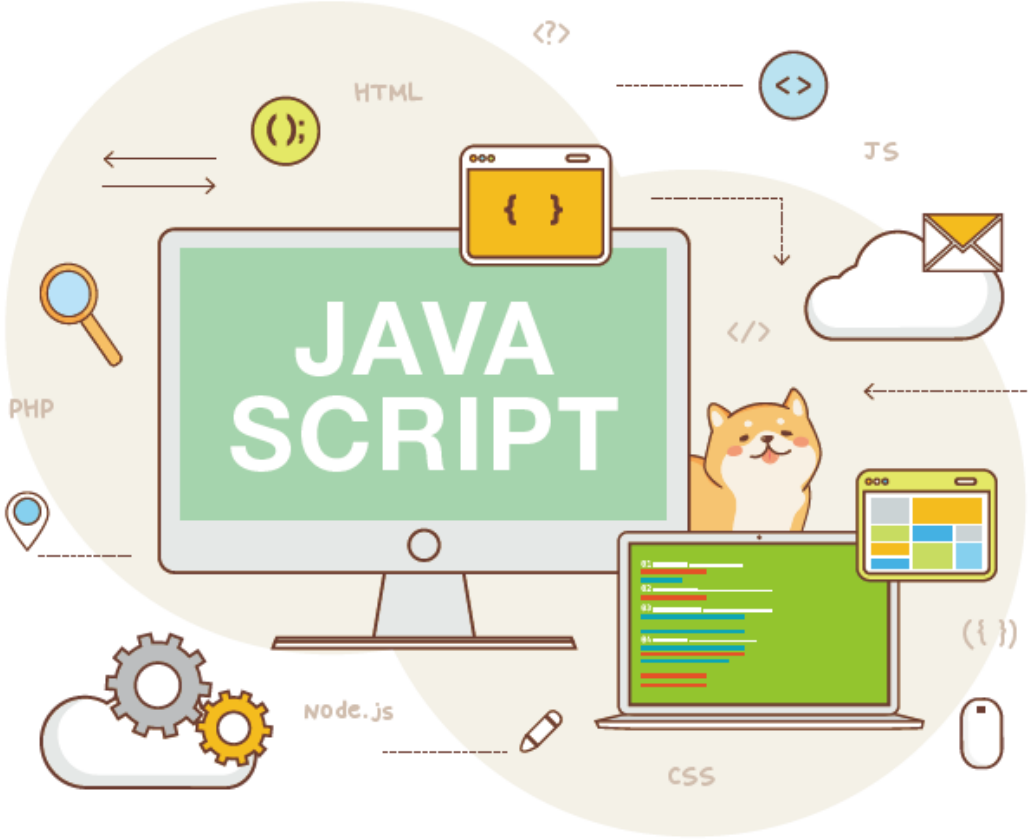
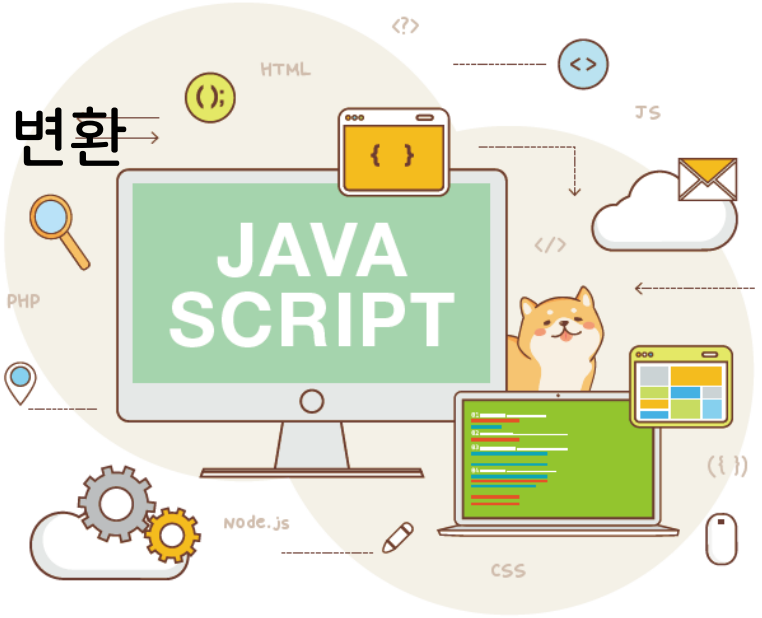


자바스크립트 기본
(Chapter7)

스마트헬스 케어 – 5주차

학습 내용

1. 다중 모듈(파일)을 이용한 프로그래밍
2. JSON 데이터 이해
3. JSON 데이터 다루기
4. JSON 문자열 자바스크립트 객체 간 변환



이론 및 예제 실습

1_다중 파일 사용 프로그래밍

1) 자바스크립트 코드가 들어간 파일은 [모듈명].js로 저장한다.

2) 자바스크립트 코드를 사용하려는 html 파일에서 <script> 태그로 자바스크립트 파일을 불러 온다

형식 : <script src="[JS파일명]"></script>

test.js

```
var num = 100;

function add100(value){
  return value + 100;
}
```

ex3.html

```
<script src="test.js"></script>
<script>
   console.log(num);

   let newNum = 200;
   console.log(add100(newNum));
</script>
```

2_JSON

- JSON 는 Douglas Crockford가 널리 퍼뜨린 Javascript 객체 문법을 따르는 문자 기반의 데이터 포맷
  JSON = Javascript Object Notation
- JSON 객체 : 객체를 사용하는 데이터 표현방법
  . 요소는 'key : value' 쌍으로 구성
  . 모든 키는 큰 따옴표로 감싸야 한다 → JS에서는 따옴표를 붙이지 않아도 됨
  . 값에는 숫자, 문자열, 불 자료형, 배열, 객체를 사용한다

```
let jsonObject = {
  squadName: "Super hero squad",
  homeTown: "Metro City",
  formed: 2016,
  secretBase: "Super tower",
  active: true,
  members: { name: "Molecule Man", age: 29, secretIdentity: "Dan Jukes",},

};

console.log(typeof jsonObject);
console.log(jsonObject);
console.log(jsonObject.squadName);
console.log(jsonObject.active);
```

2_Array, List, Map

- 배열(Array) : 데이터를 순차적으로 저장해 0부터 시작하는 인덱스를 통해 접근한다.

. 일반적으로 배열은 선언할 때 크기 고정

. 데이터를 임의 접근할 수 있어 접근을 효율적으로 할 수 있음

- 리스트(List) : 배열과 유사한 순차적인 자료구조를 제공하며 데이터 접근을 위해 인덱스를 사용해야 하는 점은 배열과 같지만 배열과 달리 초기에 크기를 고정하지 않는다.

. 데이터 크기가 고정되지 않음

. 데이터를 다루기 위한 여러 방법이 제공됨

. 리스트의 데이터는 서로 다른 타입일 수 있음

. 배열 중간에 값을 추가하거나 삭제하기 쉬움

→ 자바스크립트의 Array는 리스트에 가까움

- 맵(Map) : 데이터를 Key-Value(키-값)의 쌍으로 저장하는 방식이다.

맵을 사용했을 때 얻을 수 있는 가장 큰 장점은 원하는 데이터를 빠르게 찾을 수 있다는 점이다.

. 데이터를 저장할 때 해당 데이터를 찾기 위한 Key를 부여

. Key값을 알면 언제든 빠르게 데이터를 찾을 수 있음

. Value 에 객체형이 들어갈 수 있어 복잡한 데이터 처리가 가능

→ 자바스크립트에서 객체의 형식이 Map 구조임

3_JSON 데이터 다루기

- 내부 요소 접근
  . 배열형식 접근 : 객체명['요소키']
  . 객체지향형 접근 : 객체명.요소키

- 데이터 다루기
  . 값 사용 : let name = jsonObject.hometown;
  . 값 할당/수정 : jsonObject.hometown = "Seoul";
  . 요소 삭제 : delete jsonObject.hometown;

```
let jsonObject = {
        squadName: "Super hero squad",
        homeTown: "Metro City",
        formed: 2016,
        secretBase: "Super tower",
        active: true,
        members: {
          name: "Molecule Man",
          age: 29,
          secretIdentity: "Dan Jukes",
        },
};
```

jsonObject.hometown
jsonObject['hometown']

jsonObject.active
jsonObject['active']

jsonObject.member.age
jsonObject['member']['age']

3_JSON 데이터 다루기

```
let jsonObject = {
  squadName: "Super hero squad",
  homeTown: "Metro City",
  formed: 2016,
  secretBase: "Super tower",
  active: true,
  member: {
    name: "Molecule Man",
    age: 29,
    secretIdentity: "Dan Jukes",
  },
};

console.log(jsonObject.member.name);
console.log(jsonObject["member"]["age"]);
delete jsonObject.secretBase;
console.log(jsonObject);
console.log(jsonObject.secretBase);
```

3_JSON 데이터 다루기

- 요소 순회

```
for (let key in obj) {
   console.log(obj[key]);
}
```

```
for (let key in jsonObject) {
        console.log(key);
        console.log(jsonObject[key]);
}
```

3_배열과 JSON 복합 응용

- 리스트 구조와 Map 구조를 복합적으로 구성해서 쓰는 경우가 많음

1) 자바스크립트 객체배열 : 배열 요소로 객체를 사용

. 요소 접근 : 배열[인덱스].요소명

2) 요소에 배열 사용 : 객체요소의 값으로 배열을 사용

. 요소 접근 : 객체.요소명[인덱스]

```
const javascriptArray = [{name: '윤인성', region: ['서울']}, {name: '윤명월', region: ['도쿄']}];

console.log(javascriptArray[0].name);
console.log(javascriptArray[0].region);

for (let key in javascriptArray[1]) {
    console.log(javascriptArray[1][key]);
}
```

4_JSON 문자열과 자바스크립트 객체간 변환

- JSON 문자열 → 자바스크립트 객체 : JSON.parse([JSON 문자열])
- 자바스크립트 객체 → JSON 문자열 : JSON.stringify([자바스크립트 객체])

```
const javascriptObject = [
  { name: "윤인성", region: "서울" },
  { name: "윤명월", region: "도쿄" },
];

let outputA = JSON.stringify(javascriptObject);
console.log(outputA);

let outputB = JSON.parse(outputA);
console.log(outputB);
```

응용 예제

실습-1

실습 문제 : **국어** 성적 데이터에 대해 총점 및 평균을 구하여 출력 하시오.

실습-2

실습 문제 : 0~9 임의의 수 100개를 생성해서 각 숫자가 몇 개씩 생성되었는지 카운트

- 0~9사이 임의의 정수를 생성하는 코드

```
Math.floor(Math.random() * 10)
```

연속 완성 프로젝트

숫자 야구 게임 - 2단계

1. 0~9 사이의 임의의 숫자 3개를 생성 부분을 함수로 분리
2. 사용자 입력으로 3개의 수를 입력 받는 부분을 함수로 분리
3. 임의로 생성한 3개의 수와 사용자 입력 받은 수를 비교하는 부분(판정) 함수로 분리
4. 총 10회 반복하여 10회 이내에 클리어(스트라이크 3개) 하면 축하 메시지
   클리어 하지 못하면 '다음 기회에' 출력

숫자 야구 게임-2단계

```
let rand_num = [];
let user_input = [];
let strike_num = 0;
let ball_num = 0;
let msg = "";
const gameCount = 10;

genGameNumber();
console.log(rand_num);

for (let i = 1; i <= gameCount; i++) {
  alert(`${i}번째 시행`);

  getUserNumber();

  msg = compareNumber();
  alert(msg);

  if (strike_num == 3) {
    alert("축하합니다. 게임을 클리어 하였습니다.");
    break;
  } else if (i == gameCount) {
    alert("게임을 클리어 하지 못했습니다. 다음 기회에~~");
  }
}
```

숫자야구게임-2단계

```
function genGameNumber() {
  let value = 0;
  for (let i = 0; i < 3; i++) {
    value = Math.floor(Math.random() * 10); // 0~9
    rand_num.push(value);
  }
}
function getUserNumber() {
  user_input = [];
  for (let i = 0; i < 3; i++) {
    value = Number(prompt(i + 1 + "번째 숫자를 입력하세요 (0~9)", 1));
    user_input.push(value);
  }
}
function compareNumber() {
  strike_num = 0;
  ball_num = 0;
  let match_index = -1;
  for (i in user_input) {
    match_index = rand_num.indexOf(user_input[i]);
    if (match_index != -1) {
      if (match_index == i) {
        strike_num++;
      } else {
        ball_num++;
      }
    }
  }
  let resultStr = "판정  --   " + strike_num + " 스트라이트  ,  " + ball_num + " 볼 ";
  return resultStr;
}
```