {id: 2, name: 'Bob', age: 30}]
```

React Functional Component

```
import React from 'react';
const StudentsTable = () => {
  const students = [
    { student_id: "001", student_firstname: "John", student_surname: "Doe" },
    { student_id: "002", student_firstname: "Jane", student_surname: "Doe" },
    { student_id: "003", student_firstname: "Mary", student_surname: "Johnson" }
  ];
  return (
    <table>
      <thead>
        <tr>
          <th>Student ID</th><th>First Name</th><th>Surname</th>
        </tr>
      </thead>
      <tbody>
        {students.map((student, index) => (
          <tr key={index}>
            <td>{student.student_id}</td>
            <td>{student.student_firstname}</td>
            <td>{student.student_surname}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
}
export default StudentsTable;
```

Recursion

What is Recursion?

- Recursion is a process in which a function calls itself
- Each recursive call to a function creates its own scope/environment with its own copy of variables.
- Recursion is a common approach in functional programming and can lead to clean and elegant code.
- However, recursion can also lead to high memory use and stack overflow errors if not managed properly, as each recursive call adds a layer to the call stack.
- Recursive algorithms can often be translated into iterative algorithms, and vice versa.
- From a functional programming perspective, expresses the problem in a declarative manner, specifying what the result is in terms of smaller versions of itself, rather than describing how to compute it.

Calculate a factorial

Loop - Imperative

```
function factorialIterative(n) {  
    let result = 1;  
    for (let i = 2; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

$$n! = n * (n-1) * (n-2) * \dots * 3 * 2 * 1$$

e.g.

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

$$4! = 4 * 3 * 2 * 1 = 24$$

Recursive - Functional

```
function factorialRecursive(n) {  
    if (n === 0) {  
        return 1;  
    } else {  
        return n * factorialRecursive(n - 1);  
    }  
}
```

From a functional programming perspective, recursion aligns more closely with functional programming principles because it doesn't maintain mutable state, unlike the iterative solution which modifies the `result` variable.

Getting JSON from an API

Cats or Dogs

Cats or Dogs

<https://api.thecatapi.com/v1/breeds>

<https://api.thedogapi.com/v1/breeds>

Open API with facts about different dog/cat breeds

<https://api.thedogapi.com/v1/breeds>

JSON	Raw Data	Headers
Save	Copy	Pretty Print
[{"weight": "imperial": "6 - 13", "metric": "3 - 6"}, {"height": "imperial": "9 - 11.5", "metric": "23 - 29"}, {"id": "1", "name": "Affenpinscher", "bred_for": "Small rodent hunting, lapdog", "breed_group": "Toy", "life_span": "10 - 12 years", "temperament": "Stubborn, Curious, Playful, Adventurous, Active, Fun-loving", "origin": "Germany, France", "reference_image_id": "Bj44kxc4X", "image": "Bj44kxc4X", "width": 1600, "height": 1199, "url": "https://cdn2.thedogapi.com/images/Bj44kxc4X.jpg"}, {"weight": "imperial": "50 - 60", "metric": "23 - 27"}, {"height": "imperial": "25 - 27", "metric": "64 - 69"}, {"id": "2", "name": "Afghan Hound", "bred_for": "Coursing and hunting", "breed_group": "Hound", "life_span": "10 - 13 years", "temperament": "Alert, Clownish, Dignified, Independent, Happy", "origin": "Afghanistan, Iran, Pakistan", "reference_image_id": "My14DXR", "image": "My14DXR", "width": 606, "height": 380, "url": "https://cdn2.thedogapi.com/images/My14DXR.jpg"}, {"weight": "imperial": "50 - 60", "metric": "20 - 30"}, {"height": "imperial": "30 - 38", "metric": "70 - 76"}, {"id": "3", "name": "African Hunting Dog", "bred_for": "Wild pack animal", "life_span": "11 years", "temperament": "Hardworking, Dutiful", "origin": "Africa", "reference_image_id": "Ht4y14DXR", "image": "Ht4y14DXR", "width": 500, "height": 353, "url": "https://cdn2.thedogapi.com/images/Ht4y14DXR.jpg"}, {"weight": "imperial": "44 - 66", "metric": "20 - 30"}, {"height": "imperial": "30 - 38", "metric": "70 - 76"}, {"id": "4", "name": "Airedale Terrier", "bred_for": "Badger, otter hunting", "breed_group": "Terrier", "life_span": "10 - 13 years", "temperament": "Outgoing, Alert, Confident, Intelligent, Courageous", "origin": "United Kingdom, England", "reference_image_id": "1-7cg0ZSH", "image": "1-7cg0ZSH", "width": 645, "height": 430, "url": "https://cdn2.thedogapi.com/images/1-7cg0ZSH.jpg"}, {"weight": "imperial": "50 - 58", "metric": "21 - 23"}, {"height": "imperial": "40 - 65", "metric": "18 - 29"}, {"id": "5", "name": "Akbash Dog", "bred_for": "Sheep guarding", "breed_group": "Working", "life_span": "10 - 12 years", "temperament": "Loyal, Independent, Intelligent, Brave", "origin": "Turkey", "reference_image_id": "26pHT3QK7", "image": "26pHT3QK7", "width": 600, "height": 471, "url": "https://cdn2.thedogapi.com/images/26pHT3QK7.jpg"}, {"weight": "imperial": "65 - 115", "metric": "29 - 52"}, {"height": "imperial": "24 - 28", "metric": "61 - 71"}, {"id": "6", "name": "Akita", "bred_for": "Hunting bears", "breed_group": "Composed, Friendly, Receptive, Courageous", "life_span": "10 - 14 years", "temperament": "Docile, Alert, Responsive", "origin": "Japan", "reference_image_id": "BFRYBuFpm", "image": "BFRYBuFpm", "width": 1280, "height": 853, "url": "https://cdn2.thedogapi.com/images/BFRYBuFpm.jpg"}, {"weight": "imperial": "55 - 90", "metric": "25 - 41"}, {"height": "imperial": "18 - 24", "metric": "40 - 61"}, {"id": "7", "name": "Alapaha Blue Blood Bulldog", "description": "The Alapaha Blue Blood Bulldog is a well-developed, exaggerated bulldog with a broad head and natural drop ears. The prominent muzzle is covered by loose upper lips. The prominent eyes are set well apart. The Alapaha's coat is relatively short and fairly stiff. Preferred colors are blue merle, brown merle, or red merle all trimmed in white or chocolate and white. Also preferred are the glass eyes (blue) or marble eyes (brown and blue mixed in a single eye). The ears and tail are never trimmed or docked. The body is sturdy and very muscular. The well-muscled hips are narrower than the chest. The straight back is as long as the dog is high at the shoulders. The dewclaws are never removed and the feet are cat-like.", "bred_for": "Sled pulling", "breed_group": "Mixed", "life_span": "12 - 13 years", "temperament": "Friendly, Energetic, Loyal, Gentle, Confident", "origin": "USA", "reference_image_id": "HgNpNxG1", "image": "HgNpNxG1", "width": 500, "height": 500, "url": "https://cdn2.thedogapi.com/images/HgNpNxG1.jpg"}, {"weight": "imperial": "65 - 100", "metric": "29 - 45"}, {"height": "imperial": "23 - 25", "metric": "58 - 64"}, {"id": "9", "name": "Alaskan Malamute", "bred_for": "Hauling heavy freight, Sled pulling", "breed_group": "Working", "life_span": "12 - 15 years", "temperament": "Friendly, Affectionate, Devoted, Loyal, Dignified, Playful", "origin": "USA", "reference_image_id": "dw5UwCtIW", "image": "dw5UwCtIW", "width": 1023, "height": 769, "url": "https://cdn2.thedogapi.com/images/dw5UwCtIW.jpg"}, {"weight": "imperial": "60 - 120", "metric": "27 - 54"}, {"height": "imperial": "22 - 27", "metric": "56 - 69"}, {"id": "10", "name": "American Bulldog", "breed_group": "Working", "life_span": "10 - 12 years", "temperament": "Friendly, Assertive, Energetic, Loyal, Gentle, Confident, Dominant", "origin": "USA", "reference_image_id": "pk1AAAdLo", "image": "pk1AAAdLo", "width": 1669, "height": 1377, "url": "https://cdn2.thedogapi.com/images/pk1AAAdLo.jpg"}, {"weight": "imperial": "30 - 150", "metric": "14 - 68"}, {"height": "imperial": "14 - 17", "metric": "36 - 43"}, {"id": "11", "name": "American Bully", "breed_group": "Country", "life_span": "8 - 15 years", "temperament": "Strong Willed, Stubborn, Friendly, Clownish, Affectionate, Loyal, Obedient, Intelligent, Courageous", "origin": "USA", "reference_image_id": "sqQJDtpbY", "image": "sqQJDtpbY", "width": 1024, "height": 683, "url": "https://cdn2.thedogapi.com/images/sqQJDtpbY.jpg"}, {"weight": "imperial": "20 - 40", "metric": "9 - 18"}, {"height": "imperial": "15 - 19", "metric": "38 - 48"}, {"id": "12", "name": "American Eskimo Dog", "breed_group": "Country", "life_span": "12 - 15 years", "temperament": "Friendly, Alert, Reserved, Intelligent, Protective", "origin": "USA", "reference_image_id": "Byjmeyec4m", "image": "Byjmeyec4m", "width": 1000, "height": 802, "url": "https://cdn2.thedogapi.com/images/Byjmeyec4m.jpg"}, {"weight": "imperial": "7 - 10", "metric": "3 - 5"}, {"height": "imperial": "9 - 12", "metric": "23 - 30"}, {"id": "13", "name": "American Eskimo Dog (Minature)", "breed_group": "Companionship", "life_span": "13 - 18 years", "temperament": "Non-Sporting", "origin": "USA", "reference_image_id": "gn8GLR6E6", "image": "gn8GLR6E6", "width": 730, "height": "487", "url": "https://cdn2.thedogapi.com/images/gn8GLR6E6.jpg"}, {"weight": "imperial": "65 - 75", "metric": "29 - 34"}, {"height": "imperial": "21 - 28", "metric": "53 - 71"}, {"id": "14", "name": "American Foxhound", "breed_group": "Hound", "life_span": "8 - 15 years", "temperament": "Kind, Sweet-Tempered, Loyal, Independent, Intelligent, Loving", "origin": "USA", "reference_image_id": "S14n1x9N0", "image": "S14n1x9N0", "width": 3000, "height": 2400, "url": "https://cdn2.thedogapi.com/images/S14n1x9N0.jpg"}, {"weight": "imperial": "30 - 60", "metric": "14 - 27"}, {"height": "imperial": "17 - 21", "metric": "43 - 53"}, {"id": "15", "name": "American Pit Bull Terrier", "breed_group": "Country", "life_span": "12 - 15 years", "temperament": "Strong Willed, Stubborn, Friendly, Clownish, Affectionate, Loyal, Obedient, Intelligent, Courageous", "origin": "USA", "reference_image_id": "HkC3lgNm", "image": "HkC3lgNm", "width": 3000, "height": 2440, "url": "https://cdn2.thedogapi.com/images/HkC3lgNm.png"}, {"weight": "imperial": "50 - 60", "metric": "23 - 27"}, {"height": "imperial": "17 - 19", "metric": "43 - 48"}, {"id": "16", "name": "American Staffordshire Terrier", "breed_group": "Country", "life_span": "12 - 15 years", "temperament": "Tenacious, Friendly, Devoted, Loyal, Attentive, Courageous", "origin": "USA", "reference_image_id": "JlIakg4cm", "image": "JlIakg4cm", "width": 500, "height": 450, "url": "https://cdn2.thedogapi.com/images/JlIakg4cm.jpg"}, {"weight": "imperial": "25 - 45", "metric": "11 - 20"}, {"height": "imperial": "15 - 18", "metric": "38 - 46"}, {"id": "17", "name": "American Water Spaniel", "breed_group": "Sporting", "life_span": "10 - 12 years", "temperament": "Friendly, Energetic, Obedient, Intelligent, Protective, Trainable", "origin": "USA", "reference_image_id": "SkmrJ19QV", "image": "SkmrJ19QV", "width": 1600, "height": 1264, "url": "https://cdn2.thedogapi.com/images/SkmrJ19QV.jpg"}, {"weight": "imperial": "80 - 150", "metric": "36 - 68"}, {"height": "imperial": "27 - 29", "metric": "61 - 74"}, {"id": "18", "name": "Anatolian Shepherd Dog", "bred_for": "Livestock herding", "breed_group": "Working", "life_span": "11 - 13 years", "temperament": "Steady, Bold, Independent, Confident, Intelligent, Proud", "origin": "Turkey", "reference_image_id": "BJT0JxNm", "image": "BJT0JxNm", "width": 1216, "height": 1131, "url": "https://cdn2.thedogapi.com/images/BJT0JxNm.jpg"}, {"weight": "imperial": "48 - 55", "metric": "22 - 25"}, {"height": "imperial": "51 - 56"}, {"id": "19", "name": "Appenzeller Sennenhund", "bred_for": "Herd", "breed_group": "Hunting", "life_span": "12 - 14 years", "temperament": "Reliable, Fearless, Energetic, Loyal, Self-assured", "origin": "Switzerland", "reference_image_id": "HNKnxLqEX", "image": "HNKnxLqEX", "width": 400, "height": 600, "url": "https://cdn2.thedogapi.com/images/HNKnxLqEX.jpg"}, {"weight": "imperial": "44 - 62", "metric": "26 - 28"}, {"height": "imperial": "17 - 20", "metric": "42 - 51"}, {"id": "21", "name": "Australian Cattle Dog", "breed_group": "Herding", "life_span": "12 - 14 years", "temperament": "Cautious, Energetic, Loyal, Obedient, Protective, Brave", "origin": "Australia", "reference_image_id": "IBKMy4v1", "image": "IBKMy4v1", "width": 736, "height": 850, "url": "https://cdn2.thedogapi.com/images/IBKMy4v1.jpg"}, {"weight": "imperial": "31 - 41", "metric": "14 - 22"}, {"height": "imperial": "43 - 51", "metric": "17 - 20"}, {"id": "22", "name": "Australian Kelpie", "breed_group": "Herding", "life_span": "10 - 13 years", "temperament": "Friendly, Energetic, Alert, Loyal, Intelligent, Eager", "origin": "Australia", "reference_image_id": "Hyq1ge9V0", "image": "Hyq1ge9V0", "width": 800, "height": 533, "url": "https://cdn2.thedogapi.com/images/Hyq1ge9V0.jpg"}, {"weight": "imperial": "35 - 65", "metric": "16 - 29"}, {"height": "imperial": "18 - 23", "metric": "40 - 58"}, {"id": "23", "name": "Australian Shepherd", "breed_group": "Hunting", "life_span": "12 - 16 years", "temperament": "Good-natured, Affectionate, Intelligent, Active, Protective", "origin": "USA", "reference_image_id": "Bl-llgg4m", "image": "Bl-llgg4m", "width": 1200, "height": 773, "url": "https://cdn2.thedogapi.com/images/Bl-llgg4m.jpg"}, {"weight": "imperial": "14 - 16", "metric": "6 - 7"}, {"height": "imperial": "10 - 11", "metric": "25 - 30"}]		

<https://api.thedogapi.com/v1/breeds>

```
JSON Raw Data Headers
Save Copy Collapse All Expand All | Filter JSON

short and fairly stiff. Preferred colors are blue merle, brown merle, or red merle all trimmed in white or chocolate and
are never trimmed or docked. The body is sturdy and very muscular. The well-muscled hips are narrower than the chest. The
bred_for: "Guarding"
breed_group: "Mixed"
life_span: "12 – 13 years"
history: ""
▼ temperament: "Loving, Protective, Trainable, Dutiful, Responsible"
reference_image_id: "33mJ-V3RX"
▼ image:
  id: "33mJ-V3RX"
  width: 1828
  height: 2065
  url: "https://cdn2.thedogapi.com/images/33mJ-V3RX.jpg"
▼ 7:
  ▼ weight:
    imperial: "38 – 50"
    metric: "17 – 23"
  ▼ height:
    imperial: "23 – 26"
    metric: "58 – 66"
  id: 8
  name: "Alaskan Husky"
  bred_for: "Sled pulling"
  breed_group: "Mixed"
  life_span: "10 – 13 years"
  temperament: "Friendly, Energetic, Loyal, Gentle, Confident"
  reference_image_id: "-HgpNnGXl"
  ▼ image:
    id: "-HgpNnGXl"
    width: 500
    height: 500
    url: "https://cdn2.thedogapi.com/images/-HgpNnGXl.jpg"
```

Fetch API

```
fetch('https://api.thedogapi.com/v1/breeds')
  .then( (response) => {
    console.log("resolved", response);
  })
  .catch( (err) => {
    console.log("rejected", err);
  })
```

Browser Console

```
resolved ▾ Response ⓘ
  body: (...)

  bodyUsed: false
  ▶ headers: Headers {}
  ok: true
  redirected: false
  status: 200
  statusText: ""
  type: "cors"
  url: "https://api.thedogapi.com/v1/breeds"
  ▶ [[Prototype]]: Response
```

- Returns a Promise (which you can think of like a real promise) – at some point it will resolve for a success or reject if there is an error

Fetch API

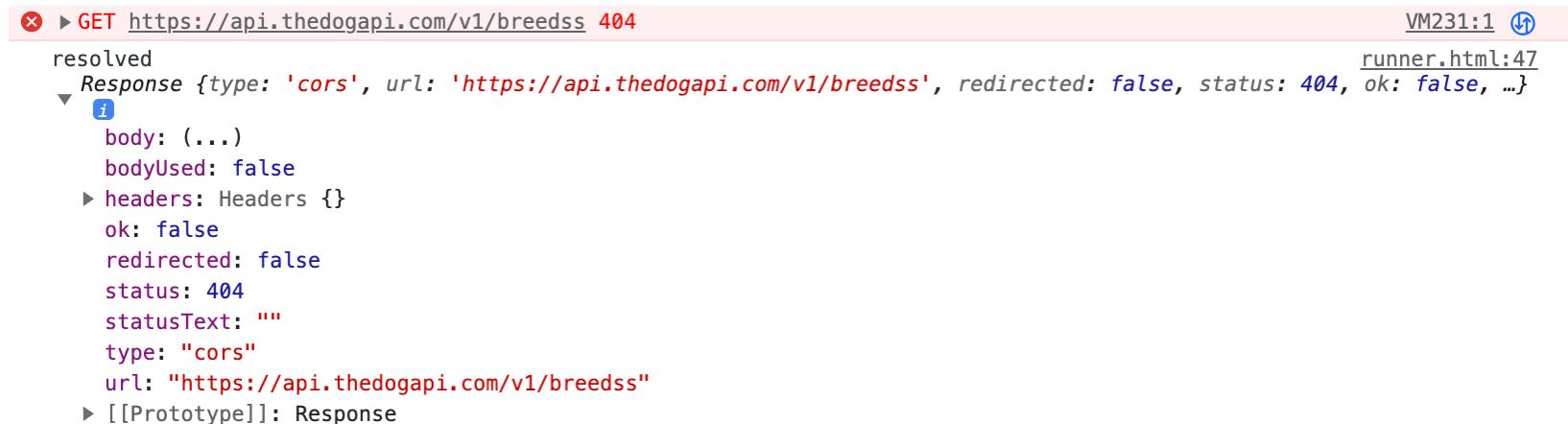
```
fetch('https://api.thedogapi.com/v1/breeds')
  .then( (response) => {
    console.log("resolved", response);
  })
  .catch( (err) => {
    console.log("rejected", err);
  })
```

- What if we have an error in the API endpoint URL?

Fetch API

```
fetch('https://api.thedogapi.com/v1/breedss')
  .then( (response) => {
    console.log("resolved", response);
  })
  .catch( (err) => {
    console.log("rejected", err);
  })
```

- What if we have an error in the API endpoint URL?



- We get a 404 error but get resolved! We will only get an error if the endpoint can't be reached or there is no internet network connection.

Fetch API

```
fetch('https://api.thedogapi.com/v1/breeds')
  .then( (response) => {
    console.log("resolved", response);
  })
  .catch( (err) => {
    console.log("rejected", err);
  })
```

- How do we get the returned JSON?

```
resolved
▼ Response {type: 'cors', url: 'https://api.thedogapi.com/v1/breeds', redirected: false, status: 200, ok: true, ...} ⓘ
  ▼ body: ReadableStream
    locked: false
    ► [[Prototype]]: ReadableStream
    bodyUsed: false
  ► headers: Headers {}
  ok: true
  redirected: false
  status: 200
  statusText: ""
  type: "cors"
  url: "https://api.thedogapi.com/v1/breeds"
  ► [[Prototype]]: Response
```

Fetch API

- How do we get the returned JSON?

```
resolved  
  ▼ Response {  
    type: 'cors', url: 'https://api.thedogapi.com/v1/breeds', redirected: false, status: 200, ok: true, ...} ⓘ  
    ▶ body: ReadableStream  
      locked: false  
    ▶ [[Prototype]]: ReadableStream  
    bodyUsed: false  
  ▶ headers: Headers {}  
  ok: true  
  redirected: false  
  status: 200  
  statusText: ""  
  type: "cors"  
  url: "https://api.thedogapi.com/v1/breeds"  
  ▶ [[Prototype]]: Response  
    ▶ arrayBuffer: f arrayBuffer()  
    ▶ blob: f blob()  
    ▶ body: ReadableStream  
      bodyUsed: (...)  
    ▶ clone: f clone()  
    ▶ formData: f formData()  
    ▶ headers: (...)  
    ▶ json: f json()  
      length: 0  
      name: "json"  
      arguments: (...)  
      ⏷ json.name ..  
    ▶ [[Prototype]]: f ()  
    ▶ [[Scopes]]: Scopes[0]  
    ok: (...)  
    redirected: (...)  
    status: (...)  
    statusText: (...)  
    ▶ text: f text()  
    type: (...)  
    url: (...)  
  ▶ constructor: f Response()  
  Symbol(Symbol.toStringTag): "Response"  
  ▶ get body: f body()  
  ▶ get bodyUsed: f bodyUsed()  
  ▶ get headers: f headers()  
  ▶ get ok: f ok()  
  ▶ get redirected: f redirected()  
  ▶ get status: f status()  
  ▶ get statusText: f statusText()  
  ▶ get type: f type()  
  ▶ get url: f url()  
  ▶ [[Prototype]]: Object
```

Response.json will return the json

Fetch API

- How do we get the returned JSON?

```
fetch('https://api.thedogapi.com/v1/breeds')
  .then( (response) => {
    console.log("resolved", response);
    return response.json(); // returns a promise
  })
  .then( data => {
    console.log(data)
  })
  .catch( (err) => {
    console.log("rejected", err);
  })
```

Fetch API

- How do we get the returned JSON?

```
resolved
▶ Response {type: 'cors', url: 'https://api.thedogapi.com/v1/breeds', redirected: false, status: 200, ok: true, ...} runner.html:47
(172) [{}]
  ↴ [0 ... 99]
    ▶ 0: {weight: {...}, height: {...}, id: 1, name: 'Affenpinscher', bred_for: 'Small rodent hunting, lapdog', ...}
    ▶ 1: {weight: {...}, height: {...}, id: 2, name: 'Afghan Hound', country_code: 'AG', ...}
    ▶ 2: {weight: {...}, height: {...}, id: 3, name: 'African Hunting Dog', bred_for: 'A wild pack animal', ...}
    ▶ 3: {weight: {...}, height: {...}, id: 4, name: 'Airedale Terrier', bred_for: 'Badger, otter hunting', ...}
    ▶ 4: {weight: {...}, height: {...}, id: 5, name: 'Akbash Dog', bred_for: 'Sheep guarding', ...}
    ▶ 5: {weight: {...}, height: {...}, id: 6, name: 'Akita', bred_for: 'Hunting bears', ...}
    ▶ 6: {weight: {...}, height: {...}, id: 7, name: 'Alapaha Blue Blood Bulldog', description: 'The Alapaha Blue Blood Bu...'}
    ▶ 7: {weight: {...}, height: {...}, id: 8, name: 'Alaskan Husky', bred_for: 'Sled pulling', ...}
    ▶ 8: {weight: {...}, height: {...}, id: 9, name: 'Alaskan Malamute', bred_for: 'Hauling heavy freight, Sled pulling', ...}
    ▶ 9: {weight: {...}, height: {...}, id: 10, name: 'American Bulldog', breed_group: 'Working', ...}
    ▶ 10: {weight: {...}, height: {...}, id: 11, name: 'American Bully', country_code: 'US', ...}
    ▶ 11: {weight: {...}, height: {...}, id: 12, name: 'American Eskimo Dog', country_code: 'US', ...}
    ▶ 12: {weight: {...}, height: {...}, id: 13, name: 'American Eskimo Dog (Miniature)', country_code: 'US', ...}
    ▶ 13: {weight: {...}, height: {...}, id: 14, name: 'American Foxhound', country_code: 'US', ...}
    ▶ 14: {weight: {...}, height: {...}, id: 15, name: 'American Pit Bull Terrier', country_code: 'US', ...}
    ▶ 15: {weight: {...}, height: {...}, id: 16, name: 'American Staffordshire Terrier', country_code: 'US', ...}
    ▶ 16: {weight: {...}, height: {...}, id: 17, name: 'American Water Spaniel', country_code: 'US', ...}
    ▶ 17: {weight: {...}, height: {...}, id: 18, name: 'Anatolian Shepherd Dog', bred_for: 'Livestock herding', ...}
    ▶ 18: {weight: {...}, height: {...}, id: 19, name: 'Appenzeller Sennenhund', bred_for: 'Herding livestock, pulling car...'}
    ▶ 19: {weight: {...}, height: {...}, id: 21, name: 'Australian Cattle Dog', country_code: 'AU', ...}
    ▶ 20: {weight: {...}, height: {...}, id: 22, name: 'Australian Kelpie', country_code: 'AU', ...}
    ▶ 21: {weight: {...}, height: {...}, id: 23, name: 'Australian Shepherd', country_code: 'AU', ...}
    ▶ 22: {weight: {...}, height: {...}, id: 24, name: 'Australian Terrier', country_code: 'AU', ...}
    ▶ 23: {weight: {...}, height: {...}, id: 25, name: 'Azawakh', bred_for: 'Livestock guardian, hunting', ...}
```

Fetch API

- Also used to send (post) data back to an API endpoint
- An example would be to post data from a form and use an API to save the data

```
const url = 'https://jsonplaceholder.typicode.com/posts';

const data = {
  title: 'foo',
  body: 'bar',
  userId: 1
};

fetch(url, {
  method: 'POST',
  body: JSON.stringify(data),
  headers: {
    'Content-Type': 'application/json'
  }
}).then(response => response.json())
  .then(json => console.log(json))
  .catch((error) => console.error('Error:', error));
```

Async and Await

Async and Await

- New additions to the Javascript language to make chaining promises more readable
- Async always returns a promise

```
const getDogBreeds = async () => {  
};  
  
const test = getDogBreeds();  
console.log(test);
```

▼ *Promise {<fulfilled>: undefined}* ⓘ
▶ [[Prototype]]: Promise
[[PromiseState]]: "fulfilled"
[[PromiseResult]]: undefined

Async and Await

- New additions to the Javascript language to make chaining promises more readable
- Async always returns a promise
- Await stops assignment until a promise is resolved

```
const getDogBreeds = async () => {
  const response = await fetch('https://api.thedogapi.com/v1/breeds');
  console.log(response)
  const data = await response.json();
  console.log(data);
  return data;
};

console.log(1)

const dogs = getDogBreeds().then(data => {
  console.log("resolved: ", data);
});

console.log(2)
```

```
▶ Response {type: 'cors', url: 'https://api.thedogapi.com/v1/breeds', redirected: false, status: 200, statusText: 'OK', headers: Headers, body: ReadableStream, bodyUsed: false}
  (172) [{}]
    ▶ 0: {weight: {}, height: {}, id: 1, name: 'Affenpinscher', bred_for: 'Small rodent hunting', country_code: 'DE', description: 'The breed is a small dog with a long, shaggy coat and a bushy tail.', temperament: 'Affenpinschers are alert, active, and curious dogs. They are intelligent and can be trained to perform various tasks.', origin: 'The breed originated in Germany, where it was developed as a ratter and companion dog.', coat_type: 'The coat is thick and shaggy, with a dense undercoat and a topcoat that is slightly wavy or curly. It comes in a variety of colors, including black, tan, and silver.', life_span: 'The life expectancy of an Affenpinscher is approximately 12-15 years.', size: 'Affenpinschers are small dogs, typically weighing between 10 and 15 pounds and standing about 10-12 inches tall at the shoulder.', activity_level: 'Affenpinschers are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Affenpinschers are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Affenpinschers are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Affenpinschers are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Affenpinschers have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 1: {weight: {}, height: {}, id: 2, name: 'Afghan Hound', country_code: 'AF', description: 'The Afghan Hound is a large, sighthound breed known for its distinctive long, flowing coat and elegant appearance. They are often referred to as "the greyhounds of the East".', temperament: 'Afghan Hounds are gentle and affectionate dogs, but they can be aloof with strangers. They are intelligent and independent, and may require some time to bond with their owners.', origin: 'The breed originated in Afghanistan, where it was used as a hunting dog to chase game like deer and antelope. It has been bred for its unique coat and graceful movement.', coat_type: 'The coat is long and flowing, with a dense undercoat and a topcoat that is wavy or curly. It comes in a variety of colors, including fawn, tan, and black.', life_span: 'The life expectancy of an Afghan Hound is approximately 10-12 years.', size: 'Afghan Hounds are large dogs, typically weighing between 50 and 70 pounds and standing about 24-28 inches tall at the shoulder.', activity_level: 'Afghan Hounds are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Afghan Hounds are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Afghan Hounds are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Afghan Hounds are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Afghan Hounds have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 2: {weight: {}, height: {}, id: 3, name: 'African Hunting Dog', bred_for: 'A wild pack hunting dog', country_code: 'TZ', description: 'The African Hunting Dog is a large, powerful breed that was originally bred for hunting wild animals in Africa. They are known for their exceptional hunting abilities and loyalty to their pack.', temperament: 'African Hunting Dogs are highly intelligent and energetic dogs that require a lot of mental stimulation and physical exercise. They are territorial and may be aggressive towards unfamiliar dogs, so early socialization is important.', origin: 'The breed originated in Africa, where it was used by tribes to hunt large game like lions and elephants. It has been bred for its hunting skills and strength.', coat_type: 'The coat is short and smooth, with a dense undercoat and a topcoat that is slightly wavy or curly. It comes in a variety of colors, including black, tan, and brown.', life_span: 'The life expectancy of an African Hunting Dog is approximately 10-12 years.', size: 'African Hunting Dogs are large dogs, typically weighing between 70 and 100 pounds and standing about 24-28 inches tall at the shoulder.', activity_level: 'African Hunting Dogs are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'African Hunting Dogs are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'African Hunting Dogs are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'African Hunting Dogs are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'African Hunting Dogs have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 3: {weight: {}, height: {}, id: 4, name: 'Airedale Terrier', bred_for: 'Badger, otter hunting', country_code: 'GB', description: 'The Airedale Terrier is a large, rugged breed that was originally bred for hunting badgers and otters in the Airedale region of England. They are known for their intelligence, tenacity, and fearless nature.', temperament: 'Airedale Terriers are intelligent and勇敢的狗，需要大量的精神刺激和体力锻炼。它们是领地性的，可能对不熟悉的狗有攻击性，所以早期的社会化很重要。', origin: 'The breed originated in England, where it was used by farmers to hunt badgers and otters. It has been bred for its hunting skills and strength.', coat_type: 'The coat is long and shaggy, with a dense undercoat and a topcoat that is wavy or curly. It comes in a variety of colors, including black, tan, and silver.', life_span: 'The life expectancy of an Airedale Terrier is approximately 10-12 years.', size: 'Airedale Terriers are large dogs, typically weighing between 50 and 70 pounds and standing about 24-28 inches tall at the shoulder.', activity_level: 'Airedale Terriers are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Airedale Terriers are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Airedale Terriers are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Airedale Terriers are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Airedale Terriers have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 4: {weight: {}, height: {}, id: 5, name: 'Akbash Dog', bred_for: 'Sheep guarding', country_code: 'TR', description: 'The Akbash Dog is a large, white breed that was originally bred to guard sheep in Turkey. They are known for their protective nature and loyalty to their flock.', temperament: 'Akbash Dogs are intelligent and勇敢的狗，需要大量的精神刺激和体力锻炼。它们是领地性的，可能对不熟悉的狗有攻击性，所以早期的社会化很重要。', origin: 'The breed originated in Turkey, where it was used by nomadic tribes to guard sheep and goats. It has been bred for its protective nature and strength.', coat_type: 'The coat is long and shaggy, with a dense undercoat and a topcoat that is wavy or curly. It comes in a variety of colors, including white, cream, and light tan.', life_span: 'The life expectancy of an Akbash Dog is approximately 10-12 years.', size: 'Akbash Dogs are large dogs, typically weighing between 70 and 100 pounds and standing about 24-28 inches tall at the shoulder.', activity_level: 'Akbash Dogs are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Akbash Dogs are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Akbash Dogs are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Akbash Dogs are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Akbash Dogs have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 5: {weight: {}, height: {}, id: 6, name: 'Akita', bred_for: 'Hunting bears', country_code: 'JP', description: 'The Akita is a large, muscular breed that was originally bred for hunting bears in Japan. They are known for their strength, agility, and tenacity.', temperament: 'Akitas are intelligent and勇敢的狗，需要大量的精神刺激和体力锻炼。它们是领地性的，可能对不熟悉的狗有攻击性，所以早期的社会化很重要。', origin: 'The breed originated in Japan, where it was used by hunters to track and kill bears. It has been bred for its hunting skills and strength.', coat_type: 'The coat is long and shaggy, with a dense undercoat and a topcoat that is wavy or curly. It comes in a variety of colors, including red, black, and tan.', life_span: 'The life expectancy of an Akita is approximately 10-12 years.', size: 'Akitas are large dogs, typically weighing between 50 and 70 pounds and standing about 24-28 inches tall at the shoulder.', activity_level: 'Akitas are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Akitas are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Akitas are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Akitas are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Akitas have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 6: {weight: {}, height: {}, id: 7, name: 'Alapaha Blue Blood Bulldog', description: 'The Alapaha Blue Blood Bulldog is a rare breed that was originally bred for bull baiting in the United States. They are known for their unique blue-colored skin and white coat.', temperament: 'Alapaha Blue Blood Bulldogs are intelligent and勇敢的狗，需要大量的精神刺激和体力锻炼。它们是领地性的，可能对不熟悉的狗有攻击性，所以早期的社会化很重要。', origin: 'The breed originated in the United States, where it was used by farmers to bait bulls. It has been bred for its unique appearance and strength.', coat_type: 'The coat is short and smooth, with a dense undercoat and a topcoat that is slightly wavy or curly. It comes in a variety of colors, including blue, white, and tan.', life_span: 'The life expectancy of an Alapaha Blue Blood Bulldog is approximately 10-12 years.', size: 'Alapaha Blue Blood Bulldogs are large dogs, typically weighing between 50 and 70 pounds and standing about 24-28 inches tall at the shoulder.', activity_level: 'Alapaha Blue Blood Bulldogs are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Alapaha Blue Blood Bulldogs are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Alapaha Blue Blood Bulldogs are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Alapaha Blue Blood Bulldogs are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Alapaha Blue Blood Bulldogs have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
    ▶ 7: {weight: {}, height: {}, id: 8, name: 'Alaskan Husky', bred_for: 'Sled pulling', country_code: 'US', description: 'The Alaskan Husky is a medium-sized breed that was originally bred for sled pulling in Alaska. They are known for their endurance and speed.', temperament: 'Alaskan Huskies are intelligent and勇敢的狗，需要大量的精神刺激和体力锻炼。它们是领地性的，可能对不熟悉的狗有攻击性，所以早期的社会化很重要。', origin: 'The breed originated in Alaska, where it was used by native peoples to pull sleds. It has been bred for its endurance and speed.', coat_type: 'The coat is long and shaggy, with a dense undercoat and a topcoat that is wavy or curly. It comes in a variety of colors, including white, black, and tan.', life_span: 'The life expectancy of an Alaskan Husky is approximately 10-12 years.', size: 'Alaskan Huskies are medium-sized dogs, typically weighing between 30 and 50 pounds and standing about 20-24 inches tall at the shoulder.', activity_level: 'Alaskan Huskies are high-energy dogs that require daily exercise to stay happy and healthy. They enjoy running, playing fetch, and exploring their surroundings.', social_needs: 'Alaskan Huskies are social dogs that thrive on attention and interaction with their owners and other dogs. They are known for their friendly and outgoing personalities.', training: 'Alaskan Huskies are intelligent and responsive to training. They can learn basic commands like sit, stay, and come quickly, but may require patience and consistency from their owners during training sessions.', health_issues: 'Alaskan Huskies are generally healthy dogs, but like all breeds, they are susceptible to certain health issues. These may include dental problems, skin allergies, and eye infections. Regular veterinary check-ups and proper dental care can help prevent many of these issues.', diet: 'Alaskan Huskies have a high metabolism and require a balanced diet to maintain their energy levels. A diet rich in protein and carbohydrates is recommended, along with plenty of fresh water and occasional treats.'}
```

What's Next

- Week 3 Contact
Convert imperative TicTacToe Javascript game to functional Javascript.
- Week 4 Contact
Using Fetch to pull data from SpaceX API and caching (very useful for Assessment 1).

How do I learn and complete Functional Programming Assessment?

Week	Items to complete
Week 2 Lecture: Introduction to Functional Programming Week 2 Contact: Recap Javascript and convert Python to Javascript imperatively	<ul style="list-style-type: none">• Make sure proxy server runs on your computer• Come up with a strategy to link together the data files• Write Javascript to load the .txt as csv files• Read in user input from the command line
Week 3 Lecture: Iteration and Recursion; Calling API's Week 3 Contact: Convert TicTacToe to a functional program	<ul style="list-style-type: none">• Validate user input• Write pure functions• Find and filter loaded data based on user input
Week 4 Contact: Call API's	<ul style="list-style-type: none">• Call API's (from the URL's of the proxy server)• Link in Live Arrival time and Location
Week 5 Contact: Cache data from API's	<ul style="list-style-type: none">• Cache data from API calls



CREATE CHANGE

Fetch API Q&A



CREATE CHANGE

Thank you