Ionic Study

Day 6

오늘 할 것들

- 자바스크립트 기본 문법 알아보기
  - 배열 다뤄보기
- HTML5와 DOM 개요

배열

- Array 객체
  - 배열
    - 데이터의 집합.
    - 일정한 주소 (인덱스)에 값 (원소)를 저장.
  - 자바스크립트에서 배열 사용
    - 자바스크립트에서는 전역객체로서 Array 객체를 제공.
    - 함수 생성자(new Array()) 또는 배열 리터럴([])로 배열 생성 가능.
    - 생성자의 매개변수 또는 리터럴 내부에 적당한 값을 주게되면?
      - 1. 생성자 함수의 매개변수가 숫자 단일값으로 지정된 경우 : 해당 크기만큼의 빈 공간을 가지는 배열로 초기화
      - 2. 생성자 함수의 매개변수가 ,로 구분된 2개 이상의 숫자 또는 1개 이상의 기타 자료일 때 : 배열 생성 시 연속된 공간의 원소값을 주어진 값으로 초기화함.

배열

- Array 객체
  - var a = new Array(1,2,3);

1	2	3

a

  - var b = ['레몬', '당근', 2018, 6, 4];

'레몬'	'당근'	2018	6	4

b

  - var c = new Array(4);

undefined	undefined	undefined	undefined

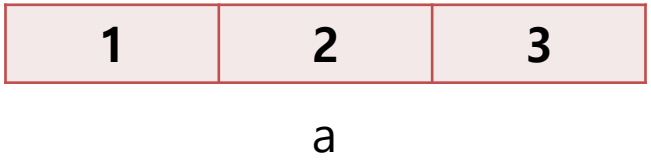
c

배열

- Array 객체
  - 원소에 접근하기
    - 인덱스로 원소에 접근할 수 있음.
    - 배열변수명 뒤에 [숫자]를 입력
      - ex) a[0], a[20] …

배열

- Array 객체
  - var a = new Array(1,2,3);



  - 인덱스로 접근

    a[0] => 1

    a[1] => 2

    a[2] => 3

    a[3] => ?

배열

- 배열 순회
  - 제일 처음 원소부터 마지막 원소까지 탐색
  - 반복문을 활용하는 방법과 별도의 메소드를 이용하는 방법이 있음.
  - 반복문 활용하기
    - for문과 Array.prototype.length 메소드를 활용하여 배열을 순회.
    - length 메소드
      - 유효한 원소의 개수를 반환.

배열

- ex1.js

JS ex1.js

```
let arr = new Array(10);
arr[0] = '참외';
arr[1] = '바나나';
arr[2] = 1990;

for(var i=0; i<arr.length; i++) {
    console.log('arr['+ i +'] = '+ arr[i]);
}
```

배열

- 배열 순회
  - 순회 메소드 활용하기
  - Array.prototype.forEach 메소드
    - 배열요소마다 지정된 함수를 실행
    - 초기화되지 않은 원소 (undefined로 초기화된 원소)는 순회하지 않음
    - 배열을 순회하는것은 동일함. 다만 forEach 메소드는 순회하면서 콜백함수를 호출.
    - 순회를 정지하거나 탈출하는 방법이 없기 때문에 주의가 필요 (continue, break 사용 불가)
    - 콜백함수에 매개변수를 지정하면, 매개변수로 배열의 인덱스와 값이 각각 전달됨 .

배열

- ex2.js

JS ex1.js ● JS ex2.js ×

```
let arr = new Array(10);
arr[0] = '참외';
arr[1] = '바나나';
arr[2] = 1990;
arr[4] = '딸기바나나';

arr.forEach(function(value, index) {
    console.log('arr['+ index +'] = '+ value);
});
```

배열

- 다차원 배열 – 배열 안의 배열
  - 배열이 중첩되어 저장되어있는 구조.
  - n차원 배열이라고 부름
    - 단, 1차원 배열은 앞에 배열 앞에 1차원을 빼거나 벡터라고 부름.
  - 중첩되는 단계가 높아질수록 잡아먹는 메모리 ↑, 성능 ↓

배열

- 다차원 배열의 선언과 순회
  - 원소값이 배열로 저장되면 OK
  - 배열 리터럴로 선언하면 깔끔

    ex) var twoD = [[1, 2], [3, 4]] // 2x2 배열
  - 순회시 중첩 루프를 이용하여 순회 가능.

배열

- ex3.js – 반복문으로 순회

JS ex1.js ● JS ex2.js ● JS ex3.js ×

```
1   let twoD = [[1,2], [3,4], ['a', 99]];
2   twoD[3] = ['happy', 'fun', 'nozam'];
3
4   console.dir(twoD);
5
6   for (var i=0; i<twoD.length; i++) {
7       for (var j=0; j<twoD[i].length; j++) {
8           console.log('twoD['+ i +', '+ j +'] = '+ twoD[i][j]);
9       }
10  }
11
```

배열

- ex4.js – forEach로 순회

배열

- Array.prototype.map
  - 배열을 순회하며 모든 원소에 정해진 함수를 호출하고, 그 결과를 모아서 새로운 배열로 반환하는 메소드.
  - 원래 배열의 원소는 변형되지 않음.

배열

- ex5.js

JS ex5.js

```
let arr = [1, 2, 3, 4, 5];
let map;

map = arr.map(function(value, index) {
    return value * 2;
});

console.log('arr');
console.dir(arr);
console.log('map');
console.dir(map);
```

배열

- Array.prototype.reduce
  - 배열을 순회하며 배열의 각 원소마다 누적된 계산값을 적용하여 하나의 결과값으로 줄이는 함수.
  - 원래 배열의 원소는 변형되지 않음.

배열

- ex6.js

JS ex5.js JS ex6.js

```
let arr = new Array(10);
let reduce;

for (var i=1; i<=10; i++) {
    arr[i - 1] = i;
}

reduce = arr.reduce(function(acc, value) {
    return acc + value;
});

console.log('arr');
console.dir(arr);
console.log('reduce');
console.dir(reduce);
```

배열

- Array.prototype.filter
  - 배열을 순회하며 배열의 각 원소마다 정해진 함수의 결과가 참인 원소만 새로운 배열로 추출해주는 함수.
  - 원래 배열의 원소는 변형되지 않음.

배열

- ex7.js

JS ex5.js JS ex6.js JS ex7.js

```
let arr = new Array(10);
let filter;

for (var i=1; i<=10; i++) {
    arr[i - 1] = i;
}

filter = arr.filter(function(value) {
    return (value % 2 === 0);
});

console.log('arr');
console.dir(arr);
console.log('filter');
console.dir(filter);
```

배열

- Array.prototype.indexOf
  - 배열에서 정해진 요소값을 찾아 첫번째 인덱스를 반환하는 함수.
  - 요소값을 찾지 못할 경우 -1을 반환함.

배열

- ex8.js

JS ex5.js JS ex6.js JS ex7.js JS ex8.js

```
let arr = [2, 4, 4, 6, 4, 10];

let idx1, idx2, idx3;
idx1 = arr.indexOf(2);
idx2 = arr.indexOf(4);
idx3 = arr.indexOf(99);

console.log('value 2 position is : '+ idx1);
console.log('value 4 first position is : '+ idx2);
console.log('value 99 position is : '+ idx3);
```

배열

- Array.prototype.push
  - 배열의 끝에 하나 이상의 요소를 삽입하고 변경된 배열의 길이를 반환하는 메소드.

배열

- ex9.js

JS ex5.js JS ex6.js JS ex7.js

```
1   let arr = [0, 1];
2   let len;
3
4   console.dir(arr);
5
6   len = arr.push(2);
7   console.dir(arr);
8   console.log('len = '+ len);
9
10  len = arr.push(4, 6, 8);
11  console.dir(arr);
12  console.log('len = '+ len);
```

HTML5

- HTML
  - HyperText Markup Language
  - 하이퍼텍스트 문서를 만들기 위한 언어 규약.
  - 웹에서 사용하는 웹문서를 만들 때 사용.
  - ‘태그’ 라는 HTML 요소를 이용하여 문서 작성
    - HTML 태그 : <와 >로 구성된 요소.

HTML5

- ex10.html

```
<!doctype html>
<html>
    <head>
        <title>My first html 문서</title>
        <meta charset="utf-8">
    </head>
    <body>
        <h1>Hello? HTML!</h1>
        <p>HTML 문서를 만들어봤습니다.</p>
    </body>
</html>
```

HTML5

- HTML 태그
  - 요소, 속성, 값의 3가지 항목으로 정의
    - 요소 : 하나의 태그를 구성하는 단위
    - 속성 : 태그의 특성을 기술하는 단위
    - 값 : 태그의 내용을 기술하는 단위.
  - 모든 태그는 열고 닫는 태그가 쌍으로 존재해야 한다. - 정의의 목적
    - 닫는 태그는 태그명 앞에 반드시 '/'를 붙인다.
      - ex) <a>,
  - But, 닫는 태그 없이 단독으로 쓰이는 일부 태그가 존재. - 지시의 목적
    - ex) <hr>

HTML5

네이버

- 요소 : a 태그
- 이 요소의 속성 : href, title
- 이 요소의 값 : 네이버

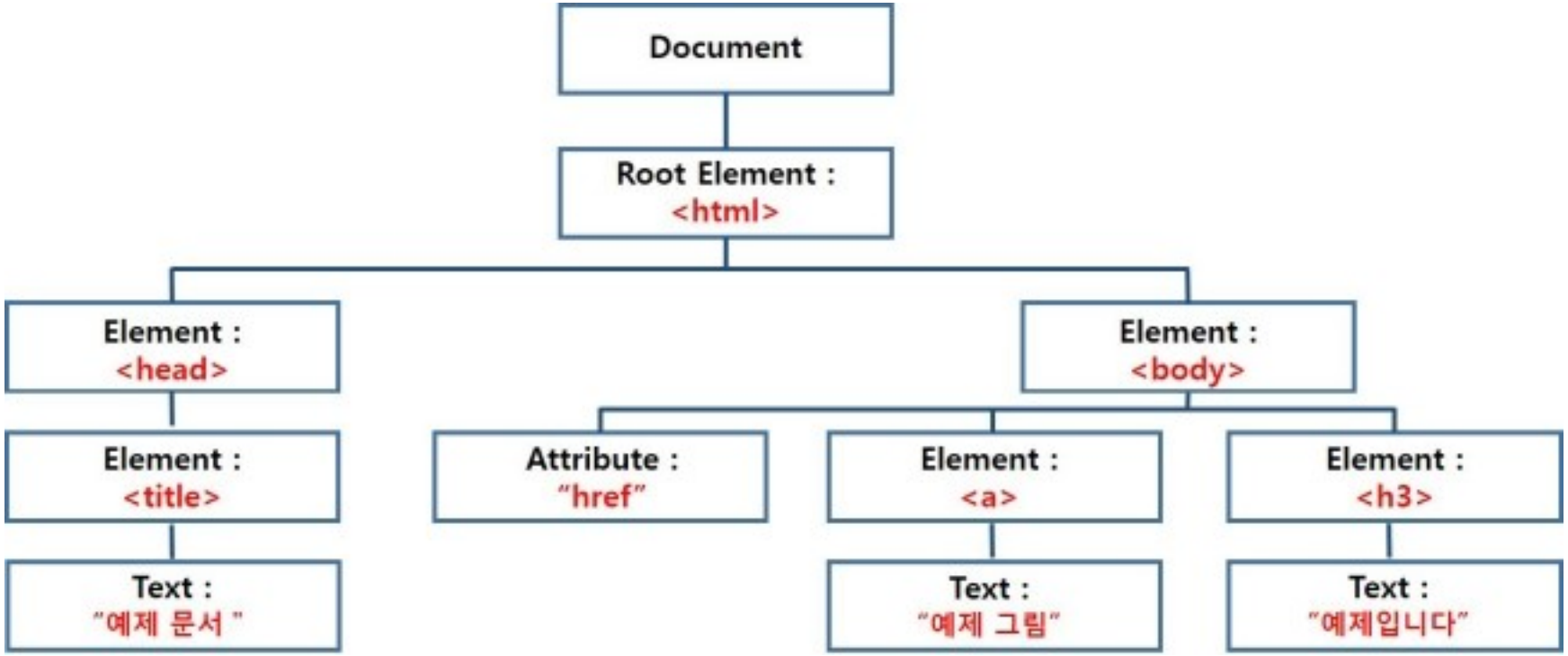
DOM

- HTML문서에서 자바스크립트 사용하기
  - <script>태그를 이용하여 자바스크립트를 사용.
  - HEAD 태그 내에 <script>태그를 사용하거나, BODY 태그 내에 <script>태그를 사용.

DOM

- Document Object Model의 약자
  - 문서 객체 모델
  - HTML 문서를 하나의 자바스크립트 객체로 취급.
    - 웹 브라우저는 웹 문서를 트리 구조를 가지는 작은 객체단위로 분해.
  - document라는 객체로 제공.

DOM



- 노드 : 트리를 구성하는 객체
  - 노드간에는 부모-자식 관계 성립.
  - 같은 단계의 노드간에는 형제 관계 성립.

DOM

- 요소를 선택해야 조작을 할 수 있다.
  - 요소의 속성, 값등을 바꾸기 위해서는 해당 요소를 우선 선택해야함.
    - document.getElementById()
    - document.querySelector() 등의 메소드 사용

DOM

- ID와 클래스
  - ID는 요소의 고유한 값을 정의하는 속성.
  - HTML 문서에서 단 한번만 정의되어야 함.
  - 클래스는 요소에서 반복되는 특성을 정의할 때 사용하는 속성

DOM

- document.getElementByID()
  - 해당 ID 속성값의 요소를 선택.
  - 올바르게 선택된 경우, 해당 요소를 객체형태로 반환함.
- Element.innerHTML
  - 요소의 내부 값을 가지고 있는 속성.
- Element.style
  - 요소의 스타일 (모양)을 가지고 있는 속성.

DOM

- ex11.html

```
<!doctype html>
<html>
    <head>
        <title>My first html 문서</title>
        <meta charset="utf-8">
    </head>
    <body>
        <h1>Hello? HTML!</h1>
        <p id="content">HTML 문서를 만들어봤습니다.</p>
    </body>
    <script>
        var p;
        p = document.getElementById('content');
        p.innerHTML = '짜자잔~ 내용이 바뀌었어요.';
        p.style.fontSize = '50px';
    </script>
</html>
```

오늘은 여기까지~

See you next day!